Klaros-Testmanagement Benutzerdokumentation



Version 5.7.3

Veröffentlicht 8. Mai 2025



Copyright © 2009-2025 verit Informationssysteme GmbH

Klaros-Testmanagement Benutzerdokumentation

von Talal Arif, Sabrina Gidley, Fabian Klaffke, Claudia Könnecke, Klaus Mandola, Patrick Reilly und Torsten Stolpmann

Version 5.7.3

Veröffentlicht 8. Mai 2025 Copyright © 2009-2025 verit Informationssysteme GmbH

Zusammenfassung

Dieses Dokument dient als Referenzdokumentation für Klaros-Testmanagement. Es enthält eine detaillierte Beschreibung der Benutzeroberfläche und der bereitgestellten Funktionalität.

Rechtliche Hinweise. Copyright 2009-2025 verit Informationssysteme GmbH, Europaallee 10, 67657 Kaiserslautern, Deutschland. Alle Rechte vorbehalten. Dieses Produkt oder Dokument ist urheberrechtlich geschützt und wird unter Lizenzen, die seine Verwendung, Vervielfältigung, Verbreitung und Dekompilierung einschränken. Kein Teil dieses Produkts oder der Dokumentation darf ohne vorherige schriftliche Genehmigung der verit Informationssysteme GmbH in irgendeiner Form vervielfältigt werden.



Markenzeichen

Docker™ und das Docker-Logo sind Marken oder eingetragene Marken von Docker, Inc. in den Vereinigten Staaten und/oder anderen Ländern. Docker, Inc. und andere Parteien können auch Markenrechte an anderen hier verwendeten Begriffen haben.

OracleTM und JavaTM sind Marken von Oracle und/oder deren Tochtergesellschaften.

Microsoft®, Excel®, SQLServer®, und Windows® sind entweder eingetragene Marken oder Marken der Microsoft Corporation in den Vereinigten Staaten und/oder anderen Ländern.

JIRA® ist eine eingetragene Marke von Atlassian Pty Ltd.

Andere Namen können Marken der jeweiligen Eigentümer sein.

Inhaltsverzeichnis

1. Feature-Überblick	. 1
1.1. Verwalten von testbezogenen Daten	1
1.2. Planen von Tests und Testaufgaben	2
1.3. Ausführen von Tests	2
1.4. Auswerten von Tests, Statistiken, Berichte	. 3
1.5. Konfiguration	4
2. Einführung	. 5
2.1. Aufbau und Objekte	5
2.2. Projekt - Ein Testvorhaben verwalten	5
2.3. Testfall - Einen Test definieren	. 5
2.4. Testschritt - Den Ablauf der Testaktivität festlegen	6
2.4.1. Testschrittergebnis	. 7
2.4.2. Automatisierte Testfälle	7
2.5. Testsegment - Testschritte wiederverwenden	7
2.6. Testsuite - Gruppieren von Testfällen	8
2.7. Testsystem - Das zu testende Objekt	8
2.8. Testumgebung - Der äußere Einfluss auf das Testergebnis	. 8
2.9. Testlauf - Die Ergebnisse der ausgeführten Testfälle	. 9
2.10. Anforderung - Eigenschaften des Testsystems	9
2.11. Iteration - Das Projekt in Phasen unterteilen	10
2.12. Aufgabe - Planen der Testaktivität	10
2.13. Benutzer - Benutzer und ihre Rollen	11
2.14. Übersicht über die Objekte	12
2.15. Issue - Anlegen und Beheben von Fehlern	12
3. Installation	14
3.1. Lizenzmodell für die Community und Enterprise Edition	14
3.2. Systemvoraussetzungen	14
3.2.1. Client-Voraussetzungen	14
3.2.2. Server-Voraussetzungen	15
3.3. Installation	17
3.3.1. Sprachauswahl	18
3.3.2. Schritt 1: Willkommen	18
3.3.3. Schritt 2: Informationen	19
3.3.4. Schritt 3: Lizenzvereinbarung	19
3.3.5. Schritt 4: Installationspfad	20
3.3.6. Schritt 5: Auswahl Installationspakete	21
3.3.7. Schritt 6: Benutzerdaten	22
3.3.8. Schritt 7: Installation	23
3.3.9. Schritt 8: Externe Prozesse ausführen	25
3.3.10. Schritt 9: Verknüpfungen einrichten	25
3.3.11. Schritt 10: Installation abgeschlossen	26
3.4. Konsolen-basierte Installation	27
3.4.1. Schritt 1: Sprachauswahl	27
3.4.2. Schritt 2: Willkommen	28
3.4.3. Schritt 3: Informationen	28
3.4.4. Schritt 4: Lizenzvereinbarung	28
3.4.5. Schritt 5: Installationspfad	29

3.4.6. Schritt 6: Auswahl Installationspakete	29
3.4.7. Schritt 7: Benutzerdaten	29
3.4.8. Schritt 8: Installation	30
3.4.9. Schritt 9: Externe Prozesse ausführen	31
3.4.10. Schritt 10: Verknüpfungen einrichten	31
3.4.11. Schritt 11: Installation beenden	31
3.5. Automatische Installations-Skripte	31
3.6. Starten der Anwendung	32
3.7. Stoppen der Anwendung	33
3.8. Zugang zur Anwendung	33
3.9. Updates einspielen	33
3.10. Wichtige Dateiverzeichnisse	34
3.10.1. Log-Dateien	35
3.10.2. Datenbankeinstellungen	35
3.10.3. Sprachunterstützung	35
3.10.4. Die Datei "quotes.txt"	35
3.10.5. Die Derby-Datenbank	35
3.10.6. Das Content-Repository	35
3.11. Die Standard-Datenbank ändern	36
3.11.1. MariaDB	36
3.11.2. Microsoft SQL-Server	36
3.11.3. MySQL	36
3.11.4. PostgreSQL	37
3.11.5. Apache Derby	37
3.11.6. Einrichten der Datenbank-Instanz	37
3.12. Als System-Service installieren	38
3.12.1. Als Linux-Service installieren	38
3.12.2. Als Windows-Service installieren	39
3.13. Monitoring	42
3.14. Externe Issue-Management-Systeme konfigurieren	43
3.14.1. Jira konfigurieren	43
3.14.2. Redmine konfigurieren	44
3.14.3. Trac Konfiguration	44
3.15. SSL-Unterstützung	45
3.16. Upgrade von Version 4	45
3.16.1. Erforderliche Version für das Upgrade	46
3.16.2. Unbeaufsichtigtes Datenbank-Update	46
3.16.3. Startup-Zeit für das erste Upgrade auf Version 5	46
3.16.4. Eingebettete Java-Laufzeitumgebung	46
3.16.5. Automatische Erkennung der Datenbank-Version	46
3.16.6. Oracle-Datenbank wird nicht mehr unterstützt	46
3.17. Deinstallation	46
Anpassungen	48
4.1. Sprachdateien aktivieren und deaktivieren	48
4.1.1. Sprachdateien definieren	48
4.2. Zitat des Tages	49
Funktionale Übersicht	50
5.1. Anmelden	50
5.2. Seitenübersicht	51

4.

5.

5.2.1. Seiten-Menü	51
5.2.2. Kopfzeile	52
5.2.3. Inhalt	55
5.3. Hauptfunktionen	71
5.3.1. Konflikte beim Speichern lösen	71
5.3.2. Objekte löschen, dauerhaft entfernen und wiederherstellen	72
5.3.3. Objekt-Eigenschaften referenzieren	75
5.3.4. Referenzieren von Anhängen	76
6. Definieren	78
6.1. Projekte	78
6.1.1. Übersichtsseite	78
6.1.2. Detailseite	80
6.2. Iterationen	90
6.2.1. Übersichtsseite	90
6.2.2. Detailseite	93
6.3. Anforderungen	
6.3.1. Übersichtsseite	98
6.3.2. Detailseite	. 101
6.4 Testumgebungen	105
6 4 1 Übersichtsseite	105
6 4 2 Detailseite	107
6.5. Testsysteme	111
6.5.1 Ühersichtsseite	111
6.5.2 Detailseite	113
6.6 Testsegmente	119
6.6.1 Ühersichtsseite	110
6.6.2 Detailseite	121
6.7 Teetfälle	124
6.7.1 Übersichtsseite	124
6.7.2 Detailseite	124
6.8 Tectsuiten	127
6.8.1 Übersichtsseite	132
6.8.2 Detaileaite	1/0
7 Planen	1/15
7.1 Aufrahen	145
7.1.1 Übereichteseite	1/15
7.1.2 Detaileaite	1/10
7.1.2. Detailseite	155
7.2. Aufgaben aus Testsuiten	155
7.1. Aufgaben nach Benutzer	150
7.4.1 Aufgaben nach Benutzer - Details	150
8 Ausführen	161
8.1 Meine Aufrahen	161
8 1 1 Aktion	162
8.1.2 Tahellenonerationen	162
8.1.3 Fine Aufrabe ausführen	162
8.2 Teetfall ausführen	162
9.2. Netion	161
8.2.2. Tabellenonerationen	161
0.2.2. Tabelienoperationen	. 104

8.2.3. Einen Testfall ausführen	164
8.3. Testsuiten ausführen	170
8.3.1. Aktion	171
8.3.2. Tabellenoperationen	171
8.3.3. Eine Testsuite ausführen	172
8.4. Testlauf fortsetzen	176
8.4.1. Aktion	177
8.4.2. Massenaktionen	178
8.4.3. Tabellenoperationen	178
8.4.4. Testlauf fortsetzen	179
8.5. Testergebnisse importieren	179
8.5.1. Unterstützte Frameworks	181
9. Auswerten	185
9.1. Das Dashboard	185
9.1.1. Standard-Dashboard	186
9.1.2. Dashboard-Berichte bearbeiten	186
9.1.3. Berichtstypen	187
9.2. Berichte	194
9.2.1. Vordefinierte Berichte	194
9.3. Testläufe	198
9.3.1. Aktion	200
9.3.2. Massenaktionen	200
9.3.3. Tabellenoperationen	200
9.3.4. Der Testlauf - Bericht	201
9.3.5. Detailseite	202
9.3.6. Änderungen	203
9.4. Testfallergebnisse	204
9.4.1. Aktion	205
9.4.2. Massenaktionen	205
9.4.3. Tabellenoperationen	206
9.4.4. Ergebnisse eines einzelnen Testfalls	206
9.5. Testsuiteergebnisse	211
9.5.1. Aktion	211
9.5.2. Tabellenoperationen	212
9.5.3. Ergebnisse einer einzelnen Testsuite	212
9.6. Issues	215
9.6.1. Übersichtsseite	215
9.6.2. Aktion	217
9.6.3. Massenaktionen	217
9.6.4. Tabellenoperationen	217
9.6.5. Detailseite	218
9.6.6. Issue - Details	222
9.6.7. Einen Issue verlinken	230
10. Konfigurieren	232
10.1. Überblick	232
10.2. Berichtsvorlagen	233
10.2.1. Aktion	234
10.2.2. Berichtsvorlage - Detailansicht	234
10.3. Benutzer	235

10.3.1. Übersichtsseite	235
10.3.2. Detailseite	239
10.4. System	242
10.4.1. Sonstiges	242
10.4.2. Benachrichtigungen	243
10.4.3. Testausführung	247
10.4.4. Oberfläche	248
10.4.5. Sprachen	249
10.5. Integration	250
10.5.1. Issue Management	250
10.5.2. Anforderungsmanagement	259
10.5.3. E-Mail	261
10.5.4. Netzwerk	263
10.5.5. LDAP	264
10.5.6. CAS	267
10.6. Sicherung	268
10.6.1. Export	268
10.6.2. Import	269
11. Benutzerdefinierte Berichte	271
11.1. Das Kontext-Objekt	272
11.2. Eine neue Berichtsvorlage erstellen	272
11.2.1. Unterstützte Parametertypen	275
11.2.2. Umgang mit Parametern	275
11.3. Einen Bericht erzeugen	276
11.4. Beispielbericht	277
11.4.1. Erzeugen des Skripts	277
11.4.2. Eine PDF-Layout-Vorlage erzeugen	277
11.5. Ein Diagramm erstellen	278
11.5.1. Script für ein Tortendiagramm	279
11.5.2. Pie Chart Report Template	279
11.5.3. Bilder einbinden	280
11.5.4. Eine Layout-Vorlage für Excel erzeugen	280
12. Import/Export	282
12.1. Testfälle aus Excel-Tabellen importieren	282
12.1.1. Import-Format	282
12.1.2. Vorbedingungen	285
12.1.3. Ausführen	285
12.2. Testfälle aus XML-Dateien importieren	286
12.2.1. Vorbedingungen	286
12.2.2. Ausführen	286
12.3. Anforderungen aus Excel-Tabellen importieren	286
12.3.1. Import-Format	286
12.3.2. Vorbedingungen	288
12.3.3. Ausführen	288
12.4. Importieren und Synchronisieren von Anforderungen aus XML	288
12.4.1. Vorbedingungen	289
12.4.2. Importieren	289
12.4.3. Synchronisierung	289
12.5. Testergehnisse importieren	291

12.5.1. Das JUnit XML++ Format	291
12.5.2. Vorbedingungen	293
12.5.3. Ausführen	293
12.5.4. Jenkins Plugin	296
12.6. Testergebnisse exportieren	296
12.6.1. Ausführen	296
12.7. Tabelleninhalte exportieren	297
12.8. Backup und Wiederherstellen	298
12.8.1. Backup via REST	298
13. Die Remote API	299
13.1. Überblick	299
13.2. REST-Clients	299
13.3. Dokumentation	300
A. Zugriffsberechtigungen	301
A.1. Rollenberechtigungen – Community Edition	301
A.2. Rollenberechtigungen – Enterprise Edition	301
B. Die Klaros Object Model API Referenz	302
B.1. Klaros Object Model API Reference	302
B.1.1. de.verit.klaros.core.model	302
B.1.2. Deprecated API	417
B.2. Scripting API Reference	417
B.2.1. de.verit.klaros.scripting	417
B.2.2. de.verit.klaros.scripting.context	421
B.2.3. de.verit.klaros.scripting.model	426
C. Dateiformat für Testfall-Import	430
C.1. <step></step>	432
C.2. <steps></steps>	432
C.3. <attachments></attachments>	432
C.4. <attachment></attachment>	432
C.5. <attribute></attribute>	433
C.6. <attributes></attributes>	433
C.7. <category></category>	433
C.8. <categorytree></categorytree>	433
C.9. <container></container>	434
C.10. <testcases></testcases>	434
C.11. <testcase></testcase>	434
C.12. <testsuite></testsuite>	435
C.13. <externalld></externalld>	435
C.14. <areatopic></areatopic>	436
C.15. <depends></depends>	436
C.16. <description></description>	436
C.17. <docbase></docbase>	436
C.18. <estimatedduration></estimatedduration>	436
C.19. <evaluation></evaluation>	436
C.20. <execution></execution>	436
C.21. <expectedresult></expectedresult>	436
C.22. <level></level>	437
C.23. <method></method>	437
C.24. <note></note>	437

C.25 <pre>c.25 <pre>condition></pre></pre>	437
C.26 <pre>precondition></pre>	437
C 27 < revision>	437
C_{28} <pre>continue</pre>	437
C 20 <shortnamas< td=""><td>137</td></shortnamas<>	137
C_{29} showing C_{29}	437
C_{31} <teams< td=""><td>120</td></teams<>	120
C_{22} $<$ traccobility	. 430 120
C_{22} <voriation< td=""><td>430</td></voriation<>	430
C.33. <valiety< td=""><td>430</td></valiety<>	430
	430
	438
	438
C.37. <categoryname></categoryname>	. 438
C.38. <categorydescription></categorydescription>	. 438
D. Dateiformat für Anforderungs-Import	. 439
D.1. <attachments></attachments>	. 440
D.2. <attachment></attachment>	440
D.3. <attributes></attributes>	440
D.4. <attribute></attribute>	440
D.5. <category></category>	440
D.6. <categorytree></categorytree>	. 441
D.7. <container></container>	441
D.8. <requirements></requirements>	. 441
D.9. <requirement></requirement>	441
D.10. <externaltestcaseids></externaltestcaseids>	442
D.11. <externalid></externalid>	. 442
D.12. <externalrevision></externalrevision>	442
D.13. <description></description>	442
D.14. <priority></priority>	443
D.15. <revision></revision>	. 443
D.16. <shortname></shortname>	443
D.17. <summarv></summarv>	443
D18 <name></name>	443
D 19 <value></value>	443
D 20 <content></content>	443
D 21 <categoryname></categoryname>	443
D.22 <category description=""></category>	.
E Die Reporting Context API	115
E 1 Der Klares Pepert Kentext	. 445
E.1. Der Klaros Report-Kontext	445
E.2. Ridlosscript Interface	. 445
	440
r. verzeichnis aller icons	449
F.I. Dereiche	449
	449
F.2.1. AKTIONEN	449
F.2.2. Austuhrungsaktionen	450
F.2.3. Aktionen bei der Testausführung	450
F.2.4. Ordnende Aktionen	451
F.3. Tabellenoperationen	451

F.4. Information	451
F.5. Eigenschaften	452
F.5.1. Ergebnisse	452
F.5.2. Prioritäten	452
F.6. Dokumentenformate	452
Glossar	454
Stichwortverzeichnis	465

Abbildungsverzeichnis

2.1. Testfallstruktur	5
2.2. Testschritt	6
2.3. Testsegment	8
2.4. Testsuite	8
2.5. Testlauf mit Testergebnissen	9
2.6. Aufgabe	11
2.7. Übersicht über die Objekte	12
3.1. Sprachauswahl	18
3.2. Die Seite "Willkommen"	18
3.3. Die Seite "Informationen"	19
3.4. Die Seite "Lizenzvereinbarungen"	20
3.5. Die Seite "Installationspfad"	21
3.6. Die Seite "Auswahl Installationspakete"	22
3.7. Die Seite "Benutzerdaten"	23
3.8. Die Seite "Installation"	24
3.9. Die Seite "Installation beendet"	24
3.10. Die Seite "Prozesse"	25
3.11. Die Seite "Verknüpfungen einrichten"	26
3.12. Die Seite "Installation abgeschlossen"	27
3.13. Zusätzliche Tomcat Speichereinstellungen und Java Optionen	41
3.14. Zusätzliche Tomcat-Einstellungen	42
3.15. Der Redmine-Authentifikations-Bereich	44
3.16. Trac-Plugin-Bereich	45
3.17. Trac-Permission-Bereich	45
3.18. Das Deinstallationsprogramm	47
5.1. Die Seite "Anmelden"	50
5.2. Die Seite "Projekte"	51
5.3. Das Suchfeld	52
5.4. Die Seite "Suchergebnisse"	53
5.5. Die Status-Zeile	53
5.6. Das Icon "Log-Panel"	54
5.7. Das Log-Panel	54
5.8. Das Hilfe-Menü	54
5.9. Das "Benutzermenü"	55
5.10. Das geöffnete Benutzermenü	55
5.11. Anzahl der aktiven Filter	56
5.12. Das Panel "Filtern und Sortieren"	56
5.13. Nach Mehrfachauswahl filtern	57
5.14. Die Ansicht "Kategorien bearbeiten"	58
5.15. Kategorien auswählen	59
5.16. Die Auswahl der Massenaktionen	60
5.17. Massenaktionen auf Objekte	60
5.18. Die Druckansicht "Testsuite"	62
5.19. Lesezeichen erstellen	63
5.20. Die Ansicht "Übersicht"	64
5.21. Die Ansicht "Benutzerdefinierte Eigenschaften"	65
5.22. Bearbeiten einer Aufzählung	66

5.23. Die Ansicht "Benutzerdefinierte Eigenschaften"	66
5.24. Die Ansicht "Revisionen"	67
5.25. Die Seite "Anhänge hochladen"	69
5.26. Die Detailseite "Ergebnisse"	70
5.27. Die Ansicht "Änderungen"	. 71
5.28. Der Dialog "Konflikt-Auflösung"	72
5.29. Der Dialog "Objekte löschen"	73
5.30. Der Dialog "Objekte dauerhaft entfernen"	74
5.31. Der Dialog "Objekte wiederherstellen"	74
5.32. Aufgelöste Objekteigenschaft	75
6.1. Die Übersichtsseite "Projekte"	78
6.2. Schnelle Auswahl von Projekten	80
6.3. Die Seite "Projekte - Eigenschaften"	80
6.4. Die Ansicht "Benutzerdefinierte Eigenschaften"	81
6.5. Die Ansicht "Objekte kopieren"	82
6.6. Der Dialog "Objekte kopieren"	83
6.7. Die Ansicht "Zugang"	84
6.8. Der Dialog "Projektrolle zuweisen"	85
6.9. Die Ansicht "Integration/Issue Management"	86
6.10. Die Ansicht "Integration/Anforderungsmanagement"	87
6.11. Die synchronisierten Typen des Anforderungsmanagementsystems	88
6.12. Verknüpfen von Aufzählungswerten	89
6.13. Die Übersichtsseite "Iterationen"	90
6.14. Schnelle Auswahl von Projekten und Iterationen	92
6.15. Die Ansicht "Übersicht"	93
6.16. Die Erfolgsrate der Iteration	94
6.17. Der Iteration-Radar-Chart	95
6.18. Die Seite "Iterationen - Eigenschaften"	96
6.19. Die Übersichtsseite "Anforderungen"	98
6.20. Die Seite "Anforderungen" mit Remote-Synchronisation	101
6.21. Die Ansicht "Übersicht"	102
6.22. Die Ansicht "Eigenschaften"	103
6.23. Die Ansicht "Testfälle"	104
6.24. Die Ansicht "Revisionen"	105
6.25. Die Übersichtsseite "Testumgebungen"	105
6.26. Die Ansicht "Übersicht"	108
6.27. Die Ansicht "Übersicht"	109
6.28. Die Ansicht "Eigenschaften"	110
6.29. Die Ansicht "Iterationen"	111
6.30. Die Übersichtsseite "Testsysteme"	112
6.31. Die Ansicht "Übersicht"	114
6.32. Die Ansicht "Übersicht"	115
6.33. Die Ansicht "Eigenschaften"	116
6.34. Die Ansicht "Issues"	117
6.35. Die Ansicht "Iterationen"	118
6.36. Die Seite "Testsegmente"	119
6.37. Die Ansicht "Eigenschaften"	122
6.38. Die Ansicht "Schritte"	123
6.39. Die Übersichtsseite "Testfälle"	125

6.40. Die Ansicht "Übersicht"	128
6.41. Die Ansicht "Eigenschaften"	131
6.42. Die Ansicht "Schritte"	132
6.43. Der Dialog "Anhang einfügen"	133
6.44. Ein Verweis auf einen Anhang im Feld Erwartetes Ergebnis	133
6.45. Ein Verweis auf einen Anhang wird durch eine Vorschau des Anhangs ersetzt	134
6.46. Ein Verweis auf einen Anhang wird durch einen Hyperlink zum Anhang ersetzt	135
6.47. Die Ansicht "Issues"	136
6.48. Die Ansicht "Aufgaben"	137
6.49. Die Seite "Testsuiten"	138
6.50. Die Ansicht "Testsuite - Übersicht"	141
6.51. Die Ansicht "Eigenschaften"	142
6.52. Aufgaben	143
7.1. Die Seite "Aufgaben verwalten"	145
7.2. Die Ansicht "Übersicht"	149
7.3. Statusgraph einer Aufgabe	151
7.4. Die Ansicht "Eigenschaften"	152
7.5. Die Ansicht "Abhängigkeiten"	153
7.6. Der "Abhängigkeit hinzufügen" Dialog	153
7.7. Der Dialog "Einen Kommentar hinzufügen"	154
7.8. Der Dialog "Arbeit protokollieren"	155
7.9. Die Seite "Aufgaben aus Testfällen"	156
7.10. Der "Ausführung als Aufgabe vorsehen" Dialog	156
7.11. Der "Review-Aufgabe anlegen" Dialog	157
7 12 Die Seite Aufgaben aus Testsuiten"	157
7.13. Der "Ausführung als Aufgabe vorsehen" Dialog	158
7 14 Der Review-Aufgabe anlegen" Dialog	158
7 15 Die Seite Aufgaben nach Benutzer"	159
7 16. Die Detailseite Aufgaben nach Benutzer"	160
81 Die Seite Meine Aufgaben"	161
8 2 Die Seite "Testfälle ausführen"	163
8.3 Der Dialog "Testfall ausführen"	165
8 4 Die Ansicht "Schritt-für-Schritt-Anleitung"	166
8.5 Die Ansicht Tabellarische Ansicht"	167
8.6 Der Dialog Testschrittergebnis bearbeiten"	168
8.7 Die Seite Testlauf - Übersicht"	169
8.8 Fine Review-Aufgabe anlegen	170
8.9 Die Seite Testsuiten ausführen"	171
8 10 Der Dialog. Testsuite ausführen"	172
8 11 Die Ansicht Testfallübersicht"	173
8 12 Der Dialog. Finen Testfall dauerhaft überspringen"	174
8.13 Der Dialog "Eine Begründung einfügen"	175
8.14 Die Ansicht Testlauf - Übersicht"	175
8 15 Die Ansicht Testsuitelauf - Übersicht"	176
8 16 Die Seite Ausführung fortsetzen"	177
8 17 Die Seite Ausführung fortsetzen"	170
8.18 Die Seite Fraehnisse importieren"	180
8 10. Die Seite Ergebnisse importieren"	100
9.1 Die Seite Deshoard"	185
	100

9.2. Der Dialog "Report konnguneren	187
9.3. Der Bericht "Projektübersicht"	188
9.4. Der Bericht "Letzte Erfolgsrate"	189
9.5. Der Bericht "Testaktivität"	189
9.6. Der Bericht "Testsystem Übersicht"	190
9.7. Der Bericht "Testumgebungen - Übersicht"	191
9.8. Der Bericht "Testfortschritt"	192
9.9. Der Bericht "Testhistorie"	193
9.10. Die Seite "Berichtsvorlagen"	194
9.11. Der Bericht "Testumgebung"	195
9.12. Der Bericht "Testsystem"	196
9.13. Der Bericht "Testsuite"	197
9.14. Der Bericht "Testlauf-Ergebnishistorie"	197
9.15. Einen parametrisierten Bericht erzeugen	198
9.16. Die Seite "Testläufe"	199
9.17. Der Bericht "Testlauf"	201
9.18. Der Bericht "Testlauf" (Fortsetzung)	201
9 19 Die Detailseite eines Testlaufs	203
9 20 Die Seite Testfallergebnisse"	204
9.21 Der Dialog Testfälle zu Testsuite hinzufügen"	205
9.22 Die Seite Testfallergebnisse"	206
9.22. Die Seite Testfallergebniese	208
9.24 Fin Testschrittergebnis hearbeiten	200
9.25. Die Druckansicht Testfallergebnis"	200
9.26. Begründung für das Überspringen eines Testfalls anzeigen	210
7 / C C E C C C C C C C C C C C C C C C C	ZIU
9.27 Die Seite Testsuiteergehnisse"	211
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211
 9.20. Degrundung für das oberspringen eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite Testsuiteergebniss" 	211 212 214
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211 212 214 215
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 219
 9.20. begründung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 219 220
 9.20. begrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 219 220 221
 9.20. begrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 219 220 221 221
 9.20. begrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 219 220 221 221 221 222
 9.20. begrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 220 221 221 221 222 224
 9.20. begrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 220 221 221 222 224 225
 9.20. Degründung für das oberspringen eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitHub" 9.40. Die Seite Jssue Details - GitHub" 	211 212 214 215 216 219 220 221 221 222 224 225 225
 9.20. Degrundung für das oberspringen eines restrans anzeigen	211 212 214 215 216 219 220 221 220 221 222 224 225 225 226
 9.20. begrundung für das oberspringen eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" in eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details Eugzilla" 9.39. Die Seite "Issue Details - GitHub" 9.40. Die Seite "Issue Details - JIRA" 	211 212 214 215 216 219 220 221 221 222 224 225 225 226 227
 9.20. begründung für das Oberspringen eines Testnahs anzeigen	211 212 214 215 216 219 220 221 221 222 224 225 225 225 226 227 228
 9.20. begründung für das oberspringer eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitLab" 9.40. Die Seite "Issue Details - Mantis" 9.43. Die Seite "Issue Details - Redmine" 	211 212 214 215 216 219 220 221 220 221 222 224 225 225 225 226 227 228 229
 9.20. begrundung für das oberspringen eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" iür eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue erstellen" 9.39. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitHub" 9.40. Die Seite "Issue Details - Mantis" 9.41. Die Seite "Issue Details - Redmine" 9.43. Die Seite "Issue Details - Redmine" 9.44. Die Seite "Issue Details - Redmine" 	211 212 214 215 216 219 220 221 221 222 224 225 225 225 226 227 228 229 230
 9.20. begründung für das öberspringen eines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testfälle zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue erstellen" 9.39. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitHub" 9.40. Die Seite "Issue Details - JIRA" 9.42. Die Seite "Issue Details - Redmine" 9.44. Die Seite "Issue Details - Trac" 9.45. Die Seite "Issue Details - Trac" 9.44. Die Seite "Issue Details - Trac" 	211 212 214 215 216 219 220 221 221 222 224 225 225 225 225 225 225 225 226 227 228 229 230 230
 9.20. begrundung für das Oberspringeneines restrans anzeigen 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitLab" 9.40. Die Seite "Issue Details - JIRA" 9.42. Die Seite "Issue Details - Redmine" 9.43. Die Seite "Issue Details - Trac" 9.44. Die Seite "Issue Details - Trac" 9.45. Die Seite "Issue Details - YouTrack" 9.46. Das "Issue verknüpfen" Panel 10.1 Die Seite übersicht" 	211 212 214 215 216 219 220 221 220 221 222 224 225 225 225 225 226 227 228 229 230 230 230
 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details GitHub" 9.39. Die Seite "Issue Details - GitLab" 9.40. Die Seite "Issue Details - Mantis" 9.43. Die Seite "Issue Details - VouTrack" 9.44. Die Seite "Issue Details - Trac" 9.45. Die Seite "Issue Details - Trac" 9.46. Das "Issue verknüpfen" Panel 10. 1. Die Seite "Übersicht" 	211 212 214 215 216 219 220 221 221 222 224 225 225 225 225 225 226 227 228 229 230 230 232 232
 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebniss" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme zuordnen" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue Petails Bugzilla" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details GitLab" 9.40. Die Seite "Issue Details - GitLab" 9.41. Die Seite "Issue Details - JIRA" 9.42. Die Seite "Issue Details - JIRA" 9.44. Die Seite "Issue Details - Kedmine" 9.44. Die Seite "Issue Details - Trac" 9.45. Die Seite "Issue Details - Track" 9.45. Die Seite "Berichtsvorlagen" 	211 212 214 215 216 219 220 221 221 222 224 225 225 225 225 225 225 225 225
 9.27. Die Seite "Testsuiteergebnisse" für eine Testsuite anzeigen 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite 9.29. Die Detailseite "Testsuiteergebnis" 9.30. Die Druckansicht "Testsuiteergebnis" 9.31. Die Übersichtsseite "Issues" 9.32. Die Ansicht "Eigenschaften" 9.33. Die Ansicht "Testfälle" 9.34. Der Dialog "Testfälle zuordnen" 9.35. Die Ansicht "Testsysteme" 9.36. Der Dialog "Testsysteme zuordnen" 9.37. Die Seite "Issue erstellen" 9.38. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details Bugzilla" 9.39. Die Seite "Issue Details - GitLab" 9.41. Die Seite "Issue Details - Mantis" 9.42. Die Seite "Issue Details - Redmine" 9.44. Die Seite "Issue Details - Redmine" 9.44. Die Seite "Issue Details - Redmine" 9.45. Die Seite "Issue Details - VouTrack" 9.46. Das "Issue verknüpfen" Panel 10.1. Die Seite "Berichtsvorlagen" 	211 212 214 215 216 219 220 221 221 222 224 225 225 225 226 227 228 229 230 230 232 233 234 225

10.5. Die Übersichtsseite "Benutzer"	236
10.6. Der Dialog "Benutzer zusammenführen"	237
10.7. Der Dialog "Neuen Benutzer speichern"	238
10.8. Die Ansicht "Eigenschaften"	239
10.9. Die Ansicht "Projektrollen"	241
10.10. Der Dialog "Projektrollen zuweisen"	241
10.11. Die Ansicht "Sonstiges"	242
10.12. Die Übersichtsseite "Benachrichtigungsschemata"	244
10.13. Der "Benachrichtigung hinzufügen" Dialog	245
10.14. Projekte einem Benachrichtigungsschema zuweisen	246
10.15. Die Ansicht "Testausführung"	247
10.16. Der Dialog "Begründung bearbeiten"	248
10.17. Die Ansicht "Oberfläche"	248
10.18. Die Ansicht "Sprachen"	250
10.19. Die Ansicht "Issue Management"	252
10.20. Die Bugzilla-Projekt-ID	253
10.21. Die Jira-Projekt-ID	253
10.22. Die Mantis-Projekt-ID	253
10.23. Die Redmine-Projekt-ID	254
10.24. Die YouTrack-Projekt-ID	255
10.25. Die Ansicht "Eigenschaften"	257
10.26. Bearbeiten der Status-Zuordnungen	258
10.27. Die Ansicht "Anforderungsmanagement"	260
10.28. Die Ansicht "E-Mail"	262
10.29. Die Ansicht "Netzwerk"	263
10.30. Die Ansicht "LDAP"	264
10.31. Der Dialog "LDAP-Authentifizierung"	267
10.32. Die Ansicht "CAS"	267
10.33. Die Ansicht "Export"	269
10.34. Die Ansicht "Import"	270
11.1. Ablauf der Berichtserstellung	271
11.2. Die Seite "Berichtsvorlagen"	273
11.3. Die Seite "Berichtsvorlagen - Details"	273
11.4. Die Ansicht "Parameter"	274
11.5. Einen Bericht erzeugen	276
11.6. Berichtsparameter eingeben	276
11.7. Ein Beispiel für ein Tortendiagramm	279
12.1. Beispiel für einen Testfall in Excel	283
12.2. Beispiel für eine Excel-Tabelle mit Anforderungen	287
12.3. Tabelleninhalte in Excel-Tabellen exportieren	298
13.1. JSON-Ausgabe nach der Suche nach einem Benutzernamen	299
13.2. Remote-API Dokumentation	300

Tabellenverzeichnis

3.1. Unterstützte Betriebssysteme	15
3.2. Unterstützte Datenbanksysteme	. 15
3.3. Unterstützte Externe Issue-Management-Systeme	. 43
12.1. Koordinaten der allgemeinen Eigenschaften von Testfällen	283
12.2. Koordinaten der Testschritte	284
12.3. Koordinaten der benutzerdefinierten Eigenschaften von Testfällen	285
12.4. Koordinaten für allgemeine Eigenschaften von Anforderungen	287
12.5. Koordinaten für benutzerdefinierte Eigenschaften	288
A.1. Rollenberechtigungen – Community Edition	301
A.2. Rollenberechtigungen – Enterprise Edition	301
C.1. Übersicht der Elemente	430
C.2. <step> -Elemente</step>	432
C.3. <steps> -Elemente</steps>	432
C.4. <attachments> -Elemente</attachments>	432
C.5. <attachment> -Elemente</attachment>	432
C.6. <attribute> -Elemente</attribute>	433
C.7. <attributes> -Elemente</attributes>	433
C.8. <category> -Elemente</category>	433
C.9. <categorytree> -Elemente</categorytree>	433
C.10. <container> -Elemente</container>	434
C.11. <testcases> -Elemente</testcases>	434
C.12. <testcase> -Elemente</testcase>	434
D.1. Übersicht der Elemente	439
D.2. <attachments> -Elemente</attachments>	440
D.3. <attachment> -Elemente</attachment>	440
D.4. <attributes> -Elemente</attributes>	440
D.5. <attribute> -Elemente</attribute>	440
D.6. <category> -Elemente</category>	441
D.7. <categorytree> -Elemente</categorytree>	441
D.8. <container> -Elemente</container>	441
D.9. <requirements> -Elemente</requirements>	441
D.10. <requirement> -Elemente</requirement>	442
D.11. <externaltestcaseids> -Elemente</externaltestcaseids>	442
E.1. Kontextvariablen	445
F.1. Bereiche	449
F.2. Allgemeine Aktionen	449
F.3. Ausführungsaktionen	450
F.4. Aktionen bei der Testausführung	450
F.5. Ordnende Aktionen	451
F.6. Tabellenoperationen	451
F.7. Information	451
F.8. Results	452
F.9. Prioritäten	452
F.10. Dokumentenformate	452

Liste der Beispiele

3.1. Beispiel auto-install.xml	32
3.2. Zusätzliche Einträge in mysql.cnf um UTF-8 Character Sets zu unterstützen	38
12.1. Testfallimport aus Excel-Tabellen über die Kommandozeile	. 285
12.2. XML-Testfall-Import über die Kommandozeile	286
12.3. Anforderungen aus Excel-Tabellen über die Kommandozeile importieren	288
12.4. Anforderungen aus XML-Dateien über die Kommandozeile importieren	289
12.5. XML-Anforderungssynchronisation über die Kommandozeile	. 290
12.6. XML-Anforderungsimport über die Kommandozeile	. 290
12.7. Beispiel Testschrittergebnis.xml	292
12.8. Beispiel QF-Test Import-URL	295
12.9. Curl Kommandozeilen Beispiel	. 296
12.10. Powershell curl Alias Kommandozeilen-Beispiel	. 296
12.11. Curl Kommandozeilen Beispiel	. 297
12.12. Powershell curl Alias Kommandozeilen-Beispiel	. 297

Kapitel 1. Feature-Überblick

1.1. Verwalten von testbezogenen Daten

Gemeinsames Verwalten aller testrelevanten Daten	Alle Daten eines Testprojekts wie <i>Testfälle, Testsuiten, Test-</i> <i>anforderungen, Testumgebungen, Testsysteme, Testläufe</i> und <i>Testfallergebnisse</i> werden gemeinsam in einer Datenbank gespeichert. Das Verwalten, Bearbeiten und Auswerten der Daten erfolgt über eine moderne und komfortable Weboberflä- che.
Agiles Testmanagement und klassische Prozesse	Neben klassischen Entwicklungsprozessen wie dem Wasser- fall- oder V-Modell, werden auch iterative und agile Methoden wie Scrum oder Kanban unterstützt.
Anforderungen und Anforde- rungsüberdeckung	Testanforderungen lassen sich entweder in Klaros-Test- management direkt verwalten oder mit externen Quellen wie z.B. JIRA synchronisieren. Die jeweilige Anforderungsüberde- ckung lässt sich jederzeit aktuell ablesen.
Modularisieren von Testschrit- ten	Häufig wiederkehrende Testschritte lassen sich zu Segmen- ten zusammenfassen und können projektübergreifend wieder- verwendet werden. Testfallschritte lassen sich so wie in einem Baukastensystem einfügen und mit minimalem Aufwand pfle- gen.
Gemeinsame Nutzung	Testfälle und Testsuiten können bei Bedarf zentral gewar- tet und in mehreren Projekten gleichzeitig verwendet werden. Änderungen sind nur noch an einer Stelle nötig und können gezielt an ausgewählte Projekte weitergegeben werden.
Änderungen verfolgen	Für ein lückenloses Nachvollziehen aller Änderungen an den verwalteten Daten wie Testfällen, Testschritten u.a. werden alle Aktualisierungen automatisch protokolliert und in einer Änderungshistorie detailliert dargestellt.
Benutzerdefinierte Felder	Jedes Testprojekt lässt sich durch benutzerdefinierte Felder individuell konfigurieren. Diese Felder stehen für Aufgaben, Anforderungen, Iterationen, Testsegmente, Testfälle, Testsui- ten, Testumgebungen, Testsysteme und Testläufe zur Verfü- gung.
Kategorisierung, Individuelle Sichten	Alle beteiligten Objekte lassen sich durch eine Kategorisierung in einer oder mehreren Baumstrukturen anordnen. Diese ist ähnlich wie in einem Dateisystem nach individuellen Kriterien aufgebaut und gestattet verschiedene Sichten auf ein Testpro- jekt.
Anhänge	Zusätzliche Informationen wie Testdaten, Textdokumente, Grafiken, Screenshots etc. lassen sich in die Anwendung hoch- laden, speichern und den Objekten Iteration, Aufgabe, Anforde-

rung, System under Test, Testumgebung, Testsegment, Testfall, Testschritt, Testsuite und Testfallergebnisse zuordnen.

Import und ExportBereits vorhandene Anforderungen und Testfälle können aus
XML und Excel-Dateien importiert werden. Über eine REST-
Schnittstelle findet bei Bedarf ein kontinuierlicher Abgleich mit
einer externen Datenquelle oder Anwendung statt.RevisionenKlaros-Testmanagement unterstützt die Versionierung von

Klaros-Testmanagement unterstutzt die Versionierung von Testanforderungen, Testsegmente, Testfällen und Testsuiten. Die einzelnen Versionen können getrennt gepflegt, ausgeführt und ausgewertet werden.

1.2. Planen von Tests und Testaufgaben

Erzeugen und Zuweisen von Aufgaben	Auf Knopfdruck werden Aufgaben zum Ausführen von Testfäl- len und Testsuiten erzeugt und einzelnen Personen zugeord- net. Der Status der Aufgabenausführung ist jederzeit ersicht- lich.
Testkoordination, Testsequen- zen	Aufgaben können in Teilaufgaben zerlegt, hierarchisch ange- ordnet und Testern zugewiesen werden. Durch das Definieren von Abhängigkeiten lassen sich Vorbedingungen für aufeinan- derfolgende Testsequenzen festlegen.
Auslastung, Fortschritt und Erfolg	Übersichtliche Tabellen und Diagramme zeigen Auslastung, Dauer und Fortschritt der Testaktivität für jeden Benutzer an. Fortschritts- und Erfolgsrate der Tests sind direkt und stets aktuell ersichtlich.
Testfall Reviews	Auch Testfälle können fehlerhaft sein. Das Überprüfen und Korrigieren dieser Tests kann als spezielle Aufgabe an Benut- zer vergeben werden. Der weitere Verlauf wird automatisch protokolliert.

1.3. Ausführen von Tests

Geführte manuelle Testausfüh- rung	Klaros-Testmanagement führt den Tester Schritt für Schritt durch die manuellen Tests. Der Testverlauf wird automatisch protokolliert und stellt die gewonnenen Daten für das Aus- werten zur Verfügung. Bei jedem einzelnen Schritt lassen sich Anmerkungen und Anhänge zur Dokumentation des Test- ergebnisses hinzufügen.
Unterbrochene Tests wieder- aufnehmen	Die manuelle Ausführung kann jederzeit angehalten und zu einem späteren Zeitpunkt wieder aufgenommen werden.
Testdaten und Testanweisun- gen trennen	Die in Testfällen verwendeten Eingabe- und Prüfdaten wie Log- ins, IDs u.a. lassen sich extern in Excel-Tabellen definieren und abspeichern. Tests sind damit gezielt parametrisierbar und können in unterschiedlichen Szenarien wiederverwendet wer- den.

Continuous Integration (Jenk- ins/Hudson)	Ein Plug-in für den Jenkins Continuous Integration Server über- trägt automatisch die in einem Build anfallenden Testergeb- nisse zur Auswertung in Klaros-Testmanagement.
Integration mit Issue-Manage- ment-Systemen	Gefundene Fehler lassen sich während der Testausführung direkt in Issue Management Systemen wie Bugzilla, GitHub, GitLab, JIRA, Mantis, Redmine oder Trac erstellen und ändern, ohne Klaros-Testmanagement verlassen zu müssen.
Import-Schnittstelle für Testau- tomatisierung	Testergebnisse, die von Testautomatisierungstools erzeugt werden, können manuell oder per REST-Schnittstelle importiert und mit manuellen Testergebnissen zusammengeführt wer- den. Es werden hierbei über 30 verschiedene Formate und Tools unterstützt: <i>AUnit, Boost Test, Check, CppTest, CppUnit,</i> <i>ctest, CUnit, embUnit, Fitnesse, Free Pascal Unit, Gauge</i> (via xml-report plugin), <i>GoogleTest, Glib / gtester, JBehave, JMe-</i> <i>ter, Jubula / GUIDancer, JSUnit, JUnit, MbUnit, MSTest, NUnit,</i> <i>PHPUnit, QF-Test, QTestLib, Ranorex, Selenium</i> (via JUnit XML result format), <i>Squish</i> (via JUnit XML result format), <i>TESSY,</i> <i>TestComplete, TestNG, TUSAR, UFT / QTP, UnitTest++, Valgrind</i> und <i>xUnit.net.</i>
Integration mit Last- und Per- formanz-Test-Tools	Um einen genauen Stand über die aktuelle Softwarequalität zu haben, ist es notwendig, die Ergebnisse aller Testarten - auch die der nicht-funktionalen - mit einzubeziehen. In Klaros-Test- management können auch die Ergebnisse aus Last- und Per- formanz-Test-Tools importiert, aufbereitet und ausgewertet werden. Unterstützte Lasttest-Software: <i>BlazeMeter, JMeter,</i> <i>Gatling, OctoPerf, SmartMeter.</i>

1.4. Auswerten von Tests, Statistiken, Berichte

Konfigurierbares Dashboard	Das Dashboard bietet zahlreiche Diagramme, auf denen sich auf einen Blick Zustand und Fortschritt der einzelnen Aktivitä- ten und des gesamten Projektes ablesen lässt. Es kann vom Benutzer nach individuellen Anforderungen konfiguriert wer- den. Berichte lassen sich in verschiedene Formate exportie- ren.
Integrierte Berichte und Aus- wertungen	Zahlreiche Auswertungen und Statistiken können einfach aus den in der Datenbank gespeicherten Informationen aufberei- tet und abgerufen werden. Klaros-Testmanagement enthält bereits zahlreiche vordefinierte Berichte.
Benutzerdefinierte Berichte	Mit Hilfe einer Programmierschnittstelle, die Zugriff auf den gesamten Datenbestand bietet, können individuell gestaltete Berichte im PDF oder Excel-Format frei definiert und erzeugt werden.
Excel Export von Datentabellen	Für ein eventuelles Nachbearbeiten und Auswerten lassen sich Datentabellen in eine Excel-Datei exportieren.

Druckfreundliche Seiten	Die meisten Ansichten, Diagramme und Tabellen lassen sich auch in einem druckfreundlichen Layout darstellen und direkt vom Browser aus auf den Drucker ausgeben. Verschiedene Anzeigeoptionen sind einstellbar.
Export von Berichten	Die Berichtsdateien können in verschiedene Dateiformate

exportiert werden, wie z.B.PDF, HTML und CSV.

1.5. Konfiguration

Rechte und Rollensystem	Klaros-Testmanagement unterstützt eine feinkörnige rollenba- sierte Benutzerverwaltung, die mit einem Rechte- und Rollen- system verbunden ist. Auf diese Weise können individuelle Zuordnungen vorgenommen oder auch die Zugriffs- und Bear- beitungsrechte einzelner Benutzer für kritische Daten einge- schränkt werden.
E-Mail-Benachrichtigungen	Der Status für ausgewählte Ereignisse, wie "Aufgabe zugewie- sen", "Testausführung fehlgeschlagen", "Aufgabe zur Ausfüh- rung bereit" u.v.m. kann auf Wunsch den beteiligten Benutzer oder anderen Personen als E-Mail-Benachrichtigung zugestellt werden. Hierzu existieren zahlreiche Konfigurationsmöglich- keiten, die auf einzelne Projekte individuell zugeschnitten wer- den können.
LDAP und Active Directory	Die Authentifizierung der Benutzer und das Verwalten der Passwörter kann direkt in Klaros-Testmanagement oder über ein externes LDAP- oder Active Directory-Verzeichnis erfolgen.
Single Sign-On Authentifizie- rung via CAS	Klaros-Testmanagement unterstützt auch die Single Sign-On Authentifizierung via CAS (Central Authentication Service).
Docker Integration	Zusätzlich zur nativen Installation unter Windows und Linux kann Klaros-Testmanagement auch als Container innerhalb einer Docker-basierten Umgebung betrieben werden. Die Anwendung kann so isoliert installiert und verwaltet werden, Updates sind einfacher durchzuführen. Unterstützte Daten- banken: Apache Derby, MariaDB, MySQL, PostgreSQL, Micro- soft SQL Server.
Remote Data Interface (REST)	Die in Klaros-Testmanagement gespeicherten Daten können per REST-Schnittstelle anderen Tools zur Verfügung gestellt werden. Individuelle Integrationen mit eigenen Werkzeugen lassen sich einfach realisieren. Dies erlaubt eine Vielzahl von zusätzlichen Integrationsmöglichkeiten.
Sichern und Wiederherstellen	Die Datensicherung erfolgt im XML-Format. Es lassen sich ein- zelne Projekte oder der gesamte Bestand speichern und bei Bedarf wiederherstellen.

Kapitel 2. Einführung

2.1. Aufbau und Objekte

Klaros-Testmanagement ist eine webbasierte Testmanagementsoftware zum Organisieren, Ausführen und Auswerten aller Testaktivitäten. Der Funktionsumfang deckt alle Bereiche des Testprozesses ab: Testplanung, Testerstellung, Testdurchführung, Zuordnen und Auswerten von Testaufgaben sowie Testauswertung und Berichtserstellung.

In dieser Einführung werden der innere Aufbau und die Funktion der einzelnen Bestandteile, der *Objekte*, beschrieben. Als *Objekte* werden in Klaros-Testmanagement die folgenden Elemente bezeichnet: **Projekt**, **Testfall**, **Testschritt**, **Testsegment**, **Testschrittergebnis**, **Testfallergebnis**, **Testsuite**, **Testsuiteergebnis**, **Testumgebung**, **Testsystem**, **Testlauf**, **Anforderung**, **Iteration** und **Aufgabe**.

2.2. Projekt - Ein Testvorhaben verwalten

Das Objekt auf oberster Ebene ist das *Projekt*. Es enthält alle anderen Objekte die zum Definieren, Planen, Ausführen und Auswerten eines bestimmten Testvorhabens erforderlich sind.

Projekte sind in der Regel in sich abgeschlossene Einheiten, es gibt jedoch auch Möglichkeiten, auf Objekte in anderen Projekten zu verweisen. Weitergehende Informationen zu Projekten finden Sie unter <u>Section 6.1</u>.

2.3. Testfall - Einen Test definieren

Testfälle sind die zentralen Objekte eines Projektes und können eine beliebige Anzahl an Testschritten enthalten. Sie können manuell oder automatisiert ausgeführt werden.



Abbildung 2.1. Testfallstruktur

Testfälle können mit Attributen versehen werden und detailliertere Informationen erhalten:

2.3.1. Beschreibende Attribute

Beschreibung

Vorbedingung

Nachbedingung

Erwartetes Ergebnis

Testschritte

2.3.2. Klassifizierende Attribute

Ausführungsart (Manuell/Automatisiert)

Priorität (Hoch/Mittel/Niedrig)

Status (Entwurf/Gesperrt/Genehmigt/Auslassen)

Testart (Funktional, Nicht-Functional, Strukturtest, Regressionstest, Fehlernachtest)

Entwurfsverfahren (Black-Box, White-Box)

Teststufe (Komponententest, Integrationstest, Systemtest, Abnahmetest)

Ergebnisart (Positiv, Negativ)

Anforderungsbezug (Dokument/Verfolgbarkeit)

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Für spezielle Anforderungen können in der Enterprise Edition zusätzlich eigene Felder definiert werden.

Die Testfälle bestehen aus einzelnen Testschritten und Testsegmenten, in denen die Testaktivitäten definiert und beschrieben werden.

2.4. Testschritt - Den Ablauf der Testaktivität festlegen

In *Testschritten* wird der Ablauf der Testausführung beschrieben. Jeder Testschritt enthält eine Anweisung an den Tester, welche Aktion er durchzuführen hat. Falls nötig, können Bilder und Dokumente zu jedem Testschritt hinzugefügt werden.



Abbildung 2.2. Testschritt

Die Beschreibung des Testschrittes umfasst folgende Bestandteile:

Aktion	Die textuelle Beschreibung der Aktion, die der Tester auszu- führen hat.
Vorbedingung	Der Zustand, in dem sich das Testsystem befinden soll, bevor die Aktion beginnt.
Erwartetes Ergebnis	Das vorausgesagte Verhalten des Testsystems oder der Kom- ponente, basierend auf der Spezifikation.
Nachbedingung	Der erwartete Zustand des Testsystems, nachdem die Aktion durchgeführt wurde.

2.4.1. Testschrittergebnis

Klaros-Testmanagement führt den Tester schrittweise durch die Testausführung. Jeder Schritt wird automatisch mitprotokolliert und durch den Tester mit einem **Testschrittergebnis** versehen.

Das Testschrittergebnis enthält die folgenden Informationen:

Ergebnis	Bestanden: Der Test konnte wie vorgesehen beendet wer- den.
	Fehlschlag: Die Überprüfung liefert nicht das erwartete Ergebnis.
	Fehler: Die Überprüfung konnte nicht durchgeführt werden.
	Unklar: Die Überprüfung liefert kein eindeutiges Ergebnis
	Übersprungen: Der Testschritt wurde ausgelassen.
Zusammenfassung (optional)	Eine kurze Zusammenfassung des Testschrittergebnisses.
Beschreibung (optional)	Eine Beschreibung des Testschrittergebnisses.
Dauer	Die automatisch mitprotokollierte Ausführungsdauer des ein- zelnen Testschrittes.

2.4.2. Automatisierte Testfälle

Automatisierte Testfälle haben in der Regel keine Testschritte, da sie von einem automatisierten Testwerkzeug (z.B. JUnit) ausgeführt werden. Die Testergebnisse werden in Form einer Ergebnisdatei zur Verfügung gestellt, die in das System importiert wird. Liegen sowohl manuelle als auch automatisierte Testergebnisse vor, können sie gemeinsam ausgewertet werden.

2.5. Testsegment - Testschritte wiederverwenden

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Ein *Testsegment* ist eine vordefinierte Abfolge von Testschritten. Es wird als eigenes Objekt erstellt und in Testfälle eingefügt.





Häufig kommen bestimmte Testschritte oder auch Abfolgen von Testschritten in mehreren Testfällen exakt gleich vor. Um sicherzustellen, dass diese auch identisch sind, können die Schritte in Testsegmente gekapselt und zentral verwaltet, bearbeitet und gespeichert werden.

Die Testschritte eines Testsegments haben die gleichen Attribute wie die Testschritte eines Testfalls.

2.6. Testsuite - Gruppieren von Testfällen

Testfälle können zu *Testsuiten* gruppiert werden, um gemeinsam in einem Testlauf nacheinander ausgeführt zu werden.





Ein Testfall kann in mehreren Testsuiten vorkommen. Er wird lediglich referenziert, sodass sich Änderungen am Testfall automatisch auf alle Testsuiten auswirken, die diesen Testfall enthalten.

2.7. Testsystem - Das zu testende Objekt

Das *Testsystem* ist die Komponente oder das System, welches getestet wird. Meist ist dies eine Softwareversion, kann aber auch ein bestimmtes Gerät oder andere Hardware sein. Ein Testfall kann erst ausgeführt werden, wenn ein Testsystem definiert ist.

2.8. Testumgebung - Der äußere Einfluss auf das Testergebnis

Die *Testumgebung* stellt die äußeren Bedingungen dar, die auf das Testsystem einwirken. Diese kann aus Hardware oder Software bestehen, wie z.B. eine verwendete Browserversion oder das Betriebssystem, in dem die Software des Testsystems installiert wurde. Ein Testfall kann erst ausgeführt werden, wenn eine Testumgebung definiert ist.

2.9. Testlauf - Die Ergebnisse der ausgeführten Testfälle

Ein *Testlauf* wird automatisch erstellt, sobald damit begonnen wird einen Testfall oder eine Testsuite auszuführen. Er ist der verwendeten Testumgebung und dem verwendeten Testsystem zugeordnet und sammelt alle Ergebnisse, die während der Ausführung des Testfalls oder der Testsuite anfallen. Die Testläufe dokumentieren somit die gesamten Testaktivitäten.



Abbildung 2.5. Testlauf mit Testergebnissen

Testläufe können pausiert und später fortgesetzt werden. Das Ergebnis (Bestanden, Fehlschlag, Fehler, Unklar, Übersprungen) erscheint erst nach vollständiger Ausführung aller Testfälle des Testlaufs in der Auswertung. Bis dahin wird der Testlauf mit seinen gesammelten Ergebnissen nicht angezeigt.

2.9.1. Protokollierte Parameter eines Testlaufs

Während eines Testlaufs werden die folgenden Parameter automatisch mitprotokolliert und können zur Auswertung herangezogen werden:

Das verwendete Testsystem

Die verwendete Testumgebung

Der ausgeführte Testfall/Testsuite

Das Ausführungsdatum

edition

Der ausführende Benutzer

2.10. Anforderung - Eigenschaften des Testsystems

^{se} Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Eine *Anforderung* ist eine vorgegebene, zu erfüllende Bedingung, die die gewünschten Eigenschaften des Testsystems beschreibt. Um sicherzustellen, dass das Testsystem diese Eigenschaften erfüllt, werden Testfälle definiert und diese den Anforderungen zugeordnet, die die Testfälle überprüfen. So kann nachvollzogen werden, welche Anforderungen durch Tests abgedeckt sind (*Abdeckung*) und welche davon mit dem Ergebnis *Bestanden* ausgeführt wurden (*Konformität*).

Aus diesen Informationen wird automatisch der Stand der Abdeckung und Konformität für das gesamte Projekt oder ausgewählte Objekte ermittelt, sobald die ersten Testergebnisse vorliegen.

2.10.1. Attribute für Anforderungen

Name

Priorität (Hoch/Mittel/Niedrig)

Kurzbeschreibung

Ausführliche Beschreibung

2.11. Iteration - Das Projekt in Phasen unterteilen

Senterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Eine *Iteration* repräsentiert eine ausgewählte Phase eines Projekts. Sie kapselt eine Untermenge von Objekten, einschließlich aller während dieser Phase ausgeführten Aufgaben und Testergebnisse. Eine solche Gruppierung der Objekte erleichtert es, verschiedene Testzyklen in einem Projekt zu identifizieren und ermöglicht eine bessere Integration des Testprozesses, speziell mit agilen Softwareentwicklungspraktiken.

In einer Iteration werden die betrachteten Testsysteme, Testumgebungen und Anforderungen innerhalb eines Projekts für einen bestimmten Zeitraum festgelegt.

2.12. Aufgabe - Planen der Testaktivität

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Mittels einer *Aufgabe* werden Testaktivitäten geplant und koordiniert. Dazu wird die Ausführung eines Testfalls oder einer Testsuite in der Aufgabe festgelegt.

Nachdem eine Aufgabe erstellt wurde, kann sie einem Benutzer zugeordnet und von ihm ausgeführt und bewertet werden. Aufgaben können mehrfach wiederholt und ausgeführt werden, sodass ihnen mehrere Testläufe zugeordnet sein können. Aufgaben lassen sich hierarchisch anordnen und Abhängigkeiten von Aufgaben untereinander können spezifiziert werden.



Abbildung 2.6. Aufgabe

Zusätzlich zur Testausführung sind weitere Aufgabentypen vorgesehen. Diese dienen zum Überprüfen von Testfällen, Planen von Anforderungen sowie zur Formulierung von beliebigen anderen Aufgaben in Textform.

Die Aufgaben können verschachtelt und den Benutzern individuell zugeordnet werden. Dadurch kann der Fortschritt von größeren, verteilten Testaktivitäten einfach verfolgt werden.

2.13. Benutzer - Benutzer und ihre Rollen

Klaros-Testmanagement bietet vier verschiedene Benutzerrollen, die beim Anlegen des Benutzers vergeben werden und bei Bedarf später geändert werden können: *Administrator, Manager, Tester* und *Gast*.

Administrator	Ein Administrator hat stets Zugriff auf alle Funktionen in Kla- ros-Testmanagement. Dazu gehören neben den testbezoge- nen Aktivitäten auch die Benutzerverwaltung, Backups und Wiederherstellung sowie systemweite Einstellungen.
Testmanager	Testmanager haben vollständigen Lese- und Schreibzugriff auf alle Daten eines Projekts. Sie können Objekte anlegen und löschen sowie Projekteinstellungen verwalten. Dazu gehört auch das Anlegen, Bearbeiten und Löschen von Benutzern mit der Rolle "Tester" oder "Gast".
Tester	Tester können Tests ausführen und Testergebnisse und Issu- es bearbeiten und Berichte anzeigen. Darüber hinaus haben sie nur Leserechte.
Gast	Die Rolle "Gast" ist z.B. für Projektmanager, Kunden oder Reviewer gedacht. Der Gast hat vollständigen, aber nur lesen- den Zugriff auf alle Daten und kann Berichte anzeigen, abspei- chern und in andere Formate exportieren.

Die globalen Benutzerrollen gelten für alle vorhandenen Projekte. Bei Bedarf kann jedoch die Rolle eines Benutzers in jedem Projekt individuell vergeben werden.

2.13.1. Projektspezifische Benutzerrollen

Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Zusätzlich zur globalen Zuordnung können die Benutzerrollen pro Projekt neu vergeben oder Benutzer von einzelnen Projekten ausgeschlossen werden. Der Benutzer behält dabei seine globale Rolle, erhält aber im ausgewählten Projekt erweiterte oder eingeschränkte Rechte oder auch keinen Zugriff.

Dies ist z.B. sinnvoll beim Einbinden von externen Mitarbeitern, die nur in einem bestimmten Projekt aktiv sind. Andererseits kann ein Tester in bestimmten Projekten Testmanager-Rechte benötigen und die entsprechende Zuordnung erhalten.

Detaillierte Informationen über das Rechte- und Rollensystem finden sich in Appendix A.



2.14. Übersicht über die Objekte



2.15. Issue - Anlegen und Beheben von Fehlern

Klaros-Testmanagement lässt sich mit Issue-Tracking-Systemen wie Jira, Redmine, GitLab und vielen anderen integrieren (siehe <u>Section 12.5</u>). Wird in einem Testschritt ein Fehler gefunden, kann er direkt aus Klaros-Testmanagement heraus an das angebundene Issue-Management-System übergeben werden, ohne die Anwendung verlassen zu müssen.

Der Eintrag wird automatisch in beiden Systemen verknüpft. Änderungen im Issue-Tracker werden im Hintergrund regelmäßig mit Klaros-Testmanagement synchronisiert. Ein automatisch erzeugter Rückverweis (Backlink) im Issue-Management-System führt mit einem Klick wieder zum Testfall in Klaros-Testmanagement zurück.

Kapitel 3. Installation

Dieses Kapitel enthält alle Informationen, die zur Installation von Klaros-Testmanagement erforderlich sind.

Klaros-Testmanagement stellt Installationsprogramme für Microsoft Windows- und Linux-basierte Betriebssysteme zur Verfügung. Im Installationsprogramm sind alle zusätzlichen Programme enthalten, die für die erste Inbetriebnahme von Klaros-Testmanagement benötigt werden. Diese umfassen die dateibasierte Datenbank (Apache Derby), den Apache Tomcat-Anwendungsserver und eine geeignete Java-Laufzeitumgebung.



Einsatzbereite Docker-Images für verschiedene Datenbanken sind ebenfalls verfügbar.

Diese finden Sie zusammen mit der ausführlichen Dokumentation unter https://github.com/klaros-testmanagement/klaros-docker.



Datenbank nicht für Produktionsumgebungen geeignet

Das Installationsprogramm konfiguriert Klaros-Testmanagement für die Verwendung der mitgeliefert Apache-Derby-Datenbank. Diese ist nicht für den produktiven Einsatz vorgesehen!

Für Produktionssysteme wird dringend empfohlen, eine Datenbank wie MariaDB, MySQL, PostgreSQL oder SQL-Server zu verwenden, da diese eine deutlich höhere Leistung bieten. Die Datenbank kann im Anschluss nach der Erstinstallation geändert werden (siehe <u>Abschnitt 3.11, "Die Standard-Datenbank ändern"</u>).

3.1. Lizenzmodell für die Community und Enterprise Edition

Das Lizenzmodell ist abhängig von der verwendeten Klaros-Testmanagement-Edition. Die Klaros-Testmanagement Community Edition bietet im Vergleich zur Klaros-Testmanagement Enterprise Edition zwar eine eingeschränkte Menge an Funktionen, begrenzt aber nicht die Anzahl der Benutzer, die mit der Software arbeiten können.

Bei der Klaros-Testmanagement Enterprise Edition bestimmt die erworbene Lizenz die Anzahl der aktiven Benutzer, die mit der Software arbeiten können.

3.2. Systemvoraussetzungen

Klaros-Testmanagement ist eine webbasierte Anwendung. Client- und die Serverseite haben daher unterschiedliche Systemvoraussetzungen, die in den nächsten Abschnitten ausführlich beschrieben werden.

3.2.1. Client-Voraussetzungen

Für den Zugriff auf Klaros-Testmanagement ist ein moderner Webbrowser, der JavaScript unterstützt, erforderlich.

3.2.1.1. Unterstützte Browser

- Apple Safari
- Google Chrome
- Microsoft Edge
- Mozilla Firefox

X

Microsoft Internet Explorer - Hinweis auf begrenzte Unterstützung

Die Unterstützung für den Microsoft Internet Explorer wird in kommenden Versionen schrittweise eingestellt. Zurzeit ist die Grundfunktionalität weiterhin verfügbar, es kann jedoch zu Darstellungsproblemen kommen.

Klaros-Testmanagement ist für eine minimale Bildschirmauflösung von 1280x960 Pixel entworfen. Full-HD (1920x1080) ist die empfohlene Auflösung.

3.2.2. Server-Voraussetzungen

Zum Betreiben von Klaros-Testmanagement unter Microsoft Windows oder Linux ist ein 64-Bit-Betriebssystem auf Intel/AMD x64 erforderlich.

Name	Version
Microsoft Windows	Windows 10, Windows 11, Windows Server 2012 Windows Server 2012 R2, Windows Server 2016, Windows Server 2019, Windows - Server 2022
CentOS, Red Hat - Enterprise Linux (RHEL)	8.x, 9.x
SuSE	SUSE Linux Enterprise Server (SLES) 15
Ubuntu	20.04, 22.04, 24.04
Debian	bullseye, bookworm
Andere Linux Distributionen	Von anderen Linux-Distributionen ist zu erwarten, dass sie ohne - Probleme funktionieren, solange die mitgelieferte 64-Bit-Java-11 Laufzeitumgebung ausführbar ist. Wir geben allerdings keine - Konformitätserklärung zu diesen Distributionen ab.

Tabelle 3.1. Unterstützte Betriebssysteme

Für eine optimale Leistung wird die Verwendung einer separaten Datenbankinstallation dringend empfohlen. Diese Datenbank muss dabei nicht auf derselben physischen Maschine wie der Klaros-Testmanagement-Server laufen.

Name	Version
Apache Derby	10.15.1.3 (standardmäßig mitgeliefert und vorkonfiguriert)

Name	Version
Microsoft SQLServer	SQLServer 2008, 2008 R2, 2012, 2014, 2016, 2017, 2019, 2022
MySQL	Version 5.5, 5.6, 5.7, 8.0
MariaDB	ab Version 10.3
PostgreSQL	ab Version 9.4

Tabelle 3.2. Unterstützte Datenbanksysteme

3.2.2.1. Microsoft Windows

Um Klaros-Testmanagement unter Microsoft Windows zu betreiben, müssen folgende Anforderungen erfüllt sein:

Mindestanforderungen

2 GB RAM, 3 GHz Dual Core CPU

Empfohlene Anforderungen (Remote-Datenbank)

4 GB RAM, 3 GHz Quad Core CPU

• Empfohlene Anforderungen (Lokale Datenbank)

8 GB RAM, 3 GHz Quad-Core CPU

• Betriebssystem

Microsoft Windows Server 2012, Microsoft Windows Server 2012 R2, Microsoft Windows Server 2016, Microsoft Windows Server 2019, Microsoft Windows Server 2022, Microsoft Windows 10, Microsoft Windows 11.

3.2.2.2. Linux

Um Klaros-Testmanagement unter Linux zu betreiben, müssen folgende Anforderungen erfüllt sein:

Mindestanforderungen

2GB RAM, 2GHz Dual-Core CPU

• Empfohlene Anforderungen (Remote-Datenbank)

4GB RAM, 3GHz Quad-Core CPU

• Empfohlene Anforderungen (Lokale Datenbank)

8GB RAM, 3GHz Quad-Core CPU

• Betriebssystem

Linux Intel64/x86-64 (Intel 64-Bit-Architektur)-Distribution mit glibc Version 2.17 oder besser und Version 2.2.1 oder besser des GTK+-Widget Toolkits und der zugehörigen Bibliotheken.



GTK+ Bibliotheken nur im Fenstermodus benötigt

Die GTK+-Bibliotheken werden nur zum Ausführen des Klaros-Testmanagement Installationsprogramms im Fenstermodus benötigt. Das Ausführen des Installationsprogramms im Konsolenmodus funktioniert ohne diese Anforderung.

Einschränkung der Anzahl offener Dateien



Achtung

Klaros-Testmanagement speichert Datenbank-Suchindizes im Verzeichnis .klaros/indices im Home-Verzeichnis des Benutzerkontos, auf dem der Anwendungsserver Klaros-Testmanagement läuft. Während des Indizierungsprozesses kann die Menge der offenen Dateien im Betriebssystem Linux die vordefinierte Grenze überschreiten

Der Befehl **ulimit -n** zeigt die konfigurierte Anzahl der erlaubten offenen Dateien an, welche einen Wert von mindestens 4096 enthalten sollte. Bitte lesen Sie in Ihrer Systemdokumentation nach, wie Sie diesen Wert einstellen können (Beispielsweise durch Bearbeiten der Datei /etc/security/limits.conf).

3.2.2.3. Verfügbarer Festplattenspeicher

Klaros-Testmanagement speichert binäre Anhänge in einem dateibasierten Repository innerhalb des Verzeichnisses .klaros im Home-Verzeichnis des Benutzerkontos, in dem der Klaros-Testmanagement Anwendungsserver läuft.

Je nach Nutzungsverhalten sollte eine ausreichende Menge an Speicherplatz für dieses Konto reserviert werden. Wenn die dateibasierte Derby-Datenbank in Betrieb ist, befinden sich die Datenbankdateien ebenfalls in diesem Verzeichnis.



Empfohlener verfügbarer Festplattenspeicher

Die Reservierung von 10 GB Festplattenspeicher sollte für typische Verwendungszwecke ausreichen.

3.3. Installation

Unter Microsoft Windows wird der Installer mit dem Start von Klaros-Testmanagement-<version>-Setup.exe aufgerufen.

Unter Linux wird der Installer mit dem Aufruf von **Klaros-Testmanagement-<version>-Setup.bin** in einer User-Shell gestartet.

Die folgenden Bildschirmansichten zeigen Schritt für Schritt die Installation von Klaros-Testmanagement.

3.3.1. Sprachauswahl

Beim Start erfragt das Installationsprogramm die Sprache, die während des folgenden Installationsvorgangs verwendet wird.

Gegenwärtig sind Englisch und Deutsch als Sprache verfügbar.

💦 Language Selection	\times
Please select your language	
Deutsch	~
ОК	

Abbildung 3.1. Sprachauswahl

3.3.2. Schritt 1: Willkommen

Der erste Schritt zeigt dem Benutzer einen Begrüßungsbildschirm (Abbildung 3.2).





Die Installation kann abgebrochen werden, indem Sie auf den Button Beenden klicken. Wenn Sie auf Weiter klicken, wird die Installation fortgesetzt.
3.3.3. Schritt 2: Informationen

Im zweiten Schritt werden Informationen über das Produkt und die Änderungshistorie angezeigt, wobei die behobenen Issues und die neu hinzugefügten Funktionen aufgelistet werden. (<u>Abbildung 3.3</u>).

💦 IzPack - In	stallation von Klaros-Testmanagement 5.1.0-SNAPSHOT	_	\times
Informati	ionen 🌙	klaro	
Schritt 2	! von 10	NUIUS)
	Bitte lesen Sie die folgenden Informationen:		
	Release Notes for Klaros-Testmanagement version 5.1.0-SNAPSH	OT:	^
	Klaros-Testmanagement is a professional, web-based test management	gement software.	
	It offers integrated components for planning, execution and test activities and enables cross-team communication.	evaluation of	
	With interfaces to more than 50 tools, including issue track, test automation tools and servers for continuous integration.	ing systems, , it is easy	
	to adapt to existing environments.		
16	The system offers numerous pre-defined reports and enables the user-defined reports according to individual requirements.	he creation of	
*			
Q2)			
88			
	<	2	. `
(Erstellt mit Iz	Pack - http://izpack.org/)		
Hilfe	Zurück	Weiter Beender	ı

Abbildung 3.3. Die Seite "Informationen"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Willkommen* und mit Weiter wird die Installation fortgesetzt.

3.3.4. Schritt 3: Lizenzvereinbarung

Der dritte Schritt zeigt den Lizenzvertrag für Klaros-Testmanagement an. Die Lizenz muss akzeptiert werden, um die Installation fortsetzen zu können. (<u>Abbildung 3.4</u>).



Abbildung 3.4. Die Seite "Lizenzvereinbarungen"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Informationen* und mit Weiter wird die Installation fortgesetzt.

3.3.5. Schritt 4: Installationspfad

Im vierten Schritt wird der Installationspfad abgefragt, auf dem Klaros-Testmanagement installiert werden soll. Mit dem Button Auswählen kann nach dem Pfad im lokalen Dateisystem gesucht werden (<u>Abbildung 3.5</u>).



Speicherort des Installationsordners

Es wird dringend empfohlen, den vorgeschlagenen Installationsort und **nicht** das Windows-Verzeichnis "Programme" zu verwenden. Standardmäßig kann Klaros-Testmanagement von dort aus nicht gestartet werden, da der Tomcat-Applikationsserver Schreibzugriff auf den Installationsordner benötigt. Dies wird durch Microsoft Windows User Account Control (UAC) blockiert.

Wenn Sie Klaros-Testmanagement an diesem Ort installieren möchten, muss UAC deaktiviert sein.

👬 IzPack - Ir	nstallation von Klaros-Testmanagement 5.1.0-SNAPSHOT	– 🗆 X
Installati	ionspfad	V klaros
Schritt 4	i von 10	
	Wählen Sie den Installationsnfad-	
		Auswählen
-	c; josers julie (valios-resultanagement	
1	C. psets (manipular os-resultanagenen)	
	C, josets (Language of Standardgenerit	
() () () ()	C, josets (La light os-resultariagenent	
8 8 8 8		
(Erstellt mit Iz Hilfe	:Pack - http://izpaok.org/)	Zurück Weiter Beenden

Abbildung 3.5. Die Seite "Installationspfad"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Lizenzvereinbarung* und mit Weiter wird die Installation fortgesetzt.

3.3.6. Schritt 5: Auswahl Installationspakete

Im fünften Schritt werden die optionalen Pakete ausgewählt, die mit Klaros-Testmanagement installiert werden.

📸 IzPack - Installation von Klaros-Testmanagement 5.1.0-SNAPSHOT	– 🗆 X
Auswahl Installationspakete Schritt 5 von 10	👱 klaros
Bitte wählen Sie hier die Installationspakete aus: Hinweis: Die grau markierten Pakete können nicht optional ausgewählt werden. Image: State	243,72 МВ 30,41 МВ 334,8 МВ 22,12 МВ
Beschreibung Das Java Server Runtime Environment - Version 11.0.9.1.1. Erforderlicher Speicherplatz: Verfügbarer Speicherplatz: (Erstellt mit IzPack - http://izpack.org/)	631,04 MB 105,55 GB Zurück Weiter Beenden
(Erstellt mit izPack - http://izpack.org/) Hilfe	Zurück Weiter Beenden

Abbildung 3.6. Die Seite "Auswahl Installationspakete"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Installationspfad* und mit Weiter wird die Installation fortgesetzt.

3.3.7. Schritt 6: Benutzerdaten

Im sechsten Schritt können die Einstellungen für den installierten Tomcat-Applikationsserver geändert werden.

💦 IzPack - Installation v	von Klaros-Testmanagement 5.1.0-SNAPSHOT	– 🗆 X
Benutzerdaten Schritt 6 von 10	2	klaros
	Tomcat Server Einstellungen	
T	Server-Port: 18005	
	HTTP-Port: 18080	
	Session Timeout: 60	
	Das Minimum und Maximum des verfügbaren Speichers für den Tomcat Prozess:	
	Minimum (MB): 512	
	Maximum (MB): 1024	
	Der Benutzername und das Passwort für den Administrator-Account des Tomcat Servers	:
10	Benutzername: admin	
	Password:	
	Retype Password:	
	Anwendungsserver starten	
	✓ Browser öffnen	
(Erstellt mit IzPack - http://	//izpack.org/)	
Hilfe	Zurück	Weiter Beenden

Abbildung 3.7. Die Seite "Benutzerdaten"

Hier können die von Tomcat verwendeten Ports (Server-Port und HTTP-Port) eingestellt werden sowie die minimale und maximale Speichermenge die dem Tomcat-Prozess zur Verfügung steht. Außerdem wird hier der Benutzernamen und das Kennwort des Tomcat-Anwendungsserver-Administrators vergeben.

Die Anwendung kann auf Wunsch im Anschluss an die Installation direkt gestartet werden. Wenn die Checkbox *Anwendungsserver starten* gesetzt ist, wird der Anwendungsserver automatisch gestartet. Wenn die Checkbox *Browser öffnen* gesetzt ist, wird ein Webbrowser-Fenster mit der Anmeldeseite der Anwendung geöffnet. Beide Aktionen finden während der Phase <u>Abschnitt 3.3.9</u>, <u>"Schritt 8: Externe Prozesse ausführen"</u> statt.

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Auswahl Installationspakete* und mit Weiter wird die Installation fortgesetzt.

3.3.8. Schritt 7: Installation

Der siebte Schritt beginnt mit der Installation von Klaros-Testmanagement und zeigt deren Fortschritt. Die folgenden Screenshots zeigen die Bildschirme *Installationsfortschritt* und *Installation beendet* (<u>Abbildung 3.8</u>).

🚮 IzPack - Installation von Klaros-Testmanagement 5.1.0-SNAPSHOT	- 🗆 X
Installation	
Schritt 7 von 10	
Installationsfortschritt des Pakets: C: Users \talal¥Garos-Testmanagement \webapps\klaros-web.war Klaros-Testmanagement 5.1.0-SN/ Installationsfortschritt gesamt: 3 4	PSHOT
(Erstellt mit IzPack - http://izpack.org/) Hilfe	Zurück Weiter Beenden
Abbildung 3.8. Die Seite "Installat	ion"
🛐 IzPack - Installation von Klaros-Testmanagement 5.1.0-SNAPSHOT	– 🗆 X
Installation Schritt 7 von 10	👱 klaros
Installationsfortschritt des Pakets:	
[Fertig]	
4/4	
(Erstellt mit IzPack - http://izpack.org/) Hilfe	Zurück Weiter Beenden

Abbildung 3.9. Die Seite "Installation beendet"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Weiter wird die Installation fortgesetzt.

3.3.9. Schritt 8: Externe Prozesse ausführen

Der optionale achte Schritt führt externe Prozesse wie das Setzen der Umgebungsvariablen und den Start des Tomcat-Anwendungsservers aus.

💦 IzPack - Ir	nstallation von Klaros-Testmana	gement 5.1.0-SNAPSHOT	– 🗆 X
Externe F Schritt 8	Prozesse ausführen 3 von 10	2	klaros
		Prozesse	
		4/4	
	Using CATALINA_BASE:	"C:\Users\talal\Klaros-Testmanagement	n
	Using CATALINA_HOME:	"C:\Users\talal\Klaros-Testmanagement	"
	Using CATALINA_TMPDI	<pre>IR: "C:\Users\talal\Klaros-Testmanagement</pre>	\temp"
	Using JRE_HOME:	"C:\Users\talal\Klaros-Testmanagement	\jre"
	Using CLASSPATH:	"C:\Users\talal\.klaros\resources;C:\	Users\talal\Klaros-T
	Using CATALINA_OPTS:	" -Doak.locksupport=disabledadd-op	ens=java.base/java.l
16			
*			
- 🎭 -			
8			
	<		>
(Erstellt mit Iz	Pack - http://izpack.org/)		
Hilfe		Zurück	Weiter Beenden

Abbildung 3.10. Die Seite "Prozesse"

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Beim Klick auf den Button Weiter wird die Installation fortgesetzt.

3.3.10. Schritt 9: Verknüpfungen einrichten

Im neunten Schritt werden die Optionen für Verknüpfungen gesetzt. Das Installationsprogramm kann Verknüpfungen im Startmenü und auf dem Desktop erstellen (<u>Abbildung 3.11</u>).

💦 IzPack - Ir	nstallation von Klaros-Testmanagement 5.1.0-SNAPSHOT		– 🗆 X
Verknüpf Schritt 9	f ungen einrichten 9 von 10		👱 klaros
	 Verknüpfungen im Start-Menü erstellen Zusätzliche Verknüpfungen auf dem Desktop erstellen Wählen Sie eine Programmgruppe für die Verknüpfung: (Default) 7-Zip Accessibility A	~	Erstelle Verknüpfung für:) aktuellen Benutzer () alle Benutzer
(Erstellt mit Iz Hilfe	[Klaros Testmanagement zPack - http://izpack.org/]		Standard Zurück Weiter Beenden

Abbildung 3.11. Die Seite "Verknüpfungen einrichten"

Dieser Schritt sieht auf Linux-Installationen etwas anders aus, hat aber dieselbe Funktion.

Die Installation kann mit einem Klick auf den Button Beenden abgebrochen werden. Mit Zurück gelangen Sie zurück zum Bildschirm *Externe Prozesse ausführen* und mit Weiter wird die Installation fortgesetzt.

3.3.11. Schritt 10: Installation abgeschlossen

Der letzte Schritt benachrichtigt den Benutzer, dass Klaros-Testmanagement erfolgreich installiert wurde und zeigt den Pfad zum Deinstallationsprogramm (<u>Abbildung 3.12</u>).



Abbildung 3.12. Die Seite "Installation abgeschlossen"

Der Button Automatisches Installationsskript generieren speichert ein komplettes Skript der Installation mit den ausgewählten Benutzeroptionen, sodass dieselbe Installation unbeaufsichtigt oder auf anderen Rechnern wiederholt werden kann.

Mit einem Klick auf den Button Fertig wird die Installation abgeschlossen.

3.4. Konsolen-basierte Installation

Der Installer unterstützt auch eine GUI-lose Variante, die nur ein Shell-/Befehlsfenster benötigt und per -- -console aufgerufen wird.

Dieser Installer zeigt den Inhalt in aufeinanderfolgenden Panels im Shell-Fenster an und lässt den Benutzer seine Auswahl über die Tastatur eingeben.

Die Buttons werden durch kurze Tastenfolgen ersetzt:

Enter	zeigt die nächste Seite an oder akzeptiert einen Standardwert
1+Enter	wählt die Checkbox aus
0+Enter	wählt die Checkbox ab
n+Enter	wählt das n-te Element aus einer Liste aus

3.4.1. Schritt 1: Sprachauswahl

Das Panel *Sprachauswahl* (<u>Abschnitt 3.3.1, "Sprachauswahl"</u>) wird angezeigt und mit **0+Enter** für Englisch oder **1+Enter** für Deutsch bestätigt.

```
Select your language
0 [x] eng
1 [] deu
Enter selection:
```

3.4.2. Schritt 2: Willkommen

Das Willkommen-Panel (<u>Abschnitt 3.3.2, "Schritt 1: Willkommen"</u>) wird angezeigt und wird mit **1+Enter** bestätigt.

```
Information
Version 5.5.5
(c) Copyright 2009-2023 verit Informationssysteme GmbH. All Rights Reserved.
Press 1 to continue, 2 to quit, 3 to redisplay
```

3.4.3. Schritt 3: Informationen

Das Panel *Informationen* (<u>Abschnitt 3.3.3, "Schritt 2: Informationen"</u>) zeigt die Versionshinweise der aktuellen Versionen an.



3.4.4. Schritt 4: Lizenzvereinbarung

Das Panel *Lizenzvereinbarung* (<u>Abschnitt 3.3.4, "Schritt 3: Lizenzvereinbarung</u>") zeigt die Lizenzvereinbarung an. Diese muss mit **1+Enter** bestätigt werden, um mit der Installation fortzufahren.

Licensing Agreements Limited Use Software License Agreement PLEASE READ THIS AGREEMENT CAREFULLY BEFORE ATTEMPTING TO USE THE SOFTWARE OF verit Informationssysteme GmbH AND RELATED DOCUMENTATION (COLLECTIVELY, THE "SOFTWARE") AND BEFORE INSTALLING OR USING THE SOFTWARE. BY INSTALLING OR USING THE SOFTWARE, YOU ARE CONSENTING TO BE BOUND BY THIS AGREEMENT....

```
limitations may not apply to you. You may also have other rights that vary from state to state.
Press 1 to accept, 2 to reject, 3 to redisplay
```

3.4.5. Schritt 5: Installationspfad

Das Panel *Installationspfad* (<u>Abschnitt 3.3.5, "Schritt 4: Installationspfad</u>") sieht mit dem Konsolen-Installationsprogramm wie folgt aus:



Mit **Enter** können Sie den vorgeschlagenen Verzeichnispfad akzeptieren oder einen anderen eingeben. Das Verzeichnis wird erstellt, falls er noch nicht vorhanden war. Falls das Verzeichnis bereits existiert, wird eine Warnung angezeigt und Sie können die Installation abbrechen oder fortsetzen.

3.4.6. Schritt 6: Auswahl Installationspakete

Die Auswahl von Installationspaketen (<u>Abschnitt 3.3.6, "Schritt 5: Auswahl Installationspakete"</u>) ist im Konsolenmodus komplexer zu handhaben. Die erforderlichen Pakete (Java, Tomcat und Klaros-Testmanagement) werden lediglich zur Information angezeigt. Die Eingabe von **Y** wählt die Installation der PDF-Dokumentation aus, während **N** diese abwählt. Anschließend werden Sie aufgefordert, die Eingabe mit **1** und **Enter** zu bestätigen.

```
Select Installation Packages

Select the packs you want to install:

[x] Pack 'Java Runtime Environment' required

[x] Pack 'Tomcat 9 Application Server' required

[x] Pack 'Klaros-Testmanagement required

[x] Include optional pack 'PDF-Documentation'

Enter Y for Yes, N for No: N

Y

Done!

Press 1 to continue, 2 to quit, 3 to redisplay

1
```

3.4.7. Schritt 7: Benutzerdaten

Danach können die Benutzerdaten angepasst werden (<u>Abschnitt 3.3.7, "Schritt 6: Benutzerda-</u> <u>ten"</u>). Das Installationsprogramm zeigt Standardwerte an, die durch Drücken von **Enter** übernommen oder einem anderen Wert überschrieben werden können. Die Checkboxen zeigen [x], wenn sie gesetzt sind, andernfalls []. Durch Drücken von **Enter** wird der aktuelle Wert übernommen, der mit **1** bzw. **0** explizit gesetzt oder zurückgesetzt werden kann.

In diesem Beispiel übernehmen wir die meisten der Standardwerte. Nur die Speichereinstellungen und das Tomcat-Admin-Passwort (ohne Voreinstellung) werden eingegeben und die beiden Checkboxen werden nicht gesetzt:

```
User Data
Tomcat Server Settings:
Server-Port: [18005]
HTTP-Port: [18080]
Session Timeout: [60]
The minimum and maximum amount of memory available to the Tomcat process:
Minimum (MB): [512]
400
Maximum (MB): [1024]
1024
The username and the password of the administrator account for managing the Tomcat server:
Username: [admin]
Password:
*******
Retype Password:
*******
 [ ] Launch Application Server
Enter 1 to select, 0 to deselect:
0
 [ ] Launch Browser
Enter 1 to select, 0 to deselect:
0
Press 1 to continue, 2 to quit, 3 to redisplay
1
```

3.4.8. Schritt 8: Installation

Die Pakete werden angezeigt, sobald sie installiert sind (Abschnitt 3.3.8, "Schritt 7: Installation").

```
[ Starting to unpack ]
[ Processing package: Java Runtime Environment (1/4) ]
[ Processing package: Tomcat 9 Application Server (2/4) ]
[ Processing package: Klaros-Testmanagement (3/4) ]
[ Processing package: PDF-Documentation (4/4) ]
[ Unpacking finished ]
Installation finished
```

3.4.9. Schritt 9: Externe Prozesse ausführen

Dieses Panel wird bei der Konsolen-basierten Installation übersprungen. Ansonsten wären hier Startmeldungen des Servers zu sehen. Unter <u>Abschnitt 3.3.9, "Schritt 8: Externe Prozesse ausführen"</u> sehen Sie, wie dies im GUI-Modus gehandhabt wird.

3.4.10. Schritt 10: Verknüpfungen einrichten

In einer GUI-losen Umgebung sind in der Regel keine Menüeinträge oder Icons auf dem Desktop nötig, daher wählen wir **N**. <u>Abschnitt 3.3.10, "Schritt 9: Verknüpfungen einrichten"</u> erläutert das Einrichten von Shortcuts unter Windows.



3.4.11. Schritt 11: Installation beenden

Hier kann ein automatisches Installationsskript mit den gewählten Benutzereingaben erstellt werden, sodass dieselbe Installation unbeaufsichtigt oder auf anderen Rechnern wiederholt werden kann. Automatische Installationsskripte funktionieren auch für unbeaufsichtigte Installationen im Konsolenmodus.

```
Generate an automatic installation script
Enter Y for Yes, N for No:
Y
Select the installation script (path must be absolute)[C:\Users\...\Klaros-Testmanagement
\auto-install.xml]
Installation was successful
Application installed on C:\Users\...\Klaros-Testmanagement
[ Writing the uninstaller data ... ]
[ Console installation done ]
```

Nach dem Beenden der Installation befindet sich Klaros-Testmanagement im Verzeichnis /opt/ klaros-testmanagement. In <u>Abschnitt 3.12</u>, <u>"Als System-Service installieren"</u> finden Sie weitere Informationen wie Klaros-Testmanagement als System-Dienst eingerichtet werden kann, um die Anwendung beim Systemstart automatisch auszuführen.

3.5. Automatische Installations-Skripte

Wurde bei der Klaros-Installation ein automatisiertes Installationsskript gespeichert, kann das Programm mit den gleichen Parametern erneut installiert werden. In diesem Beispiel wird angenommen, dass das automatisierte Installationsskript unter auto-install.xml gespeichert wurde.

Linux: Klaros-Testmanagement-<version>-Setup.bin -- -console auto-install.xml

Windows: Klaros-Testmanagement-<version>-Setup.exe auto-install.xml

Sie können auch das folgende Beispielskript verwenden. Bitte passen Sie die Einstellung für installpath und die Felder entry unter userInput an. Die Installation kann einige Warnungen des Formulars ausgeben AutomationHelper-Klasse für Panel nicht gefunden ..., die gefahrlos ignoriert werden können.

```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<AutomatedInstallation langpack="eng">
 <com.izforge.izpack.panels.htmlhello.HTMLHelloPanel id="hello" />
 <com.izforge.izpack.panels.htmlinfo.HTMLInfoPanel id="info" />
 <com.izforge.izpack.panels.licence.LicencePanel id="license" />
 <com.izforge.izpack.panels.target.TargetPanel id="target">
   <installpath>/opt/Klaros-Testmanagement</installpath>
 </com.izforge.izpack.panels.target.TargetPanel>
 <com.izforge.izpack.panels.packs.PacksPanel id="packs">
   <pack index="0" name="Java Runtime Environment" selected="true"/>
   <pack index="1" name="Tomcat 9 Application Server" selected="true" />
   <pack index="2" name="Klaros-Testmanagement" selected="true" />
   <pack index="3" name="PDF-Documentation" selected="false" />
 </com.izforge.izpack.panels.packs.PacksPanel>
 <com.izforge.izpack.panels.userinput.UserInputPanel id="portselection">
   <userInput>
     <entry key="tomcat_server_port" value="18005" />
     <entry key="tomcat_http_port" value="18080" />
     <entry key="tomcat_https_port" value="18443" />
     <entry key="tomcat_session_timeout" value="60" />
     <entry key="tomcat_memory_min" value="256" />
     <entry key="tomcat_memory_max" value="1024" />
     <entry key="tomcat_admin_name" value="tcadmin" />
     <entry key="tomcat_admin_password" value="IfYouDontChangeThisDontBlameMe" />
     <entry key="start_server_flag" value="false" />
     <entry key="launch_browser_flag" value="false" />
   </userInput>
 </com.izforge.izpack.panels.userinput.UserInputPanel>
 <com.izforge.izpack.panels.install.InstallPanel id="install" />
 <com.izforge.izpack.panels.process.ProcessPanel id="process" />
 <com.izforge.izpack.panels.shortcut.ShortcutPanel id="shortcuts">
   <createMenuShortcuts>false</createMenuShortcuts>
   <programGroup>Klaros Testmanagement</programGroup>
   <createDesktopShortcuts>false</createDesktopShortcuts>
   <createStartupShortcuts>false</createStartupShortcuts>
 </com.izforge.izpack.panels.shortcut.ShortcutPanel>
 <com.izforge.izpack.panels.finish.FinishPanel id="finish" />
</AutomatedInstallation>
```

Beispiel 3.1. Beispiel auto-install.xml

3.6. Starten der Anwendung

Die Klaros-Testmanagement Anwendung wird innerhalb eines Tomcat-Applikationsservers ausgeführt. Dieser kann von der Kommandozeile aus unter Windows mit dem folgenden Aufruf gestartet werden:

<install-dir>/bin/startup.bat

Unter Linux lautet dieser:

<install-dir>/bin/startup.sh

Falls bei der Installation die Erzeugung von Startmenü-Einträgen oder Desktop-Shortcuts angewählt war, können diese ebenfalls zum Starten von Klaros-Testmanagement verwendet werden.

3.7. Stoppen der Anwendung

Ein laufender Tomcat-Applikationsserver kann unter Windows mit dem folgenden Aufruf von der Kommandozeile gestoppt werden:

<install-dir>/bin/shutdown.bat

Unter Linux lautet dieser:

<install-dir>/bin/shutdown.sh

Falls bei der Installation die Erzeugung von Startmenü-Einträgen oder Desktop-Shortcuts angewählt war, können diese ebenfalls zum Stoppen von Klaros-Testmanagement verwendet werden.

3.8. Zugang zur Anwendung

Um auf Klaros-Testmanagement zuzugreifen, öffnen Sie die folgende URL in einem Webbrowser: http://\${host}:\${port}/klaros-web/ wobei \${host} die IP-Adresse oder der Domänenname des Anwendungsservers und \${port} der HTTP-Port ist, der in <u>Abschnitt 3.3.7, "Schritt 6: Benutzerdaten"</u> definiert ist.



Standard-URL (lokal)

Unter der Annahme, dass die Standard-Installationsparameter verwendet wurden, kann auf Klaros-Testmanagement lokal über die URL http://localhost:18080/kla-ros-web/ zugegriffen werden.

3.9. Updates einspielen

Klaros-Testmanagement kann einfach durch die Installation der neuen Version aktualisiert werden. Bestehende Einstellungen und Daten werden dabei automatisch übernommen. Weitere Informationen über den Installationsvorgang finden Sie unter <u>Abschnitt 3.3, "Installation"</u>. Das Klaros Home-Verzeichnis bleibt unangetastet, sodass alle Einstellungen, die Datenbankkonfiguration und das Content-Repository vom Aktualisierungsprozess unberührt bleiben.



Wichtig

Der Tomcat-Applikationsserver muss vor Beginn der Aktualisierung gestoppt werden. Andernfalls kann der Installationsprozess nicht erfolgreich abgeschlossen werden.



Sichern Sie Ihre Daten vor der Aktualisierung!

Es wird dringend empfohlen, vor Beginn des Aktualisierungsprozesses ein Backup sowohl des Klaros Home-Verzeichnis als auch Ihrer Datenbank zu erstellen.

Die meisten Datenbankprodukte bieten Dienstprogramme zur Sicherung von Datenbankschemata an. Bitte entnehmen Sie die Vorgehensweise den entsprechenden Abschnitten in der Dokumentation des Datenbankanbieters.

Falls Sie noch die vorkonfigurierte Apache-Derby-Datenbank verwenden, sind hierzu keine weiteren Maßnahmen erforderlich und die Sicherung des Klaros Home-Verzeichnisses ist ausreichend.

3.10. Wichtige Dateiverzeichnisse

Es gibt zwei wichtige Verzeichnisse einer Klaros-Testmanagement Installation. Eines ist das **Installationsverzeichnis**, in dem der Tomcat-Anwendungsserver und die Webanwendung installiert werden. Das Andere ist das **Klaros Home-Verzeichnis**, das Laufzeitdaten wie Konfigurationsdateien, Suchindizes und Anhänge enthält. Standardmäßig hat dieses Verzeichnis den Namen . klaros und befindet sich im Home-Verzeichnis des Benutzers, der Klaros-Testmanagement ausführt.

Um ein einfaches Upgrade zu ermöglichen, ändert der Installations-/Upgrade-Prozess nur Dateien im Installationsverzeichnis und lässt den Inhalt im Klaros Home-Verzeichnis unangetastet.

Auf Microsoft Windows Systemen befindet sich das Klaros Home-Verzeichnis standardmäßig unter C:\Users\<username>\.klaros. Dies kann auch durch Eingabe von %UserProfile%\.klaros in der Adressleiste eines Explorer-Fensters oder durch Eingabe von cd %UserProfile%\.klaros in der Befehlszeile erreicht werden.

Auf Linux-Systemen kann dieses Verzeichnis über den Pfad ~/.klaros gefunden werden und befindet sich in der Regel unter /home/<username>/.klaros.



Das Klaros Home-Verzeichnis verschieben

Durch Setzen der Umgebungsvariablen KLAROS-HOME vor dem Start der Anwendung kann der Speicherort des Klaros Home-Verzeichnisses an einen anderen Ort verschoben werden.

export KLAROS_HOME=/var/application-data/klaros



Vorsicht beim Verschieben des Klaros Home-Verzeichnisses!

Wenn die Anwendung keine frühere Installation im Klaros Home-Verzeichnis findet, wird eine leere Standard-Installation einschließlich einer neuen lokalen Datenbank erstellt. Geschieht dies versehentlich, z.B. durch Angabe eines ungültigen Klaros Home-Verzeichnisses, so scheint es, dass die bisherigen Daten verschwunden sind.

Dies ist natürlich nicht der Fall. Um diese Situation zu lösen, setzen Sie die Variable KLAROS-HOME auf den richtigen Wert und starten Sie neu.

3.10.1. Log-Dateien

Die vollständigen Log-Dateien für Klaros-Testmanagement befinden sich im Verzeichnis logs im Installationsverzeichnis und in einer komprimierten Form im Verzeichnis logs im Klaros Home-Verzeichnis.

Das Anhängen dieser Protokolldateien an einen Fehlerbericht hilft dem Support-Team, auftretende Probleme schnell zu identifizieren und zu beheben.

3.10.2. Datenbankeinstellungen

Die Datei .klaros/hibernate.properties wird verwendet, um die Datenbank anzugeben, die für Klaros-Testmanagement verwendet werden soll. Sie definiert die verwendete Treiberklasse, die URI unter der die Datenbank erreichbar ist sowie die benötigten Authentifizierungsinformationen. Weitere Informationen zur Änderung der Datenbankeinstellungen finden Sie unter <u>Abschnitt 3.11</u>, <u>"Die Standard-Datenbank ändern"</u>.

3.10.3. Sprachunterstützung

Die Benutzeroberfläche von Klaros-Testmanagement kann durch Hinzufügen der Unterstützung für weitere Sprachen angepasst werden. Dieser Prozess wird in <u>Abschnitt 4.1.1, "Sprachdateien</u> <u>definieren"</u> beschrieben. Die Sprachdateien sind unter .klaros/resources/messages zu finden.

3.10.4. Die Datei "quotes.txt"

Klaros-Testmanagement stellt auf der Login-Seite ein täglich wechselndes *Zitat des Tages* dar. Diese Zitate sind in der Datei .klaros/resources/quotes.txt gespeichert. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 10.4.4</u>, <u>"Oberfläche"</u> und <u>Abschnitt 4.2</u>, <u>"Zitat des Tages"</u>.

3.10.5. Die Derby-Datenbank

Wenn Sie die voreingestellte Derby-Datenbank verwenden, befinden sich deren Dateien im Verzeichnis .klaros/klarosDerby.

3.10.6. Das Content-Repository

Alle Anhänge und benutzerdefinierten Berichte werden im Verzeichnis .klaros/content-Repository gespeichert. Werden die Klaros-Testmanagement-Installation und die Datenbank an einen anderen Ort migriert **muss** sichergestellt werden, dass dieses Verzeichnis ebenfalls kopiert wird.

3.11. Die Standard-Datenbank ändern

Initial verwendet Klaros-Testmanagement die Apache-Derby-Datenbank, die keine weitere Installation oder Konfiguration erfordert, aber im Vergleich zu anderen Datenbanken eine erheblich geringere Leistung aufweist. Für den produktiven Einsatz wird daher dringend empfohlen, ein leistungsfähigeres Datenbanksystem zu verwenden, wie z.B. die Open Source-Datenbankserver PostgreSQL oder MariaDB/MySQL oder das kommerzielle Datenbankprodukt Microsoft SQLServer.

Um zu einem anderen Datenbanksystem zu wechseln, muss Klaros-Testmanagement gestoppt werden und die Datei hibernate.properties, die sich in <user.home>/.klaros/hibernate.properties befindet, muss wie unten gezeigt bearbeitet werden.

Die Eigenschaft hibernate.connection.url muss mit dem Standort der Datenbank in Ihrem Netzwerk übereinstimmen. Bitte wenden Sie sich an Ihren Datenbankadministrator um die Zugangsdaten für hibernate.connection.username und hibernate.connection.password zu erhalten.



Warnung

Leerzeichen am Ende von Zeilen in der Datei hibernate.properties könnten von hibernate falsch interpretiert werden! Zum Beispiel ist 'hibernate.connection.password=root ' nicht dasselbe wie 'hibernate.connection.password=root' (Anführungszeichen ignorieren).

Eine ausführliche Liste aller Parameter finden Sie im Hibernate Core Manual.

3.11.1. MariaDB

Ändern Sie für MariaDB den Inhalt der Datei in:

MariaDB 10.3 oder höher:

```
hibernate.connection.driver_class=org.mariadb.jdbc.Driver
hibernate.connection.url=jdbc:mariadb://localhost:3306/klaros
hibernate.connection.username=root
hibernate.connection.password=root
```

3.11.2. Microsoft SQL-Server

Um einen Microsoft SQL-Server zu verwenden, ändern Sie den Inhalt der Datei in:

Microsoft SQL-Server 2008 oder nachfolgende Versionen:

```
hibernate.connection.driver_class=com.microsoft.sqlserver.jdbc.SQLServerDriver
hibernate.connection.url=jdbc:sqlserver://localhost:1433;databaseName=KLAROS
hibernate.connection.username=root
hibernate.connection.password=root
```

3.11.3. MySQL

Ändern Sie für MySQL den Inhalt der Datei in:

MySQL 5.5 oder nachfolgende Versionen:

```
hibernate.connection.driver_class=com.mysql.jdbc.Driver
hibernate.connection.url=jdbc:mysql://localhost:3306/klaros
hibernate.connection.username=root
hibernate.connection.password=root
```



Wichtig

Wenn Sie eine MySQL-Datenbank verwenden, ist es wichtig, die folgende Option in der Datei my.ini zu setzen.

```
# The maximum size of a query packet the server can handle as well as
# maximum query size server can process (Important when working with
# large BLOBs). enlarged dynamically, for each connection.
max_allowed_packet = 64M
```

3.11.4. PostgreSQL

Um PostgreSQL zu verwenden, ändern Sie den Inhalt der Datei in:

PostgreSQL 9.4 oder nachfolgende Versionen:

```
hibernate.connection.driver_class = org.postgresql.Driver
hibernate.connection.url = jdbc:postgresql://localhost/klaros
hibernate.connection.username=root
hibernate.connection.password=root
```

3.11.5. Apache Derby

Zurückschalten auf die integrierte Derby-Datenbank:

```
hibernate.connection.driver_class=org.apache.derby.jdbc.EmbeddedDriver
hibernate.connection.url=jdbc\:derby\:${user.home}/.klaros/klarosDerby;create\=true
hibernate.connection.username=root
hibernate.connection.password=root
```

3.11.6. Einrichten der Datenbank-Instanz



Datenbank-Instanz erzeugen

Klaros-Testmanagement legt nicht automatisch die Datenbankinstanz (klaros im obigen Beispiel) oder den Datenbankbenutzer (Benutzer root mit Kennwort root im obigen Beispiel) auf dem Datenbankserver an.

Das Anlegen einer Datenbankinstanz und das Hinzufügen eines Benutzers sind im entsprechenden Datenbankhandbuch beschrieben und werden hier nicht behandelt. Der Datenbankbenutzer benötigt Berechtigungen zum Erstellen, Löschen und Ändern von Tabellen, um die Klaros-Testmanagement Datenbankinstanz ordnungsgemäß zu starten.



Unterstützung für den UTF-8 Schriftsatz sicherstellen!

Je nach Datenbank kann es erforderlich sein, die Unterstützung für UTF-8-Zeichensätze manuell zu aktivieren, was für die Unterstützung von Sprachen mit ungewöhnlichen Zeichensätzen erforderlich sein kann. Insbesondere ältere Versionen von MySQL und MariaDB sind dafür bekannt, dass sie nur eine begrenzte Zeichenunterstützung in ihrer Standardkonfiguration haben. Das folgende Beispiel zeigt, wie dies durch Editieren der Konfigurationsdatei mysql.cnf aktiviert werden kann.

```
character-set-server=utf8
collation-server=utf8_general_ci
```

Beispiel 3.2. Zusätzliche Einträge in mysql.cnf um UTF-8 Character Sets zu unterstützen

3.12. Als System-Service installieren

Für den dauerhaften Einsatz von Klaros-Testmanagement ist es unerlässlich, dass die Anwendung direkt nach einem Neustart des Rechners, auf dem sie installiert ist, verfügbar ist. Je nach Betriebssystem müssen dazu unterschiedliche Maßnahmen ergriffen werden.

3.12.1. Als Linux-Service installieren

Abhängig von Ihrem Linux-Distributionssystem werden Dienste über verschiedene Systeme gestartet.

Wir stellen hier Beispielskripte für die gebräuchlichsten Daemons zur Verfügung: SysVInit und systemd.

3.12.1.1. Als SysVInit-Service installieren

Um Klaros-Testmanagement automatisch zu starten, sollte das unten gezeigte Startskript im Verzeichnis /etc/init.d als Datei klaros gespeichert und für die gewünschten Run-Level via **chkconfig** aktiviert werden. Einzelheiten entnehmen Sie bitte der Dokumentation Ihrer Linux-Distribution.

Die Werte der Felder KLAROS_USER und KLAROS_PATH müssen zunächst durch die vorgesehene Benutzerkennung und den Installationspfad ersetzt werden. Es ist wichtig, dass das Skript ausführbar ist. Dies kann mit dem Befehl **chmod a+x klaros** sichergestellt werden.

Der Service kann jetzt über **service klaros start** gestartet und über **service klaros stop** gestoppt werden. **chkconfig --add klaros** ermöglicht den Start des Dienstes beim Systemstart.

```
#!/bin/bash
#
#
#
# This will be ignored by systems that don't use chkconfig.
# chkconfig: 345 98 2
# description: klaros (inside Tomcat, a servlet container)
#
### BEGIN INIT INFO
# Provides: klaros
```

-	<pre># Required-Start:</pre>	
	# Required-Stop:	
-	# Default-Start:	3 4 5
-	# Default-Stop:	0 1 2 6
	<pre># Short-Description:</pre>	klaros
	<pre># Description:</pre>	klaros-testmanagement
	### END INIT INFO	
	#	
1	KLAROS_USER=klarosder	no
l	KLAROS_PATH="/opt/kla	aros-testmanagement"
ļ	/bin/su - "\${KLAROS_l	JSER}" -c "\${KLAROS_PATH}/bin/catalina.sh \$@ >/var/tmp/klaros.log 2>&1"

3.12.1.2. Als systemd-Service installieren

Zum automatischen Starten von Klaros-Testmanagement sollte das unten gezeigte Startskript im /etc/systemd/system Verzeichnis als klaros.service gespeichert werden.

Die Werte der Einträge User, ExecStart und ExecStop müssen zunächst durch die vorgesehene Benutzerkennung und den Installationspfad ersetzt werden.

Der Dienst kann nun über **systemctl start klaros** gestartet und über **systemctl stop klaros** gestoppt werden. **systemctl enable klaros** ermöglicht den Start des Dienstes beim Systemstart.

```
[Unit]
Description=Klaros Testmanagement Service
After=syslog.target network.target
[Service]
Environment="JRE_HOME=/opt/klaros-testmanagement/jre"
Environment="PATH=/opt/klaros-testmanagement/jre/bin:/usr/local/sbin:/usr/local/bin:/usr/sbin:/usr/bin"
Type=forking
User=klaros
ExecStart=/opt/klaros-testmanagement/bin/startup.sh
ExecStop=/opt/klaros-testmanagement/bin/shutdown.sh
[Install]
WantedBy=multi-user.target
```

3.12.2. Als Windows-Service installieren

Unter der Windows-Betriebssystemfamilie kann das mitgelieferte Skript **service.bat** verwendet werden, um den Tomcat-Anwendungsserver als Windows-Dienst zu installieren.

Bitte wenden Sie sich an die offizielle Apache Tomcat 9 Windows service HOW-TO für weitere Einzelheiten.



Ändern des Service Benutzerkontos

Standardmäßig wird der erstellte Windows-Dienst das Konto LocalSystem verwenden, um die Anwendung Klaros-Testmanagement auszuführen. Diese Einstellung muss in den Anmelde-Einstellungen des Services in das Konto des Benutzers geändert werden, der Klaros-Testmanagement ausführen soll. Diese Änderung ist erforderlich, da sich der Standardspeicherort für das Verzeichnis .klaros im Home-Verzeichnis des jeweiligen Benutzerkontos befindet.

Das Auslassen dieses wichtigen Schritts führt zu einem neu erstellten .klaros Verzeichnis in C:\Windows\System32\systemuser (dieser Speicherort kann je nach verwendeter Windows-Version variieren; neuere Versionen verwenden stattdessen C:\Windows\system32\config\systemprofile). Der Dienst verwendet dann die neu erstellte leere Datenbank, anstelle der beim manuellen Start von Klaros-Testmanagement verwendeten.

In diesem Fall beenden Sie den Dienst, löschen das Verzeichnis .klaros in Windows\System32, ändern die Benutzerkontoeinstellungen des Dienstes und starten den Dienst neu.



Speichereinstellungen und Java Optionen anpassen

Abhängig von der Datenbankgröße und der Anzahl der parallelen Benutzer kann Klaros-Testmanagement eine größere Menge an Speicher benötigen, als es das Installationsprogramm mit den angegebenen Standardwerten vorschlägt. Wenn Sie die angegebenen Speichereinstellungen für eine bereits installierte Installation ändern möchten, können Sie diese Einstellungen wie unten gezeigt im **tomcat9w** Administrations-Interface vornehmen.

Zusätzlich müssen die folgenden Java 9 Optionen gesetzt werden um den reibungslosen Einsatz der Software zu gewährleisten:

--add-opens=java.base/java.lang=ALL-UNNAMED

--add-opens=java.base/java.io=ALL-UNNAMED

--add-opens=java.base/java.util=ALL-UNNAMED

--add-opens=java.base/java.util.concurrent=ALL-UNNAMED

--add-opens=java.base/java.net=ALL-UNNAMED

--add-opens=java.rmi/sun.rmi.transport=ALL-UNNAMED

Um sicherzustellen, dass die Einstellungen des installierten Dienstes tatsächlich geändert werden, fügen Sie den Dienstnamen als Befehlszeilenargument wie folgt hinzu: [Pfad]\Klaros-Testmanagement\bin\Tomcat9w.exe //ES//Tomcat9.

neral Les On Lensier	lava Ctartur	Chutdaum			
eneral Log On Logging	g suvu startup	Shutdown			
Use default					
Java Virtual Machine:					
C: \Program Files \Eclipse Adoptium \jdk-21.0.3.9-hotspot \bin \server					
Java Classpath:					
C:\Program Files\Apach	e Software Foundat	tion\Tomcat 9.0\bin\b	ootstrap		
Java Options:					
-Djava.util.logging.conf	fig.file=C:\Program F	Files (Apache Software	age e Fou		
-Djava.util.logging.man -Djava.util.logging.conf Java 9 Options:	fig.file=C:\Program F	Files Apache Software	e Fou		
-Djava.util.logging.man -Djava.util.logging.con Java 9 Options: add-opens=java.bas add-opens=java.bas add-opens=java.bas	e/java.lang=ALL-UN e/java.lang=ALL-UN e/java.util=ALL-UNN e/java.util=ALL-UNN	INAMED AMED AMED INAMED INAMED			
-Djava.util.logging.man -Djava.util.logging.conf Java 9 Options: add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas	e/java.lang=ALL-UN e/java.lang=ALL-UN e/java.io=ALL-UNN/ e/java.util=ALL-UNN/ e/java.util.concurre	INAMED AMED AMED INAMED MAMED MB			
-Djava.util.logging.man -Djava.util.logging.conf Java 9 Options: add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas	e/java.lang=ALL-UN e/java.lang=ALL-UN e/java.io=ALL-UNN e/java.util=ALL-UNN e/java.util=ALL-UNN e/java.util.concurre 256	INAMED AMED AMED AMED AMED MEALL-UNNAMED MB			
-Djava.util.logging.man -Djava.util.logging.conf Java 9 Options: add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas add-opens=java.bas Initial memory pool: Maximum memory pool: Thread stack size:	e/java.lang=ALL-UN e/java.lang=ALL-UN e/java.io=ALL-UNN e/java.util=ALL-UNN e/java.util.concurre 256	INAMED AMED VAMED INAMED MB MB KB			

Abbildung 3.13. Zusätzliche Tomcat Speichereinstellungen und Java Optionen



Weitere Einstellungen

Das Content-Repository (siehe <u>Abschnitt 3.10.6, "Das Content-Repository"</u>) benötigt eine weitere Einstellung im **tomcat9w** Administrations-Interface. Hier muss der Parameter **-Doak.locksupport=disabled** hinzugefügt werden.

eneral	Log On	Logging	Java	Startup	Shutdowr		
	e default						
lava V	Java Virtual Machine:						
[C:(U	C:\Users\stolp\Kiaros-Testmanagement\jre\bin\server\jvm.dll						
Java (Classpath:						
C:\U	sers\stolp\	Klaros-Te:	stmanage	ement\bin'	bootstrap.;	ar;C:\Us	ers\stolp\
Java	Options:						
-Djav -Djav -Djav -Doa	/a.io.tmpd /a.util.logg /a.util.logg k.locksupp	ir=C:\Use jing.mana jing.config ort=disab	rs\stolp\l ger=org, l.file=C:\ led	apache.ju Users\stol	tmanageme li.ClassLoad p\Klaros-Te	ntitemp lerLogMa stmanag	emen
Java 9	9 Options:						
Java 9 ado ado ado	9 Options: 1-opens=ja 1-opens=ja 1-opens=ja	ava.base/ ava.base/ ava.rmi/su	java.lang java.io=, in.rmi.tra)=ALL-UNI ALL-UNNA ansport=A	Named Med Ll-Unname	D	< >
Java 9 ado ado ado Initial	9 Options: J-opens=ja J-opens=ja J-opens=ja memory po	ava.base/ ava.base/ ava.rmi/su pol:	java.lang java.io=, in.rmi.tra 512	j=ALL-UNI ALL-UNNA ansport=A	NAMED MED LL-UNNAME	D MB	< >
Java 9 add add add Initial Maxim	9 Options: d-opens=ja d-opens=ja d-opens=ja memory po num memor	ava.base/ ava.base/ ava.rmi/su pol: [y pool: [java.lang java.io=, in.rmi.tra 512 1024	j=ALL-UNI ALL-UNNA Insport=A	NAMED MED LL-UNNAME	D MB MB	< >
Java 9 ado ado ado Initial Maxim Threa	9 Options: d-opens=ja d-opens=ja d-opens=ja memory po num memor d stack size	ava.base/ ava.base/ ava.rmi/su pol: [y pool: [e: [java.lang java.io= n.rmi.tra 512 1024	g=ALL-UNI ALL-UNNA ansport=A	NAMED MED LL-UNNAME	D MB MB KB	< >

3.13. Monitoring

Durch die eingebaute JMX-Unterstützung in Tomcat ist es möglich, die Klaros-Testmanagement Installation mit jedem JMX-fähigen Monitoring-Tool (wie z.B. VisualVM) zu überwachen.

Fügen Sie dazu die folgenden Zeilen zu Klaros-Testmanagement/bin/catalina.sh hinzu:

```
# Set JMX connector
RMI_SERVER='[FQDN]'
JMXREMOTE="-Dcom.sun.management.jmxremote"
JMXREMOTE="$JMXREMOTE -Dcom.sun.management.jmxremote.port=[PORT]"
JMXREMOTE="$JMXREMOTE -Dcom.sun.management.jmxremote.ssl=false"
JMXREMOTE="$JMXREMOTE -Dcom.sun.management.jmxremote.authenticate=false"
JMXREMOTE="$JMXREMOTE -Dcom.sun.management.jmxremote.ssl=false"
JAVA_OPTS="$JAVA_OPTS $JMXREMOTE -Djava.rmi.server.hostname=$RMI_SERVER"
```

[FQDN] und [PORT] bezeichnen den voll qualifizierten Domänennamen und Port des JMX-Servers, dem Sie die Daten übermitteln.

3.14. Externe Issue-Management-Systeme konfigurieren

Abhängig vom verwendeten Issue-Management-System kann eine zusätzliche Konfiguration erforderlich sein, um es mit Klaros-Testmanagement zu verbinden. Dieser Abschnitt beschreibt die erforderlichen Konfigurationsschritte für jedes System.

Name	Version
Bugzilla	3.0.9 oder höher
GitHub	Jede
GitLab	API V3/V4
Jira	3.7 oder höher
Mantis	1.1.8 oder höher
Redmine	3.0 oder höher
Тгас	0.10 oder höher
YouTrack	2021.3 oder höher

Tabelle 3.3. Unterstützte Externe Issue-Management-Systeme

3.14.1. Jira konfigurieren

Für aktuelle Jira Versionen sind keine Maßnahmen zur Konfiguration notwendig.



Jira-Cloud-Instanzen

Seit dem 25. April 2019 hat Atlassian den Zugang zu ihren Cloud-Instanzen auf Tokenbasierte Authentifizierung beschränkt, sodass die Authentifizierung über Benutzername und Passwort nicht mehr unterstützt wird.

Klaros-Testmanagement unterstützt ebenfalls das Token-basierte Authentifizierungsschema. Verwenden Sie hierzu einfach Ihre E-Mail-Adresse als Benutzernamen und das Token als Passwort.

3.14.1.1. Jira Legacy Server (Version 4.2 oder niedriger)

Um die Verbindung zu einer Jira Legacy Server-Instanz zu erlauben, muss das Jira RPC-Plugin in Ihrer Jira-Installation aktiviert werden. Sie finden diese Option unter *Accept remote API calls* in **General Configuration** unter **Global Settings**. Dort aktivieren Sie das JIRA RPC Plugin in **Plugins** unter **System** im seitlichen Menü.

3.14.2. Redmine konfigurieren



REST Web Service aktivieren

Um sich mit einer Redmine Installation verbinden zu können, muss die Checkbox *Authentication Required* und die Checkbox *Enable REST Web Service* im Abschnitt Authentifizierung der Redmine Administrationseinstellungen aktiviert sein (<u>Abbildung 3.15, "Der Redmine-Authentifikations-Bereich"</u>).

Abbildung 3.15. Der Redmine-Authentifikations-Bereich

3.14.3. Trac Konfiguration

Die Trac Integration erfordert die Installation des TracXMLRPC-Plugins. Die Download-Archive und Installationshinweise sind unter http://trac-hacks.org/wiki/XmlRpcPlugin/ zu finden. Nach erfolgreicher Installation des Plugins muss dieses über die Web-Administrationsschnittstelle von Trac aktiviert werden. <u>Abbildung 3.16, "Trac-Plugin-Bereich"</u> Nach der erfolgreichen Installation des Plugins erscheint der Bereich für das Trac-Plugin. <u>Abbildung 3.16, "Trac-Plugin. Bereich"</u> zeigt den Abschnitt Trac-Plugin nach der erfolgreichen Installation des Plugins.

Administration	Manage Plugins (2)	
General Basic Settings Logging Permissions	Install Plagic File: Datei ausgawähten Kaine Datei ausgawäht Install Upload a plugin packaged as Python egg.	
Plugins	▶ Trac 1.4.2	
Ticket System Components Milestones	Integrated SCM, wiki, insue tracker and project environment	
Priorities Resolutions Severities Ticket Types Versions Version Control	TracXMLRPC 1.1.9 IPC Interface to Trac Aufor Mantatare Add Smons Monse Monses Monses	
Repositories	Component [+] [-]	🖾 Enabled
	▶ tracrps.api.* - License: BSD	
	XNLRECSystem — Core of the RPC system.	
	▶ tracrpc.json_rpc.* - License: BSD	
	▶ JsonRpcProtocol - Example 'POST' request using 'curl' with 'Content-Type' header and body:	
	▶ tracrpo.xml_rpo.* - License: BSD	
	▶ XmlRpcProtocol — There should be XML-RPC client implementations available for all popular programming languages. Example call using `curl':	
	A	pply changes

Abbildung 3.16. Trac-Plugin-Bereich

Um das Plugin zugänglich zu machen, müssen authentifizierte Benutzer auf das XML_RPC-Plugin zugreifen können.

integrated SCM & Project Manage	perment			WIKI TIMELINE R	logged in as user Logo OADMAP VIEW TICKETS	Nut Preferences Help/Guide About Trac NEW TICKET SEARCH ADMIN
The subject authenticated h	ias been granted th	e permission XML_RPC.				close
Administration	Manage P	ermissions and G	roups			
General Basic Settings Logging Permissions	Grant Permis Action: XML	sion: RPC v S	ubject: authenticated	Add Grant permise	sion for an action to a subject, v	which can be either a user or a group.
Plugins	Subject:	Targ	et:	Add Copy all of subject	's permissions to target. Subje	ct and target can be either users or groups.
Components Milestones Priorities Resolutions	Add Subject (o Group:	ıp:	Add Add a user or grou	p to an existing permission gro	up.
Severities Ticket Types Versions	Permissions					
Version Control	Subject	Action				
Repositories	anonymous	BROWSER_VIEW REPORT_SQL_VIEW TIMELINE_VIEW	CHANGESET_VIEW REPORT_VIEW WIKI_VIEW	FILE_VIEW ROADMAP_VIEW	LOG_VIEW	MILESTONE_VIEW TICKET_VIEW
	authenticated	TICKET_CREATE	TICKET_MODIFY	WIKI_CREATE	WIKI_MODIFY	XML_RPC
	user	TRAC_ADMIN				

Abbildung 3.17. Trac-Permission-Bereich

Die folgende Kombination wurde erfolgreich getestet:

• Trac 0.12 / TracXMLRPC 1.1.0

Spätere Versionen werden voraussichtlich funktionieren, sind aber derzeit nicht getestet.

3.15. SSL-Unterstützung

Ein Hinweis zu selbstsignierten Zertifikaten: Wenn Sie auf HTTPS-Server mit selbstsignierten Zertifikaten zugreifen, müssen Sie diese Zertifikate in Ihren Java trusted keystore importieren. Eine Anleitung dazu finden Sie z.B. unter http://stackoverflow.com/questions/2893819/telling-java-toaccept-self-signed-ssl-certificate.

3.16. Upgrade von Version 4

Dieser Abschnitt beschreibt wichtige Änderungen für Benutzer, die ein Upgrade von Version 4 auf Version 5 durchführen. Bitte lesen Sie diesen Abschnitt vollständig durch, bevor Sie ein Upgrade durchführen. Unter support@verit.de können Sie uns bei Fragen zum Upgrade-Prozess kontaktieren.

3.16.1. Erforderliche Version für das Upgrade

Für ein erfolgreiches Upgrade auf Version 5 ist es zwingend erforderlich, dass die installierte Klaros-Testmanagement Version 4.12.0 (veröffentlicht am 29. Juli 2019) oder höher ist. Ist dies nicht der Fall, verweigert die Anwendung den Start und Sie sehen einen entsprechenden Eintrag in der Konsole und im Protokoll. In diesem Fall laden Sie bitte zuerst die neueste verfügbare Version 4.xx herunter, installieren Sie diese und aktualisieren Sie Ihre Installation, indem Sie sich mit den Zugangsdaten des Administrators anmelden. Danach ist das Upgrade auf Version 5 möglich.

3.16.2. Unbeaufsichtigtes Datenbank-Update

In Version 4 war es obligatorisch, dass ein Administrator ein Datenbank-Update auslöst, nachdem eine neue Version installiert wurde. In Version 5 ist dies nicht mehr erforderlich, das System ist nun direkt nach einem Update vollständig nutzbar.

3.16.3. Startup-Zeit für das erste Upgrade auf Version 5

Version 5 baut beim ersten Start bestehende Datenbanken aus früheren Installationen neu auf. Abhängig von der Menge der vorhandenen Daten und der Rechenleistung Ihrer Datenbankinstallation kann dieser Vorgang zwischen einigen Minuten und bis zu einer Stunde dauern.

Bitte reservieren Sie dafür einen angemessenen Zeitrahmen und haben Sie beim Start etwas Geduld. Jeder Schritt des Wiederherstellungsprozesses wird auf der Konsole und/oder der Protokolldatei protokolliert.

3.16.4. Eingebettete Java-Laufzeitumgebung

Im Gegensatz zu vorherigen Versionen ist es nicht mehr erforderlich, eine Java-Laufzeitumgebung vor der Klaros-Installation installiert zu haben. Klaros-Testmanagement wird mit einer Java 11-Laufzeitumgebung ausgeliefert, die automatisch verwendet wird.

3.16.5. Automatische Erkennung der Datenbank-Version

Ab Version 5 wird die verwendete Datenbankversion automatisch erkannt und sollte nicht mehr in der Datei <user.home>/.klaros/hibernate.properties hinterlegt werden. Entfernen Sie bitte die Zeile hibernate.dialect=... aus dieser Datei, bevor Sie das Upgrade durchführen.

3.16.6. Oracle-Datenbank wird nicht mehr unterstützt

In Version 5 wurde die Unterstützung für Oracle-Datenbanken eingestellt. Wenn Sie eine bestehende Oracle-Installation haben und auf ein anderes DBMS migrieren möchten, kontaktieren Sie uns bitte unter support@verit.de für weitere Informationen.

3.17. Deinstallation

Um Klaros-Testmanagement zu deinstallieren wählen Sie *Klaros-Testmanagement Server Uninstaller* aus dem Startmenü (wenn Sie unter Windows arbeiten) oder löschen Sie den Installationsordner manuell.

Wenn Sie eine Installation als Dienst hinzugefügt haben, wie unter <u>Abschnitt 3.12, "Als System-Ser-</u> <u>vice installieren"</u> beschrieben, denken Sie bitte daran, den Windows-Dienst oder das Linux-Init-Skript zu deaktivieren.

👫 Deinstallieren			×		
9 Jetzt werden die installierten Anwendungen entfernt!					
Lösche alle Dateien in C:\Users\stolp\Klaros-Testmanagement					
[Nicht	s]				
Deinstallieren		<u> 8</u> ee	nden		

Abbildung 3.18. Das Deinstallationsprogramm



Klaros Home-Verzeichnis

Genau wie beim Upgrade-Prozess wird bei der Deinstallation das Klaros Home-Verzeichnis (<user.home>/.klaros) nicht berührt.

Nach der Deinstallation sollten Sie dieses Verzeichnis daher manuell entfernen. Wenn Sie aber versehentlich deinstalliert haben, können Sie Ihr System durch eine Neuinstallation sicher wiederherstellen.

Kapitel 4. Anpassungen

In diesem Kapitel wird gezeigt, wie die angezeigte Sprache und die Datei für das "Zitat des Tages" angepasst werden können.

4.1. Sprachdateien aktivieren und deaktivieren

Klaros-Testmanagement wird mit Sprachdateien in deutscher und englischer Sprache ausgeliefert. Die Unterstützung für weitere Sprachen kann bei Bedarf mittels zusätzlicher Sprachdateien hinzugefügt werden. Die entsprechende Sprachdatei muss hierzu im Verzeichnis .klaros/ resources/messages gespeichert werden.

Sprachdateien werden unter unter Konfigurieren / System / Sprachen aktiviert. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 10.4.5.1, "Sprachen aktivieren und deaktivieren"</u>.

4.1.1. Sprachdateien definieren

Für die korrekte Anzeige der hinzugefügten Sprache muss die Sprachdatei ein gültiges Format haben: messages_\${language}_\${COUNTRY}_\${variation}.properties wobei \${language} der aus zwei Buchstaben bestehende ISO 639-1 Code für die Sprache in Kleinbuchstaben, \${COUNTRY} der aus zwei Buchstaben bestehende ISO 3166-Code für das Land in Großbuchstaben und \${varia-tion} ein optionaler, benutzerdefinierbarer Code für die Variation der Sprache (z. B. Dialekt) ist.



Encoding der Sprachdateien

Die Sprachdatei muss eine ISO-8859-1-Kodierung mit Unicode-Escapezeichen verwenden (\uXXXX).

Wenn Sie die Dateien lieber in UTF-8 bearbeiten möchten, können Sie einen Konverter wie **native2ascii** oder **iconv** verwenden, um sie in ein geeignetes Format zu übersetzen.

Nachdem Klaros-Testmanagement zum ersten Mal ausgeführt wurde, werden die Sprachdateien für amerikanisches Englisch, Deutsch und eine Fallback-Sprachdatei in den Ordner .klaros/ resources/messages kopiert. Um eine neue Sprachdatei zu erstellen, sollte der Text aus einer dieser Dateien in eine entsprechend benannte neue Datei kopiert werden. Der Inhalt der Datei besteht aus Schlüssel-Wert-Paaren in folgender Form: key=Dieser Satz ist der Wert. Keiner der Schlüssel darf verändert werden. Die Werte (der Text nach dem '=') wird nun in die Zielsprache übersetzt.

Wenn Klaros-Testmanagement gestartet wird, werden die Standard-Sprachdateien (messages. properties, messages_en_US.properties und messages_de_DE.properties) in den Ordner .klaros/resources/messages kopiert, wobei alle an diesen Dateien vorgenommenen Änderungen überschrieben werden. Es wird daher empfohlen, alle Benutzeränderungen an den Schnittstellen für amerikanisches Englisch und Deutsch in einer Datei mit einer Variation der Dateinamen, z. B. messages_en_US_companyName.properties zu speichern.

Neben dem Kopieren über die Standarddateien führt Klaros-Testmanagement auch alle Änderungen mit benutzerdefinierten Dateien zusammen. Werden Schlüssel-Werte-Paare während einer Aktualisierung mit den Standarddateien zusammengeführt, werden diese ebenfalls am Ende der benutzerdefinierten Sprachdateien hinzugefügt, zusammen mit einem Kommentar, der angibt, wann die Einträge hinzugefügt wurden. Diese Einträge sollten vom Ersteller der benutzerdefinierten Sprachdatei übersetzt werden.

4.2. Zitat des Tages

Klaros-Testmanagement zeigt auf dem Anmeldebildschirm täglich ein neues "Zitat des Tages" an. Die ausgelieferten Zitate können um eigene Einträge erweitert oder deaktiviert werden. Die Zitat-Datei kann in Klaros-Testmanagement auf der Seite *General Settings* direkt in den Ordner . klaros/resources im Home-Verzeichnis des Benutzers hochgeladen und aktiviert werden.

Diese Datei ist ein .txt-Dokument in ISO-8859-1 Kodierung und verwendet pro Zitat eine Zeile. Abschnitt <u>Abschnitt 10.4, "System"</u> beschreibt die Installation dieser Datei. Die jeweiligen Zitate wechseln täglich.

Kapitel 5. Funktionale Übersicht

Dieses Kapitel stellt die Bedienoberfläche und ihre grundlegenden Funktionen vor.

5.1. Anmelden

<u>Abbildung 5.1</u> zeigt den Anmelde-Bildschirm von Klaros-Testmanagement. Zum Anmelden geben Sie den Benutzernamen in das Feld *Benutzername* und das zugehörige Passwort in das Feld *Passwort* ein.



Abbildung 5.1. Die Seite "Anmelden"

In der Datenbank sind drei Benutzerkonten mit den folgenden Rollen vordefiniert:

Administrator	Benutzername: admin / Passwort: admin
Manager	Benutzername: manager / Passwort: manager
Tester	Benutzername: tester / Passwort: tester

Bestätigen Sie die Anmeldung mit dem Button Anmelden .

Eine Beschreibung der Benutzerrollen und den mit der Rolle verbundenen Berechtigungen finden Sie unter <u>Anhang A, Zugriffsberechtigungen</u>.

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Die Klaros-Testmanagement Enterprise Edition unterstützt auch die Authentifizierung mittels eines externen Verzeichnisdienstes (LDAP / Active Directory). Weitere Informationen zum Konfigurieren eines externen Verzeichnisdienst finden Sie unter <u>Abschnitt 10.5.5, "LDAP"</u>.



Option zur automatischen Benutzeranlage

Wird ein Benutzer erstmalig mittels eines externen Verzeichnisdienstes authentifiziert, so kann für diesen Benutzer automatisch ein Benutzerkonto mit der Rolle *Tester* erstellt werden. Die LDAP-Attribute für den Benutzernamen und die E-Mail-Adresse werden in diesem Fall automatisch in das neu erstellte Benutzerkonto übertragen.



Die Sprache bei der Anmeldung festlegen

Standardmäßig wird die Benutzeroberfläche in der Sprache dargestellt, die durch die Spracheinstellung des Server-Betriebssystems vorgegeben ist. Mit einem Klick auf die entsprechende Flagge rechts über dem Textfeld *Benutzername* kann auch eine andere Sprache für die Benutzeroberfläche ausgewählt werden.

Nach dem erfolgreichen Anmelden wird die Seite Projekte angezeigt (Abbildung 5.2).

≡ 👱 K L A R (DS TEST	MANAGEMENT				DE FI	nanz-Tracker		Q x	0 ~	* ~
📝 Definieren	11:45:55 1	Login erfolgreich - Willkommen bei Kla	aros-Testmanageme	ent							+ ×
	Projekte										
🎥 Planen											
	Neu								Speichern	Verwe	erfen
🛱 Ausführen			9 Einträge - Seite 1 vo	on 1 🔣 ┥ 🚺	► N	10 🗸			Alle an	zeigen	& ≡
	ID≑ (Beschreibung 🗢	Iterationen 🖨	Anforderungen 🗢	Testfälle 🗢	Testsuiten 🗘	Testläufe 🗢	Erstellt 🗢	Letzter Zugriff 🖨	Akti	on
🕒 Auswerten	O P00011		0	0	1	0	0	vor etwa einem Jahr	vor 2 Monaten	o 🖻	8 ₪
	O P00010	Test PRV	0	0	5	1	3	vor etwa einem Jahr	vor etwa einem Jahr	0 🖻	8 Ŵ
差 Konfigurieren	O P00009	PRV 2	0	0	1	0	0	vor etwa einem Jahr	vor etwa einem Jahr	് മ്	8 1 1 1
	O P00008	PRV	0	0	3	1	1	vor etwa einem Jahr	vor etwa einem Jahr	0 🗹	8 ŵ
	O P00007	Issue Management Integration	0	0	4	0	0	vor 2 Jahren	vor etwa einem Jahr	o 🖻	8 Ŵ
	• P00005	DE Finanz-Tracker	9	6	24	7	8768	vor 3 Jahren	vor wenigen Sekunden	• 🖻	8 ŵ
	O P00003 8	Printer Tester	0	4	1	0	0	vor 3 Jahren	vor 9 Minuten	്മ്	8 ŵ
	O P00002	Finance Tracker	9	6	24	7	338	vor 3 Jahren	vor 5 Stunden	്മ്	a
	O P00001	FinanceTrackerLocalTester	9	6	26	9	204	vor 3 Jahren	vor 2 Minuten	0 Z	a 🖞
	ID	Beschreibung	Iterationen	Anforderungen	Testfälle	Testsuiten	Testläufe	Erstellt	Letzter Zugriff	Akti	ion
	Neu								Speichern	Verwe	erfen

Abbildung 5.2. Die Seite "Projekte"

5.2. Seitenübersicht

Im Folgenden werden die einzelnen Elemente einer Seite vorgestellt.

5.2.1. Seiten-Menü

Die Einträge im Menü im linken Teil des Bildschirms sind den Aktivitäten in einem Testprojekt nachempfunden. Die einzelnen Bereiche lauten:

Definieren

Im Bereich *Definieren* werden alle Objekte, die für ein Testprojekt benötigt werden erstellt, bearbeitet und verwaltet. Zu den hier verwalteten Objekten zählen das Projekt selbst, Testumgebungen, Testsysteme, Testfälle mit ihren einzelnen Testschritten sowie Testsuiten. In der Klaros-Testmanagement Enterprise Edition kommen noch Iterationen, Anforderungen und Testsegmente hinzu.

- PlanenIm Bereich Planen werden Aufgaben zum Durchführen oder
Überprüfen von Tests erstellt und verwaltet. Aufgaben kön-
nen einzelnen Benutzern zugeordnet und der Fortschritt detail-
liert verfolgt werden. Dieser Bereich ist nur in der Klaros-Test-
management Enterprise Edition enthalten.
- AusführenIm Bereich Ausführen werden Aufgaben, manuelle Testfälle
und Testsuiten ausgeführt. Bevor ein Test ausgeführt werden
kann, müssen ein Testsystem und eine Testumgebung ange-
legt und ausgewählt werden. Klaros-Testmanagement führt
den Tester Schritt für Schritt durch den manuellen Test und
protokolliert die Ergebnisse automatisch mit. Zu jedem Test-
schritt können Ergebnisse und Kommentare hinzugefügt wer-
den. Tests können unterbrochen und zu einem späteren Zeit-
punkt weiter ausgeführt werden.
- AuswertenIm Bereich Auswerten bieten ein konfigurierbares Dashboard
und zahlreiche Berichte einen Überblick über den Stand der
Tests. Dabei stehen sowohl Übersichtsberichte als auch fein-
granulare Informationen zur Verfügung, sodass jede einzelne
Testausführung nachvollzogen und visualisiert werden kann.
- KonfigurierenDer Bereich Konfigurieren ist für administrative Aufgaben
vorgesehen, wie die Systemeinstellungen, die Benutzerver-
waltung, das Erstellen von Berichtsvorlagen, die Integrati-
on von Issue-Management-Systemen, Anforderungsmanage-
ment, LDAP und andere. Auch die Datensicherung und der
Import und Export von Projekten werden hier vorgenommen.

In den folgenden Kapiteln werden die einzelnen Bereiche detailliert beschrieben.

5.2.2. Kopfzeile

5.2.2.1. Suche / Schnellauswahl

Abhängig von der Klaros-Testmanagement Edition, finden Sie rechts oben auf der Seite (siehe <u>Abbildung 5.3</u>) entweder ein Suchfeld für die schnelle Navigation zu Objekten mit einer bestimmten ID oder für eine Volltextsuche im aktuell ausgewählten Projekt.

≡	KLAROS	TEST MANAGEMENT	DE Finanz-Tracker 🔳 📕	Dauerauftrag	Q @ ⊻	. ~

Abbildung 5.3. Das Suchfeld

5.2.2.1.1. Volltextsuche

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Im Feld *Suchen* kann nach Objekt-IDs, Wörtern oder Phrasen gesucht werden, die in Feldern von Objekten enthalten sind. Die für die Suche verwendete Syntax ist auf der Webseite von Apache Lucene beschrieben.

= 📌 KLAR	OS TEST MANAGEMENT •	DE Finanz-Tracker 🛅 📑 Dauerauftrag	Q × 0∕	* ~
📝 Definieren	♀ 06:15:42 Die Suche wurde nach 0.3 Sekunden erfolgreich beendet und lieferte 20 Treffer in 8 Ergebnissen.			+ ×
🏩 Planen	Suchergebnisse			
🏟 Ausführen	Aufgabe (2) Testfall (5) Testfallergebnis (1)			
🕒 Auswerten	1. JOB00355 - Ausführung TC00015 - Einen Dauerauftrag hinzufügen. Zusammenfassung Ausführung TC00015 - Einen <mark>Dauerauftrag</mark> hinzufügen.			
🖋 Konfigurieren	 2. JOB00360 - Ausführung TC00022 - Aktualisierung des Dashboards durch Hinzufügen eines Dauerauftrags. Beschreibung Hinzufügen eines neuen Dauerauftrags. 			
	Der Dauerauftrag sollte folgende Informationen enthalten:			
	Begünstigter der Zahlung: Max Mustermann			

Abbildung 5.4. Die Seite "Suchergebnisse"

<u>Abbildung 5.4</u> zeigt die Seite *Suchergebnisse* an. Diese Seite zeigt alle Vorkommen pro Objekttyp in einer eigenen Ansicht an. Der Suchbegriff ist farblich unterlegt.

In der Statuszeile wird die für die Suche benötigte Zeit und die Anzahl der Treffer angezeigt, siehe <u>Abbildung 5.5</u>

🖇 14:05:52 Die Suche wurde nach 0.1 Sekunden erfolgreich beendet und lieferte 20 Treffer in 8 Ergebnissen.

+ ×

Abbildung 5.5. Die Status-Zeile

5.2.2.1.2. Schnell-Navigation

Die Volltextsuche ist nur in der Klaros-Testmanagement Enterprise Edition enthalten. In der Klaros-Testmanagement Community Edition ist statt der Volltextsuche ein Schnellnavigationsfeld für bekannte Element-IDs (wie z.B. TC00001) vorhanden. Dort werden die folgenden Objekttypen unterstützt:

- Projekt (P)
- Testumgebung (ENV)
- Testsystem (SUT)
- Testfall (TC)
- Testfallergebnis (TCR)
- Testsuiteergebnis (TSR)
- Testlauf (TR)

Issue-Management-System (IMS)

Bitte beachten Sie, dass IDs nur im aktuellen Projekt gesucht werden.



Abkürzungen bei der Schnellnavigation

Die Schnellnavigation ist unabhängig von Groß-/Kleinschreibung und kann auch durch das Weglassen von führenden Nullen abgekürzt werden. So verweisen TC00001, TC1 oder tc1 alle auf das gleiche Element.

5.2.2.2. Das Log-Panel



Abbildung 5.6. Das Icon "Log-Panel"

Das Log-Panel zeigt Statusmeldungen wie Warnungen oder Informationen an. Standardmäßig wird nur die letzte Statusmeldung angezeigt. Durch Klicken auf das Icon + auf der rechten Seite des Log-Panels werden die Statusmeldungen seit der letzten Anmeldung angezeigt. Durch Klicken auf das Icon \times wird das Log-Panel geschlossen.

- 🖇 14:22:34 Die Suche wurde nach 0.0 Sekunden erfolgreich beendet und lieferte 20 Treffer in 8 Ergebnissen. 🗕 🗙
- INFORMATION 14:22:34 Die Suche wurde nach 0.0 Sekunden erfolgreich beendet und lieferte 20 Treffer in 8 Ergebnissen.
- INFORMATION 14:22:34 Die Suche wurde nach 0.0 Sekunden erfolgreich beendet und lieferte 20 Treffer in 8 Ergebnissen
- Sinformation 14:22:26 Das zur Zeit ausgewählte Projekt ist: DE Finanz-Tracker (P00005)
- INFORMATION 14:22:23 Das zur Zeit ausgewählte Projekt ist: Printer Tester (P00003)

Abbildung 5.7. Das Log-Panel

5.2.2.3. Das Hilfemenü



Abbildung 5.8. Das Hilfe-Menü

Das Hilfemenü bietet direkte Verweise zu verschiedenen Ressourcen in Klaros-Testmanagement. Beim Klick auf das Icon 😮 erscheint ein Auswahlmenü mit den folgenden Einträgen:

Dokumentation	Öffnet die entsprechende Seite im Handbuch.
Tutorial	Öffnet das Klaros-Testmanagement Tutorial.
Forum Öffnet das Klaros-Testmanagement Benutzerforum.

Support kontaktieren Öffnet eine E-Mail, die an das Supportteam von Klaros-Testmanagement gesendet wird. Diese Option kann verwendet werden, wenn eine Frage oder ein Problem auftritt, das nicht mithilfe der Dokumentation oder des Forums gelöst werden konnte.

5.2.2.4. Das Benutzermenü



Abbildung 5.9. Das "Benutzermenü"

Das Benutzermenü rechts oben öffnet sich beim Klick auf das Icon.



Abbildung 5.10. Das geöffnete Benutzermenü

Hier werden der Name und die Rolle des aktuell eingeloggten Benutzers angezeigt. Es ist möglich die Sprache der Benutzeroberfläche zu wechseln und sich aus dem System auszuloggen.

5.2.3. Inhalt

Der Inhalt einer Seite ändert sich durch die Navigation über die Menüleiste oder andere Aktionen.

Die Seiten haben in der Regel den gleichen Aufbau. Es bestehen die folgenden zwei Grundmuster: Übersichtsseiten und Detailseiten.

5.2.3.1. Übersichtsseite

Auf einer Übersichtsseite werden mehrere Objekte des gleichen Typs (z.B. Testfälle) in einer Tabelle dargestellt und verwaltet.

5.2.3.1.1. Schnelle Tabellenfilter



Auf ausgewählten Tabellen können wie in <u>Abschnitt 5.2.2.1.1, "Volltextsuche"</u> beschrieben, die gelisteten Einträge per Volltextsuche gefiltert werden.

Durch Eingabe eines Suchbegriffs in das Eingabefeld oberhalb der Tabelle werden die Textfelder der angezeigten Objekte nach dem angegebenen Suchbegriff durchsucht und die Liste entsprechend gefiltert.

5.2.3.1.2. Filtern und Sortieren

Die in einer Tabelle angezeigten Objekte können nach ihren Feldern gefiltert und sortiert werden. Dies schließt auch Felder ein, die nicht direkt in einer Listenspalte sichtbar sind. Mit Klick auf das Icon *Filtern/Sortieren* ∇ oberhalb der Tabelle öffnet sich ein Panel mit zwei weiteren Tabellen. Die linke Tabelle enthält die aktiven Filter, die rechte Tabelle die aktiven Sortierkriterien. Wird ein Filter verwendet, ist das Icon mit einer orangefarbenen Markierung versehen.



Abbildung 5.11. Anzahl der aktiven Filter

= 📌 KLAR	OS TEST	MAN	AGEMENT				DE Finanz-Tra	cker 🖬 🕯	8		Q	× 0~ ±~
📝 Definieren	Testfälle											
🎎 Planen	Neu Filter / Sortierung									Speichr	em	Verwerfen
🏚 Ausführen	Fe	ld	Kriterium	Wert	Aktion	Kales Fintere under des	Sort	tierfeld			F	ichtung Aktion
🚯 Auswerten	Abhängigkeit Neu		✓ Enthält ✓			C C						
🖋 Konfigurieren									Anwenden	Rücksetz	en	Schließen
	+00%	€ 🗊 🗅		24 Einträge - Seite 1 von 3	K ┥ 🚺 2 3	▶ N 10 ▼		6	a √ Alle anzeigen		_	Q × & ≡
	ID 0 Q	Revision		Name 🔍	Ver	folgbarkeit ©	Priorität 🔍	Status 🕈	Ausführung 🔍 Sc	witte 🔍 Is:	sues 🖗	Aktion
	TC00024	1.0	Uberprüfen Sie die Verbit	ndung zum Server	Requirement 1.2.4 in test	tplan_16.txt	Hoch	Entwurf	Automatisiert	0	2/1	
	TC00022	1.0	Aktualisieren Sie das Das	shboard durch Hinzufügen eines	Requirement 1.2.4 in test	tplan_16.txt	Mittel	Entwurf	Manuell	6	5/0	ROAM
			Dauerauftrags.	-								
	1000021	1.0	Aktualisieren Sie das Das Dauerauftrags.	anboard durch stornierung eines	Requirement 1.2.4 in test	plan_16.txt	Mittel	Entwurf	Manuell	0	072	
	TC00020	1.0	Aktualisieren Sie das Das Lastschriftmandats.	shboard durch Stornierung eines	Requirement 1.2.4 in test	tplan_16.txt	Mittel	Entwurf	Manuell	6	0/0	C C & Ö
	TC00019	1.0	Testen der Verbindung zi	ar Datenbank.	/FinanceTracker/src/test nection.java	t/java/de/sabrina/testing/web/Con	Hoch	Entwurf	Automatisiert	0	0/0	000
	TC00018	1.0	Testen der Verbindung zi	im Server der Bank.	/FinanceTracker/src/tes/ nection.java	t/java/de/sabrina/testing/web/Con	Hoch	Entwurf	Automatisiert	0	0/0	000
	TC00017	1.0	Lastschriftmandat lösch	en.	Requirement 1.2.4 in test	tplan_16.txt	Mittel	Entwurf	Manuell	5	0/0	000
	TC00016	1.0	Daueraufträge löschen.		Requirement 1.2.4 in test	iplan_16.txt	Mittel	Entwurf	Manuell	5	0/0	C C & Ó
	TC00015	1.0	Fügen Sie einen Dauerau	ftrag hinzu.	Requirement 1.2.4 in test	tplan_16.txt	Mittel	Entwurf	Manuell	5	0/0	roði
	ID	Revision		Name	V	erfolgbarkeit	Priorităt	Status	Ausführung S	chritte li	ssues	Aktion
	Neu									Speich	em	Verwerfen

Abbildung 5.12. Das Panel "Filtern und Sortieren"

Unterhalb der beiden Tabellen befinden sich drei Buttons:

Anwenden wendet Filter und Sortierung auf die untenstehende Tabelle an.

Rücksetzen löscht alle Filter- und Sortiereinträge.

Schließen schließt das Filter- und Sortier-Panel.

Filterkriterien festlegen

Jede Zeile in der Filtertabelle stellt einen Filter dar, der auf die Tabelle angewendet wird.

Durch Klicken auf den Button Neu wird der Filtertabelle ein neuer Eintrag hinzugefügt. Jeder Eintrag besteht aus vier Spalten:

- Die Spalte *Feld* gibt das Feld an, nach dem gefiltert werden soll. Die Felder, nach denen gefiltert werden kann, variieren von Objekt zu Objekt.
- Die Spalte *Kriterium* legt den Typ der Einschränkung, die bei der Filterung vorgenommen werden soll, fest. Die Einschränkungstypen, die in einem Eintrag ausgewählt werden können, variieren von Feld zu Feld.
- Die Spalte *Wert* legt den Wert einer Einschränkung fest, z.B. den Benutzer nach dem ein Feld gefiltert werden soll.
- Das Icon Löschen (面) in der Spalte Aktion entfernt einzelne Filterkriterien.



Mehrfachauswahl

Bei manchen Angaben, wie z.B. Benutzern, ist auch eine Mehrfachauswahl möglich. Dazu die Taste STRG gedrückt halten und die gewünschten Einträge einzeln auswählen.

Filter (1) / Sortierung (0)									
Feld	Kriteriu	n	Wert		Aktion	Sortierfeld			Richtung Aktion
Erstellt von 🗸	Gleich	~	C Erika Mustermann Felix Mustermann Goldmirer Gress Markus Miyore Mass Mustermann Gover forann Patrick Relify	•		Kene Entridge vorhanden Neu			
Neu							Anwenden	Rücksetzen	Schließen

Abbildung 5.13. Nach Mehrfachauswahl filtern

Sortierkriterien festlegen

Tabelleneinträge können nach mehreren Kriterien sortiert werden. Die Sortierkriterien werden in der rechten Tabelle im Panel *Filtern/Sortieren* angegeben.

Es ist möglich, mehr als eine Sortieroption anzugeben. In diesem Fall hat die oberste Zeile der Tabelle die höchste und die unterste Zeile die niedrigste Priorität.

Beim Klick auf den Button Neu wird eine neue leere Zeile für ein Sortierkriterium am unteren Ende der Tabelle erzeugt. Zum Entfernen der Zeile dient das Icon Löschen m in der Spalte Aktion.

Die Tabelle der Sortieroptionen enthält drei Spalten:

- Die Spalte *Sortierfeld* bezeichnet das Feld der aufgelisteten Objekte nach dem sortiert werden soll.
- Die Spalte *Richtung* definiert die Richtung der Sortierreihenfolge, d.h. absteigend oder aufsteigend.
- Das Icon fil in der Spalte Aktion entfernt die jeweiligen Sortierkriterien.

5.2.3.1.3. Kategorisierung



Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Um eine große Anzahl von Objekten zu verwalten, können diese nach beliebigen Kriterien kategorisiert werden. Diese Funktion ist für die folgenden Objekte verfügbar: *Iterationen, Anforderungen, Testsysteme, Testumgebungen, Testfälle* und *Testsuiten*. Das Kategorisierungs-Panel wird mit einem Klick auf das Icon *Kategorien* 品 über der Tabelle geöffnet.

Im Kategorisierungs-Panel können *Kategoriegruppen* und *Kategorien* erstellt werden. Jede Kategoriegruppe enthält eine beliebige Anzahl von Kategorien, die in einer Baumstruktur angeordnet werden können. Objekte können in jeder Kategoriegruppe einer beliebigen Kategorie zugeordnet sein. Die Anzahl der Kategorien oder Kategoriegruppen ist nicht begrenzt.

Um die Kategorisierung anzuwenden, muss das Panel *Kategorien* geöffnet sein. Es enthält zwei verschiedene Ansichten: *Bearbeiten* und *Baum*. Sind noch keine Kategorien definiert, kann nur die Bearbeitungsansicht verwendet werden. Jede Ansicht ermöglicht die Auswahl der aktiven Kategoriegruppe über ein Dropdown-Menü oder die Auswahl der leeren Kategoriegruppe, um die Kategorisierung zu deaktivieren.

Kategorien bearbeiten

Ξ κια κα	DS TEST	MAN	AGEMENT		DE Finanz-Tracker				Q	× 0~ ±~
📝 Definieren	Testfälle									
📇 Planen	Neu Kategorisierung							Speic	hern	Verwerfen
🏟 Ausführen	Kategoriegruppe Name	Testfälle Testfälle	mit Aktionen 🗸 🕂 🗊 mit Aktionen							∷ ⊄
🕒 Auswerten	Aktione	n beiten		+ 2 m + 2 m						
🗲 Konfigurieren	II Lösc	hen ellen		+ 12° 10 + 12° 10			Speichern	Verwer	fen	Schließen
	+ ៤ ០ ኑ ៩) 1 1 1	17 Einträge - Seite 1 v	ron 2 🕅 ┥ 🚺 2 🕨 🗎 10 🗸		L.	Alle anzeigen			Q × & ≡
	□ ID≑ 🖓	Revision	Name 🗢	Verfolgbarkeit 🗢	Priorität 🗘 Sta	itus 🗘	Ausführung 🗘 Se	chritte 🗘 I	ssues 🖨	Aktion
	TC00024	1.0	Überprüfen Sie die Verbindung zum Server	Requirement 1.2.4 in testplan_16.txt	Hoch Ent	wurf	Automatisiert	0	2/1	COQÛ
	TC00019	1.0	Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/te sting/web/Connection.java	e Hoch Ent	wurf	Automatisiert	0	0/0	▨ёё
	TC00018	1.0	Testen der Verbindung zum Server der Bank.	/FinanceTracker/src/test/java/de/sabrina/te sting/web/Connection.java	e Hoch Ent	wurf	Automatisiert	0	0 / 0	┏┖ᇢᄈ
	TC00014	1.0	Aktualisierung des Dashboards nach einer Banküberweisung.	Requirement 1.2.4 in testplan_16.txt	Hoch Ent	wurf	Manuell	5	0/0	┏┏₽₫
	TC00013	1.0	Überweisen Sie Geld ohne Angabe einer Referenz.	Requirement 1.2.4 in testplan_16.txt	Niedrig Ent	wurf	Manuell	5	0 / 0	┏┖ᇢᄈ
	TC00012	1.0	Überweisen Sie Geld, ohne einen Betrag anzugeben.	Requirement 1.2.4 in testplan_16.txt	Hoch Ent	wurf	Manuell	4	0/0	ぴ₽₿₫

Abbildung 5.14. Die Ansicht "Kategorien bearbeiten"

In dieser Ansicht werden Kategoriegruppen und Kategorien erstellt und bearbeitet sowie die Kategoriehierarchien organisiert. Durch Klick auf das Icon + im oberen Teil dieser Ansicht wird eine neue Kategoriegruppe erstellt.

In dieser Ansicht können Kategorien hinzugefügt, bearbeitet und gelöscht werden. Eine Kategoriegruppe enthält immer eine Wurzelkategorie, die nicht gelöscht werden kann. Durch Klick auf das Icon + neben einer Kategorie wird dieser Kategorie eine neue Unterkategorie hinzugefügt. Durch Klick auf das Icon m wird die jeweilige Kategorie gelöscht.

Kategorien können im Feld *Name* umbenannt werden und mit Klick auf das Icon 😰 eine Beschreibung erhalten.

Das Ändern der Reihenfolge und Hierarchie der Kategorien ist auch per Drag-and-Drop möglich.

Die Baum-Ansicht

≡	Y KLAR	os test	ΜA	N A G	B E M E N T		DE Finanz-Tracker				Q ;	< 0 × ≗ ×
ľ	Definieren	Testfälle										
:9	2 Planen	Neu								Speich	hern	Verwerfen
		Kategorisierung	1									
\$	Ausführen	Testfälle mit A	ktionen		~							∷≡ 🖉
_		 Aktionen (17/24)									
¢	Auswerten	Bearbe Lösche	iten (3/ en (2/2)	3)								
ىر	K onfigurieren	Erstelle	en (2/2)									
												Schließen
		+ ៤ ០ ៦	8 û		17 Einträge - Seite 1 vor	12 K 4 1 2 F H 10 V		#7	Alle anzeigen			Q × & ≡
		□ ID ♦ 🤇	2 🗘	Revision	Name 🗢	Verfolgbarkeit 🗢	Priorität 🗢 Sta	tus 🗢 Aus	sführung 🗘 Se	chritte 🗢 🛛 Is	ssues 🖨	Aktion
		TC00024	▲	1.0	Überprüfen Sie die Verbindung zum Server	Requirement 1.2.4 in testplan_16.txt	Hoch Enty	vurf Aut	omatisiert	0	2/1	₢₽₽₫
		TC00019	0	1.0	Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/ testing/web/Connection.java	Hoch Enty	vurf Aut	omatisiert	0	0/0	ぴ∪₿₫
		TC00018	0	1.0	Testen der Verbindung zum Server der Bank.	/FinanceTracker/src/test/java/de/sabrina/ testing/web/Connection.java	Hoch Enty	vurf Aut	omatisiert	0	0/0	┎╻ᇢ╓
		TC00014	▲	1.0	Aktualisierung des Dashboards nach einer Banküberweisung.	Requirement 1.2.4 in testplan_16.txt	Hoch Enty	vurf Ma	nuell	5	0/0	┏┍₽₫
		TC00013	0	1.0	Überweisen Sie Geld ohne Angabe einer Referenz.	Requirement 1.2.4 in testplan_16.txt	Niedrig Entv	vurf Ma	nuell	5	0/0	┎╻ᇢ╓
		TC00012	0	1.0	Überweisen Sie Geld, ohne einen Betrag anzugeben.	Requirement 1.2.4 in testplan_16.txt	Hoch Enty	vurf Ma	nuell	4	0/0	┏┖₽₫
		TC00011	0	1.0	Überweisen Sie Geld ohne Angabe der	Requirement 1.2.4 in testplan_16.txt	Hoch Enty	vurf Ma	nuell	4	0/0	┏┍₽₫

Abbildung 5.15. Kategorien auswählen

Diese Ansicht zeigt einen Überblick über die Kategoriestruktur. Die vorhandenen Kategorien werden als zusammenklappbarer Baum dargestellt, wobei der Name jeder Kategorie zusammen mit der Anzahl der darin enthaltenen Objekte angezeigt wird. Die Beschreibung jeder Kategorie wird als Tooltip angezeigt, wenn sich der Mauszeiger über dem Namen der Kategorie befindet.

Objekte Kategorien zuordnen

Um Objekte einer Tabelle Kategorien zuzuordnen, muss das Panel *Kategorien* geöffnet sein und darin die Ansicht *Baum* aktiviert sein.

Um Objekte zuzuordnen, aktivieren Sie zunächst die Kategorie, die die zuzuordnenden Objekte enthält, und wählen Sie die Objekte aus. Klicken Sie dann auf das Icon 🗅 oberhalb der Tabelle. Es erscheint ein Dialog. Wählen Sie die entsprechende Kategorie, der Sie die Objekte zuordnen möchten, und klicken Sie auf den Button Zuordnen. Die ausgewählten Objekte sind nun der gewählten Kategorie zugewiesen.

Bitte beachten Sie, dass Objekte aus unterschiedlichen Kategorien nicht gleichzeitig einer einzigen Kategorie zugeordnet werden können. Für den Fall, dass die gewünschten Objekte verschiedenen Kategorien zugeordnet sind, muss der Schritt für jede Kategorie wiederholt werden.

5.2.3.1.4. Anzeige gelöschter Objekte

Wird ein Objekt gelöscht, ist es zunächst mit einer Löschmarkierung versehen und wird auf der Übersichtsseite nicht mehr angezeigt. Administratoren können sich gelöschte Objekte mit Klick auf den Button Zeige alle anzeigen lassen. Mit Klick auf den Button Nur aktive werden sie wieder ausgeblendet. Diese Funktion ist für die folgenden Objekte verfügbar: *Anforderungen, Aufgaben, Benutzer, Iterationen, Projekte, Testfälle, Testläufe, Testsegmente, Testsuiten, Testsysteme* und *Testumgebungen*.



Automatisches Speichern der Einstellung

Die Auswahl, ob gelöschte Objekte angezeigt werden sollen, wird pro Übersichtsseite und Administrator automatisch gespeichert.

5.2.3.1.5. Massenaktionen

Senterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

In vielen der Tabellen sind Massenaktionen auf die dargestellten Objekte anwendbar. Das heißt, mehrere Objekte können ausgewählt werden, um die gewünschte Aktion auf diese Auswahl durchzuführen. Dies gilt z.B. für Löschen, Duplizieren, Bearbeiten oder das Zuordnen zu Kategorien.

Um eine Massenaktion auszuführen, werden zunächst mithilfe der Auswahlfelder links neben den Einträgen der Tabelle die entsprechenden Objekte ausgewählt. Danach werden die bisher ausgegrauten Icons über der Tabelle aktiv. <u>Abbildung 5.16</u> zeigt die Icons für Massenaktionen auf der Testfall-Übersichtsseite.

Те	stfälle													
	Neu											Spe	ichern	Verwerfen
+	C D)	γĘ	Ì û ⊡	24 Einträge - Seite	1 von 1 🛛 📕	4		▶ 40) ~	1	Alle anzei	gen		Q × & ≡
			Revision	Name 🗢	Ve	rfolgbarkei	t 🗘		Priorität 🖨	Status 🖨	Ausführung 🖨	Schritte 🖨	Issues 🖨	Aktion

Abbildung 5.16. Die Auswahl der Massenaktionen

Bevor eine Massenaktion ausgeführt werden kann, müssen alle Änderungen in der Tabelle gespeichert sein. Sind noch ungespeicherte Änderungen vorhanden, erscheint ein Dialog.

Ein Bestätigungsdialog wird angezeigt, sobald auf ein Icon für Massenaktionen geklickt wurde. Diese Dialoge beschreiben die auszuführende Aktion und enthalten manchmal Eingabefelder, z.B. zur Eingabe des Revisionskommentars für Massen-Revisionsoperation. Alle Änderungen werden in die Datenbank geschrieben, sobald dieser Dialog bestätigt wird.

Definieren Testsysteme Neu	
±®t Planen Neu	
C2 A III 6 you 6 ausnewählt 6 Einträge - Seite 1 you 1 M	Speichern Verwerfen
Image: Ausführen Image: Departure for the second	555 U Alexandrager ⊂
Auswerten SUT00005 Finance Tracker 2.0.1 SUT00004 Finance Tracker 2.0.0 SUT00003 Finance Tracker 1.11	1/0 ピ合節 0/0 ピ合節 1/0 ピ合節
	0/0 戊子前 0/0 戊子前 0/0 戊子前
ID Version Neu Mehrere Objekte bearbeiten Buildnumber range	elter Abbrechen

Abbildung 5.17. Massenaktionen auf Objekte

5.2.3.2. Detailseite

Auf einer Detailseite wird ein einzelnes Objekt dargestellt und verwaltet.

5.2.3.2.1. Druckansichten

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Objekte können durch Klicken auf das Icon 🖨 in einer druckfreundlichen Darstellung auf einer eigenen Seite angezeigt werden. Das Icon ist in der Spalte *Aktion* in der Tabelle, in der Liste der Massenaktionen links oberhalb von Tabellen und als großes Icon oben rechts auf den *Detailseiten* zu finden.

Die Anzeige kann durch die folgenden Parameter verändert werden.

- Mit der Option Leere Einträge ausblenden werden Felder ohne Inhalt in der Ansicht ausgeblendet.
- Wird die Option *Diagramme anzeigen* ausgewählt, werden in der Druckansicht Diagramme mit eingebettet.
- Mit *Testumgebungen* und *Testsysteme* können die Testumgebungen und Testsysteme ausgewählt werden, deren Ergebnisse auf der Seite dargestellt werden sollen.
- Die Einstellung *Details* legt fest, ob die Daten vollständig oder verkürzt angezeigt werden sollen.

Die ausgewählten Druckoptionen bleiben aktiv, solange der Benutzer eingeloggt ist.

Abbildung 5.18 zeigt die Druckansicht für eine Testsuite an.



Druckeinstellungen im Browser

Die Option *Hintergrund drucken (Farbe und Bilder)* muss aktiviert sein, um die beste Ausgabequalität zu erreichen.



Abbildung 5.18. Die Druckansicht "Testsuite"

5.2.3.2.2. Lesezeichen

Die Detailseite eines beliebigen Objekts kann direkt über eine spezielle, vom System generierte URL aufgerufen werden. Durch Klicken auf das Icon \square wird diese URL der aktuellen Seite in die Zwischenablage kopiert und kann mit jeder anderen Anwendung geteilt werden.

≡ 🎐 K L A R	OS TEST MANAGEMENT	DE Finanz-Tracker 🖬 📑	Q @v ≜v
📝 Definieren	STestfallergebnis: TCR0006128 (TC00021 - Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.)	*	₽₽₽₽
2 Planen	Eigenschaften Anhänge Issues (2) Änderungen	Speichern Verwerfen	Zurück
🔅 Ausführen	Zusammenfassung		
🔶 Auswerten	Beschreibung		
Y Kontigurieren	Ergebeite Start 15.03.21, 08:45.12 Augedführ von Sein Deutsch Ausführungsber Otox20.21, 08:45.12 Tertialer TRUDOCHARG: 15.03.21, 08:45.12 Tertialer TRUDOCHARG: 15.03.21, 08:45.12 Tertialer TRUDOCHARG: 15.03.21, 08:45.12 Tertialer Struttgebeiten ErvOtox20: Audeold Struttghorie Tertistyter Struttgebeiten ErvOtox20: Audeold Struttghorie Tertistyter Trade 21.0		
	Schritte		
	topenit Constitution Constit Constit Constitution Constitution Constitution	zusämmentassung Beschreibung	00:00:09
	2 Vestunden Alter Erstellenden Machen Erwettete Eightets Voldedingung Nachbedingung Berutzendefiniert Wahnen Sie der Taste Einstelletlittig, die sich im Hauptemeine befindet.		00:00:01 🔗
	3 🖉 🗸 Bestanden 🛛 Allice Diractifek Dipolnis Vobelingung Nachbedingung Besutzestefiniet Wahlen Sie die Schattläche Loschen 🚯		00:00:01 🥔

Abbildung 5.19. Lesezeichen erstellen

<u>Abschnitt 6.1.2.5.1, "Projektspezifische Rollen zuweisen"</u> erläutert die Konfiguration der Zugriffskontrolle zu den mit Lesezeichen versehenen Seiten.

5.2.3.2.3. Übersicht



Für jedes Objekt in Klaros-Testmanagement gibt es eine eigene Übersichtsseite, die die wichtigsten Daten zu diesem Objekt auf einen Blick zeigt. Zusätzlich zu den reinen Daten enthalten viele Übersichtsseiten verschiedene Tabellen und Diagramme, wie zum Beispiel die Erfolgshistorie für Testsuiten, die hilft, den Testfortschritt zu visualisieren.



Abbildung 5.20. Die Ansicht "Übersicht"

5.2.3.2.4. Benutzerdefinierte Eigenschaften



Um Projekte an spezifische Anforderungen anzupassen, können Objekte um benutzerdefinierte Felder erweitert werden.

Diese werden für jedes Projekt getrennt im Bereich *Projekt* in der Ansicht *Benutzerdefiniert* angelegt, siehe <u>Abbildung 5.21</u>.

= 📌 K L A R	d s test management	DE Finanz-Tracker 🛅	≡ Q x 0 × ≛×
📝 Definieren	P00005 - DE Finanz-Tracker		🖨 C 오
😤 Planen	Eigenschaften Benutzerdefiniert (6) Objekte kopieren Zugang In	egration Ergebnisse Änderungen	Speichern Verwerfen Zurück
🅸 Ausführen	Anforderung Aufgabe Iteration Testfall (1) Testfallergebnis	Testlauf Testschritt Testsegment Tests	uite Testsystem (1) Testumgebung (4)
🕒 Auswerten	Name	Тур	Werte Voreinstellung Aktion
🔎 Konfigurieren	Zusätzliche Informationen bzw. Bemerkungen	Textfeld	- 1 m
	II Laoorexperiment	Aufzählung (5)	· ↓ Ш ② Temperatur < 0°C ↓ 面
	Erstellt vor 3 Jahren von Felix Mustermann		Zuletzt geändert vor 2 Jahren von Selen Deutsch
			Speichern Verwerfen Zurück

Abbildung 5.21. Die Ansicht "Benutzerdefinierte Eigenschaften"

Mit Klick auf den Button Neu wird eine neue benutzerdefinierte Eigenschaft erzeugt. Nach dem Hinzufügen einer neuen Eigenschaft muss der Eigenschaftstyp ausgewählt werden.

Für die folgenden Objekte können benutzerdefinierte Felder angelegt werden, diese werden jeweils in einer eigenen Ansicht verwaltet: **Testfall**, **Testschritt**, **Testsegment**, **Testsuite**, **Testumgebung**, **Testsystem**, **Testlauf**, **Anforderung**, **Iteration**.

Die vier möglichen Eigenschaftstypen sind:

Text	Die Eigenschaft wird als einzeiliges Textfeld angelegt.
Textfeld	Die Eigenschaft wird als mehrzeiliges Textfeld angelegt.
Boolescher Wert	Die Eigenschaft wird als Check-Box angelegt.
Aufzählung	Die Eigenschaft ist ein Dropdown-Feld zur Auswahl eines ein- zelnen Eintrags aus einer vordefinierten Liste von Werten.

Der Vorgabewert der Eigenschaft kann hier ebenfalls vergeben werden.

Nachdem die neue Eigenschaft gespeichert wurde, kann der Eigenschaftstyp nicht mehr geändert werden. Der Name der Eigenschaft und – falls vorhanden – die Liste der Aufzählungswerte können jederzeit geändert werden.

Aufzählungswerte

Das Icon 😰 neben dem Namen der Aufzählung öffnet den Dialog zum Bearbeiten der Aufzählungswerte, siehe <u>Abbildung 5.22</u>.

≡ 5	🞐 KLAR	o s test management	DE Finanz-Tracker 🖬 🗮 🔍 Q	0° 1°
📝 D	efinieren	P00005 - DE Finanz-Tracker		a 🖓 🖓
🚢 Pi	Planen	Eigenschaften Benutzerdefiniert (6) Objekte kopieren Zugang Integration Ergebnisse	Speichern Verwerfen Änderungen	Zurück
🏚 A	usführen	Anforderung Aufgabe Iteration Testfall (1) Testfallergebnis Testlauf Testschritt	Testsegment Testsuite Testsystem (1) Testumgeb	ung (4)
🕒 A	uswerten	Name Geprüft von Zusätzliche Informationen bzw. Bemerkungen	Typ Werte Voreinstellung Text - Textfeld -	Aktion 章 団 章 団
, р к	Configurieren	Laborexperiment Temperatur Elemente der Aufzählung bearbeiten	Boolescher Wert - V Aufzählung (5) C Temperatur < 0°C	‡ ∰ ‡ ∰
		Name Erst # 0°C < Temperatur < 20°C # Raumtemperatur # 24°C < Temperatur < 40°C # 40°C < Temperatur # 40°C < Temperatur Neu Neu	Aktion 1 1 1 Indert vor 21 Tager 1 1 1 Verwerfen 1 1 1 1 1 1 1 1 1 1 0 K Abbrechen	n von selen2 selen2 Zurück

Abbildung 5.22. Bearbeiten einer Aufzählung

Sobald benutzerdefinierte Eigenschaften angelegt sind, können sie in der Ansicht *Benutzerdefiniert* des zugehörigen Objekts bearbeitet werden (<u>Abbildung 5.23</u>).

≡	👱 KLAR	o s test management	DE Finanz-Tracker	■ ≡ Q × Q × L×
Ľ	Definieren	ENV00005 - Android 8 Smartphone		음 🗋 😌 😌
***	Planen	Übersicht Eigenschaften <u>Benutzerdef</u> i	ert Anhänge Iterationen (5) Aufgaben Ergebnisse (9920) Ände	Speichern Verwerfen Zurück
\$	Ausführen	Temperatu	Temperatur < 0°C	~
¢	Auswerten	Geprüft vor Laborexperimen		
ير	Konfigurieren	Zusätzliche informationen bzw. Bernerkunger		
		Erstellt vor 3 Jahren von Sabrina Gidley		Zuletzt geändert vor 2 Jahren von Felix Mustermann
				Speichern Verwerfen Zurück

Abbildung 5.23. Die Ansicht "Benutzerdefinierte Eigenschaften"

Für Testläufe werden die Werte der benutzerdefinierten Eigenschaften im jeweiligen Ausführungsdialog erfasst, siehe <u>Abbildung 8.3</u>.

5.2.3.2.5. Revisionen

Revisionierbare Objekte sind Anforderungen, Testfälle, Testsegmente und Testsuiten.

Eine neue Revision eines Objekts sollte erstellt werden, wenn eine größere Änderung vorgenommen wird. Beispielsweise kann eine ältere Revision eines Testfalls nur in Verbindung mit älteren Versionen eines Testsystems ausgeführt werden, da neu hinzugefügte Testschritte nur auf neuere Versionen des zu testenden Systems anwendbar sind.

Revisionen von Objekten können über die Ansicht *Revisionen* verwaltet werden. Diese Ansicht zeigt die Revisionshistorie für das Objekt an und erlaubt den mit jeder Revision verbundenen Revisionskommentar zu ändern oder neue Revisionen zu erstellen.



Abbildung 5.24. Die Ansicht "Revisionen"

Beziehungen zwischen Objekt-Revisionen

Viele Objekte stehen in einer direkten Beziehung zu anderen Objekten. Objekte wie z.B. Testfälle oder Anforderungen können in verschiedenen Revisionen existieren, sodass eine Revision eines Objekts von jeder Revision anderer Objekte referenziert werden kann. Jede Revision eines Objekts kann seine eigenen Referenzen auf andere Objekte enthalten, unabhängig von den Beziehungen der anderen Revisionen. Darüber hinaus kann jede Revision eines Objekts mit einer oder mehreren spezifischen Revisionen eines anderen Objekts in Beziehung stehen.

Beispiel: Gegeben ist der Testfall TC0001 und die Anforderung R0001. Die Anforderung R0001 wird durch den Testfall TC0001 abgedeckt. Beide befinden sich noch in der ursprünglichen Revision 1.0.



Nun wird eine neue Revision 1.1 der Anforderung R0001 erstellt, die auch vom Testfall TC0001 abgedeckt wird. Damit deckt TC0001 die Anforderung R0001 in beiden Revisionen ab.



Als nächstes wird eine neue Revision des Testfalls TC0001 erstellt, da die Revision 1.0 der Anforderung R0001 veraltet ist. Daher wird in Revision 1.1 des Testfalls nun auf die neueste Revision der Anforderung verwiesen.



Nachdem wir die neue Revision des Testfalls TC0001 erstellt haben, stellen wir fest, dass die Anforderung R0002 ebenfalls von dieser Testfallrevision abgedeckt wird. Daher fügen wir eine weitere Abdeckungsbeziehung zwischen R0002 und den Testfall TC0001 in Revision 1.1 hinzu.



Dieses Beispiel zeigt, dass, obwohl der Testfall und die Anforderung die gleichen Objekte sind, sich ihre Beziehung im Laufe ihrer Revisionshistorie ändern kann. Referenzen werden pro Revision definiert, und kann daher bei jeder Revision unterschiedlich sein. Dies gilt nicht nur für Testfälle und Anforderungen, sondern auch für andere revisionsfähige Objekte wie Testsuiten oder Testsegmente.

5.2.3.2.6. Anhänge

Zu vielen Objekten können Dateianhänge hochgeladen und mit diesem Objekt verknüpft werden. Dies kann innerhalb der Ansicht *Anhänge* auf der entsprechenden Detailseite des Objekts oder während der Ausführung eines Testfalls erfolgen.

In der Tabelle auf der Detailseite für Anhänge werden der *Name*, die *Größe*, der *Dateityp*, die *Version* und der Name des Erstellers angezeigt. Die Zeileninhalte können direkt in der Tabelle bearbeitet werden.

Mit Klick auf den Button Anhang hochladen erscheint ein Dialog. Der Button Q Auswählen öffnet ein Dialogfeld zum Auswählen der Datei. Mit OK wird die ausgewählte Datei in die Anwendung hochgeladen. Möchten Sie weitere Dateien hochladen, wiederholen Sie die Schritte. Klicken Sie Speichern , um alle Änderungen zu speichern oder Verwerfen , um den Dialog zu schließen.



Clipboard-Unterstützung

Auf Browsern der Chrome-Familie (Chrome, Brave Microsoft Edge) wird außerdem ein Upload via **Ctrl+V** unterstützt. So können z.B. Screenshots ohne Speicherung im Filesystem übertragen werden.



Abbildung 5.25. Die Seite "Anhänge hochladen"

In der Aktionsspalte können gespeicherte Anhänge heruntergeladen (\underline{x}) und gelöscht (\overline{m}) werden.



Beschränkung der Dateigröße

Die Dateigröße eines Anhangs kann begrenzt werden, um das Hochladen von übermäßig großen Dateien zu verhindern. Dies kann unter *Konfigurieren -> System -> Sonstiges* eingestellt werden.

Wird eine Datei mit demselben Namen mehrfach hochgeladen, wird jedes Mal eine neue Version des Anhangs angelegt. Diese enthält die neu hochgeladene Datei.

5.2.3.2.7. Testläufe und Ergebnisse



Einige Detailseiten zeigen eine Ansicht der Testläufe und Testergebnisse, die sich auf das angezeigte Objekt beziehen, z. B. ausgeführt in der angezeigten Testumgebung oder ausgeführt durch den angezeigten Benutzer. Ein Beispiel für die Ansicht *Testergebnisse* ist in <u>Abbildung 5.26</u> dargestellt.

= 👱 K L A R	d s test management	DE Finanz-Tracker	≣ ≣ Q × Q × ≗×
📝 Definieren	SUT00005 - Finance Tracker 2.0.1		🖶 🛛 🔇 😜
😂 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge	Issues (1) Iterationen Aufgaben Ergebnisse (61)	Speichern Verwerfen Zurück Änderungen
🅸 Ausführen	Testfallergebnisse (61) Testsuiteergebnisse (6) Testlä	ufe (39)	
욙 Auswerten	61 Einträge-Seite 1 von 7 K ID ♦ 🛛 Start ♦ Iteration ♦	1 2 3 4 5 ▶ 10 ✓ Testlauf \$ Testfall \$ Testumgebung \$	↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓ </th
∲ Konfigurieren	TCR0001530 vor 4 Jahren TCR0000352 vor 4 Jahren TCR0000334 vor 4 Jahren TCR0000329 vor 4 Jahren TCR0000329 vor 4 Jahren TCR0000328 vor 4 Jahren TCR0000324 vor 4 Jahren TCR0000324 vor 4 Jahren TCR0000324 vor 4 Jahren TCR0000324 vor 4 Jahren TCR0000325 vor 4 Jahren TCR0000321 vor 4 Jahren TCR0000321 vor 4 Jahren TCR0000321 vor 4 Jahren TCR0000325 vor 4 Jahren TCR0000321 vor 4 Jahren TCR0000321 vor 4 Jahren TCR0000315 vor 4 Jahren TCR00015 vor 4 Jahren	TRU0001115 TC00021 Android 8 Smartphone TRU0001113 TC00021 Android 8 Smartphone TRU0000207 TC00017 Android 10 Smartphone TRU000201 TC00017 Android 10 Smartphone TRU000201 TC00011 Android 10 Smartphone TRU000201 TC00010 Android 10 Smartphone TRU0000201 TC00013 Android 10 Smartphone TRU0000195 TC00012 Android 10 Smartphone TRU0000193 TC00005 Android 10 Smartphone TRU0000187 TC00005 Android 10 Smartphone	Felix Mustermann 00:00:06 ▲ ▲ ⊕ Q Felix Mustermann 00:00:11 ∅ ▲ ⊕ Q Tim Thiel 00:00:25 ∅ ⊕ Q Till Tegen 00:00:20 ∅ ⊕ ⊕ Q Markus Meyer 00:00:20 ∅ ⊕ ⊕ Q Tim Thiel 00:00:22 ∅ ⊕ ⊕ Q Thomas Tafel 00:00:29 ∅ ⊕ ⊖ Q Timo Tunklik 00:01:24 ∅ ⊕ ⊖ Q Timo Tunklik 00:01:24 ∅ ⊕ ⊖ Q
	Erstellt vor 4 Jahren von Talal Arif		Zuletzt geändert vor 4 Jahren von Felix Mustermann Speicherm Verwerfen Zürück



In diesen beiden Tabs sind die Namen der Objekte Links zu den *Detailseiten* der Objekte. Das Icon Q zeigt die *Detailseite* des entsprechenden Testlaufs oder Testergebnisses an.

In der Ansicht *Testläufe* (<u>Abbildung 5.26</u>) werden ebenfalls Berichte erzeugt, wie in <u>Abschnitt 9.3.4, "Der Testlauf - Bericht"</u> beschrieben.

5.2.3.2.8. Änderungshistorie

enterprise edition Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

In dieser Ansicht wird die gesamte Änderungshistorie des Objektes seit der Erstellung angezeigt.

Abbildung 5.27 zeigt die Ansicht Änderungen für einen Testfall mit seiner Historie.



Abbildung 5.27. Die Ansicht "Änderungen"

Alle Änderungen werden farbig gekennzeichnet: grün für neu hinzugekommene Einträge, rot für gelöschte Einträge. Änderungen für nicht-textuelle Werte werden in der Form *Alter Wert -> Neuer Wert* markiert.

Außerdem ist angegeben, welcher Benutzer die Änderung zu welchem Zeitpunkt vorgenommen hat. Wurden mehrere Änderungen auf einmal vorgenommen, werden sie pro Speichervorgang zusammengefasst.

5.3. Hauptfunktionen

In diesem Abschnitt werden allgemeine Querschnittsfunktionen von Klaros-Testmanagement im Detail beschrieben.

5.3.1. Konflikte beim Speichern lösen

Wird ein Objekt von zwei Benutzern gleichzeitig bearbeitet, so können diese Änderungen kollidieren. Hierfür sind in der Anwendung die folgenden Maßnahmen vorgesehen.

Beim Speichern eines Objekts wird versucht, mögliche Konflikte mit gleichzeitigen Änderungen weitestgehend automatisch aufzulösen. Alle von anderen Benutzern vorgenommenen Änderungen, die sich nicht auf Felder auswirken, die der aktuelle Benutzer geändert hat, werden ohne zusätzliche Benutzereingaben zusammengeführt.

Wurden Feldinhalte geändert, die in der Zwischenzeit auch von einem anderen Benutzer geändert wurden, wird ein Dialog angezeigt. Dort kann der Benutzer wählen wie der Konflikt aufgelöst werden sollen.

Abbildung 5.28 zeigt den Dialog Konflikt Auflösung für einen Konflikt bei einer Testfalländerung.

≡ 👱	DE Finanz-Tracker 🔳 📕 🔍	0~ 1 ~
📝 Definieren	Testsuiten	Manualan
😤 Planen	Neu Speichern + 2 □ 2 □ 2 □ 7 Einträge-Seite 1 von 1 M 1 ▶ № 10 ··· 2 = 0 7 Alle anzeigen Q	× & ≡
🏟 Ausführen	ID+ Q Revision Name \$ Testsystem \$ Testfälle \$ TS00008 1.0 Überprüfen Sie die Verbindung zum Server Finance Tracker 2.1.0 1 Z	Aktion
🔥 Auswerten	T S00007 1.0 Daueraufträge bearbeiten 3 2 T S00006 1.0 Übersicht 7 2	1080 1080
🖌 Konfigurieren	T 500004 1.0. Vickfierungen übergriffen Einanse Tracker 1.0.0 2 2 T 500003 T 500003 4 2 T 500002 Ihre Änderungen kollidieren mit gleichzeitigen Änderungen anderer Benutzer. Bitte 8 2	?040 ?040 ?040
	ID Name 4 2	名 Aktion
	Neu Übersichtseite (Benutzer) Speichern Übersicht (Datenbank) Speichern Speichern	Verwerfen
	Speichern Verwerfen	

Abbildung 5.28. Der Dialog "Konflikt-Auflösung"

Bei jedem Konflikt hat der Benutzer die Wahl, das Objekt mit dem eingegebenen Wert zu speichern oder den aktuell in der Datenbank vorhandenen Wert beizubehalten. Bei Listen, wie z.B. der Liste der Testschritte in einem Testfall, hat der Benutzer die Möglichkeit, mit *Zusammenführen* die Änderungen zu übernehmen die aktuell in der Datenbank befindliche Version zu behalten. Damit soll der unbeabsichtigte Datenverlust durch das Löschen von Objekten, die von anderen Benutzern hinzugefügt wurden, verhindert werden.

Die Option *Zusammenführen* erstellt eine Liste, die alle Elemente beider Listen (Benutzereingaben und Datenbank) in einer Annäherung an die richtige Reihenfolge enthält.

Mit Klick auf den Button Speichern werden die ausgewählten Änderungen in der Datenbank gespeichert, Verwerfen verwirft die Änderungen des Benutzers.

- 5.3.2. Objekte löschen, dauerhaft entfernen und wiederherstellen
- 5.3.2.1. Objekte löschen

Objekte können durch Klick auf das Icon i deaktiviert bzw. gelöscht werden, sodass es für Berichte und Auswertungen nicht mehr herangezogen wird. Ist das zu löschende Objekt mit anderen Objekten verknüpft (z.B. eine Anforderung, die mit einem Testfall verknüpft ist) oder in anderen Objekten enthalten ist (z.B. ein Testfall, der in einer Testsuite enthalten ist), wird ein Dialogfenster angezeigt (siehe <u>Abbildung 5.29</u>).

≡ 😤 К L A R	OS TEST MANAGEMENT DE Frienz-Tracker 🖬 🗮 📃	۹ 0۰ ≛۰
Dofinioron	Anforderungen	
Dennieren	Sind Sie sicher?	
-	Möchten Sie diese Anforderungen wirklich löschen?	Speichern Verwerfen
📇 Planen		Q x & =
	ID ¢ Beschreibung ¢	
🏟 Ausführen	R00005 Revision: 1.2 - ID: R00005, Name: Die Anwendung ist mit Android-Smartphones, Tablets und Smartwatches kompatibel	
	R00005 Revision: 1.1 - ID: R00005, Name: The application runs smoothly and without failures on Android smartphones and tablets.	
	R00005 Revision: 1.0 - ID: R00005, Name: The application runs smoothly and without failures on Android smartphones.	
Addition	Die obenstehenden Objekte werden aus den folgenden Iterationen entfernt	
	ID 🗘 Name 🗘	
🏓 Konfigurieren	ITR00003 1.1.X-Sprint 01 - Übersichten	
	ITR00007 2.0.x-Sprint 01-Dashboard hinzufügen	s Testfälle Aktion
	ITR00008 2.0.x-Sprint 02-Dashboard erweitern	Quither Manufac
	ITR00006 1.1.x-Sprint 04-Optimierung für Tablets	Speichern verwerten
	ITR00005 1.1.x-Sprint 03-Annullierung von Einzugsermächtigungen	
	ITR00002 1.0.x-Sprint 02-Erweiterung der Grundfunktionen	
	ITR00004 1.1.x-Sprint 02-Daueraufträge bearbeiten	
	ITR00001 1.0.x-Sprint 01-Grundfunktionen	
	ITR00009 2.1.x-Sprint 01-Optimierung für die Verwendung auf Smartwatches	
	Die obenstehenden Objekte werden aus den folgenden Testfällen entfernt	
	ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID ID	
	TC00021 Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	
	TC00016 Daueraufträge löschen.	
	TC00003 Uberweisen Sie Geld auf ein anderes Bankkonto.	
	TC00015 Fugen Sie einen Dauerauttrag hinzu.	
	TCUU1/ Lastschrittmandat loschen.	
	TCOUDD Komostano protein.	

Abbildung 5.29. Der Dialog "Objekte löschen"

Im Dialogfenster werden alle Abhängigkeiten, Beziehungen und Vorkommen des zu löschenden Objekts angezeigt. Mit OK wird das Objekt gelöscht, mit Abbrechen wird das Fenster geschlossen und das Objekt nicht gelöscht.

5.3.2.2. Objekte dauerhaft entfernen

Objekte können durch Klick auf das Icon \bigotimes unwiederbringlich entfernt werden. Hierfür ist es notwendig, dass diese Objekte bereits gelöscht bzw. deaktiviert wurden. Standardmäßig werden gelöschte bzw. deaktivierte Objekte nicht angzeigt. Durch ein Klick auf Alle anzeigen können diese eingeblendet werden. Nachdem Klicken auf das Icon \bigotimes erscheint ein Dialogfenster (siehe Abbildung 5.30). Mit OK wird das Objekt dauerhaft entfernt, mit Abbrechen wird das Fenster geschlossen und das Objekt nicht endgültig gelöscht.

≡ ५		DS TES	Т	MANA	GEMENT			DE Finanz-Tracker 🔤			۹	× @~ ≗ ~
📝 Det	efinieren	Anforderu	ngen									
• • • Di-		Neu									Speichern	Verwerfen
	anen	+ @ 0	e e)	Ť	7 Einträge - Seite 1 von 1	H 4 1 Þ H [10 🗸	品了 Nu	aktive		Q × & ≡
άτ Διι-	ısführen	ID 🗘	0	Revisior			Name 🗢		Priorität :	Status 🕯	Testfälle	Aktion
	ordinen -	R02048	•	1.0	Keine Verbindungsfehler					Entwurf	0	
		R00006	S	1 .0	Das Dashboard wird korrekt	aktualisiert, nachdem e	etwas geändert wurde.		Mittel	Entwurf	4	嫦健暮前
C, Au	Iswerten	R00005	S	1.2	Die Anwendung ist mit Andre	oid-Smartphones, Table	ets und Smartwatches kompa	itibel	Hoch	Entwurf	5	$C \square \ominus \square$
		R00004		1.0	Es ist erforderlich, dass sich	die Datenbank immer i	in einem konsistenten Zustan	id befindet.	Hoch	Entwurf	0	嫦健骨前
🖉 🔑 Koi	onfigurieren	R00003	S	1.3	Mindestens 100 Benutzer so	ollten die Anwendung gl	leichzeitig ohne Leistungsverl	luste nutzen können.	Hoch	Entwurf	8	嫦健ᇢ໖
		R00002	S	1.0	Kein Verbindungsfehler.				Niedrig	Entwurf	2	₡₽₽₫
		R00001	S	1 .0	Die Verarbeitung der Ergebni	isse der Benutzereingal	ben dauert weniger als 4 Seku	unden.	Mittel	Entwurf	2	┏┍₽ΰ
		ID		Revisior			Name		Priorität	Status		Aktion
			3ind S	ie sicher, d	ass die Anforderung unwiederl	bringlich entfernt werde	en soll? Dieser Vorgang kann i	nicht rückgängig gemacht werd	en.		Speichern	Verwerfen
		ID≑										
		R02048	1.0 -	- R02048, k	eine Verbindungsfehler							
								ОК	Abbre	chen		

Abbildung 5.30. Der Dialog "Objekte dauerhaft entfernen"

5.3.2.3. Objekte wiederherstellen

Gelöschte bzw. deaktivierte Objekte können durch Klicken auf das Icon 🗗 wiederhergestellt werden. Nachdem Klicken auf das Icon 🗊 erscheint ein Dialogfenster (siehe <u>Abbildung 5.31</u>). Mit OK wird das Objekt wiederhergestellt, mit <u>Abbrechen</u> wird das Fenster geschlossen und das Objekt bleibt deaktiviert. Wiederhergestellte Objekte werden bei Berichten und Auswertungen miteinbezogen.



Abbildung 5.31. Der Dialog "Objekte wiederherstellen"

5.3.3. Objekt-Eigenschaften referenzieren

Allgemeine und benutzerdefinierte Eigenschaften (siehe <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte</u> <u>Eigenschaften"</u>) können während der Ausführung von Aufgaben, Testfällen oder Testsuiten in den angezeigten Textfeldern referenziert und aufgelöst werden. Beispielsweise referenziert die folgende Testfallvorbedingung das Testsystem, in dem sie ausgeführt wird:

"Bitte vergewissern Sie sich, dass die Betriebssystemversion, die unter /config/os eingetragen ist, mit %sut:buildnumber% übereinstimmt, bevor Sie die Testfall-Ausführung starten."

Beim Ausführen des Testfalls wird *%sut:buildnumber%* durch den Wert der benutzerdefinierten Eigenschaft *buildnumber* des Testsystems mit dem der Testfall ausgeführt wird ersetzt. Lautet also angenommen der Wert für die Eigenschaft 1.44_alpha3, wird der Text wie folgt ausgegeben:

"Bitte vergewissern Sie sich, dass die Betriebssystemversion, die unter /config/os eingetragen ist, mit 1.44_alpha3 übereinstimmt, bevor Sie die Testfall-Ausführung starten."



Abbildung 5.32. Aufgelöste Objekteigenschaft

Das Ersetzen der Eigenschaft wird sowohl für die gemeinsamen vordefinierten Attribute eines Objekts (z.B. den Namen oder die ID) als auch für die benutzerdefinierten Eigenschaften unterstützt. Der Name der gemeinsamen Attribute kann von den entsprechenden Bean-Methoden abgeleitet werden, die in der <u>de.verit.klaros.core.model</u> beschrieben sind.

Für die folgenden Objekte werden Eigenschaften derzeit unterstützt:

Iteration	itr
Aufgabe	job

Testsystem	sut
Testfall	tc
Testumgebung	env
Testlauf	tr
Testsegment	seg
Testsuite	ts

Angenommen, der folgende Beispieltext wird in der Testfallvorbedingung verwendet:

"Bitte stellen Sie sicher, dass Ihr Testsystem die Version %sut:productversion% verwendet."

Den Attributnamen productversion finden Sie in der Klassendokumentation für die Klasse SUT-Implementation in <u>de.verit.klaros.core.model.KlarosSUTImplementation</u>. Wenn die dort definierte Version einen Wert von 1.0.1 hat, wird bei der Anzeige der Vorbedingung die folgende Ausgabe ausgegeben:

"Bitte stellen Sie sicher, dass Ihr Testsystem die Version 1.0.1 verwendet."



Eigenschaftsreferenzen beachten keine Groß-/ Kleinschreibung!

Bei der linken Seite eines Verweises wird die Groß-/Kleinschreibung nicht berücksichtigt. %sut:productversion%, %SUT:productversion% und %Sut:productversion% liefern dasselbe Ergebnis.

5.3.4. Referenzieren von Anhängen

Ähnlich wie bei der Referenzierung von benutzerdefinierten Eigenschaften in Testfall- und Testfallschritten (<u>Abschnitt 5.3.3, "Objekt-Eigenschaften referenzieren"</u>) können auch Werte aus binären Anhängen referenziert werden. Derzeit werden CSV- und XLS-Dateien unterstützt.

Binäre Anhänge der folgenden Objekte werden derzeit unterstützt:

Iteration	itr-att
Aufgabe	job-att
Testsystem	sut-att
Testfall	tc-att
Testumgebung	env-att
Testlauf	tr-att
Testsegment	seg-att
Testsuite	ts-att

Zusätzlich zu den obigen Ausführungen kann der globale Platzhalter att verwendet werden. Die Position des Anhangs ist damit nicht näher spezifiziert. Es werden in diesem Fall alle an der Testausführung beteiligten Objekte nach dem Anhang durchsucht.

Ein Platzhalter kann in der Form %[Objekttyp]:[Dateiname]:[Spalte]:[Zeile]% unter Verwendung der folgenden Parameter definiert werden:

	Platzhalter-Parameter
[object type]	Der Typ des Objekts, das den binären Anhang enthält.
[file name]	Der Name des binären Anhangs.
[column]	Die Spaltennummer des Wertes des binären Anhangs.
[row]	Die Zeilennummer des Wertes des binären Anhangs.



Excel-Dateien referenzieren

Beim Referenzieren von Excel-Dateien können Zellen über Excel-Koordinaten (z.B. C1 oder AB24) adressiert werden. Dies verkürzt den Platzhalter auf %[object type]:[file name]:[coordinate]%.

Der Dateiname und die Spalten- und Zeilenparameter können auch über benutzerdefinierte Eigenschaften definiert werden. Zum Beispiel verweist *%itr-att:customers.csv:%job:Customer%: %job:CustomerNr.%%%* auf die Anlage customers.csv der aktuellen Iteration, wobei der Wert in der im Attribut *Customer* definierten Spalte des aktuellen Auftrags und der im Attribut *CustomerNr* definierten Zeile verwendet wird.



Bei Anhangsreferenzen wird die Groß-/Kleinschreibung nicht beachtet!

Die linke Seite einer Anhangsreferenz unterscheidet nicht zwischen Groß- und Kleinschreibung. %itr-att:kunden.csv:A1%, %ITR-att:kunden.csv:A1% und %ITR-Att:kunden.csv:A1% liefern dieselbe Ausgabe.

Kapitel 6. Definieren

Im Bereich *Definieren* können <u>Projekte</u>, <u>Iterationen</u>, <u>Anforderungen</u>, <u>Testumgebungen</u>, <u>Testsysteme</u>, <u>Testsegmente</u>, <u>Testfälle</u> und <u>Testsuiten</u> erstellt und bearbeitet werden. Für jeden dieser Objekttypen gibt es auf der linken Seite einen Menüeintrag.

6.1. Projekte

Ein *Projekt* vereint alle in einem Testprojekt benötigten und erzeugten Objekte wie Testfälle, Testläufe und Testergebnisse. In diesem Kapitel wird gezeigt, wie *Projekte* erstellt und bearbeitet werden.

6.1.1. Übersichtsseite

Die Übersichtsseite zeigt alle bereits erstellten Projekte in einer Tabelle an. Neue Projekte werden hier erstellt.

≡ 🙅 KLAR	os test					DE Finanz-Tra	cker 🖻 📑		<i>م</i> ×	0~ * ~
📝 Definieren	Projekte									
n Planen	Neu								Speichern	Verwerfen
			9 Einträge - Seite 1 von 1	н 🖪 🚺 🕨	10 🗸				Alle an	zeigen 🖧 ☰
Ausführen	ID 🗢 🛛 🖓	Beschreibung 🗢	Iterationen 🗢	Anforderungen 🗢	Testfälle 🗢	Testsuiten 🗢	Testläufe 🖨	Erstellt 🗢	Letzter Zugriff 🖨	Aktion
Ausiunien	O P00011		0	0	1	0	0	vor 2 Jahren	vor 5 Monaten	○ ໕ 🖨 🛍
A • • •	O P00010	Test PRV	0	0	5	1	3	vor 2 Jahren	vor 9 Monaten	o ピ 🖨 🛍
C Auswerten	O P00009	PRV 2	0	0	1	0	0	vor 2 Jahren	vor 2 Jahren	○ 🖒 🖨 🛍
	O P00008	PRV	0	0	3	1	1	vor 2 Jahren	vor 2 Jahren	o ピ 🖨 🛍
📕 Konfigurieren	O P00007	Issue Management Integration	0	0	4	0	0	vor 3 Jahren	vor 7 Monaten	○ 🖒 🖨 🛍
	• P00005	DE Finanz-Tracker	9	6	24	7	340	vor 4 Jahren	vor wenigen Sekunden	⊙ ໕ 🖨 🛍
	O P00003 8	Printer Tester	0	4	1	0	0	vor 4 Jahren	vor 7 Stunden	○ 🖒 🖨 🛍
	O P00002	Finance Tracker	9	6	24	7	338	vor 4 Jahren	vor 7 Stunden	o ピ 🖨 🛍
	O P00001	FinanceTrackerLocalTester	9	6	25	7	204	vor 4 Jahren	vor 7 Stunden	○ 🖒 🖨 🛍
	ID	Beschreibung	Iterationen	Anforderungen	Testfälle	Testsuiten	Testläufe	Erstellt	Letzter Zugriff	Aktion
	Neu								Speichern	Verwerfen

Abbildung 6.1. Die Übersichtsseite "Projekte"

Die Tabelle zeigt die folgenden Werte an:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Beschreibung	Die Beschreibung des Projekts.
Iterationen	Die Anzahl der Iterationen in diesem Projekt.
Anforderungen	Die Anzahl der Anforderungen in diesem Projekt.
Testfälle	Die Anzahl der Testfälle in diesem Projekt.
Testsuiten	Die Anzahl der Testsuiten in diesem Projekt.
Testläufe	Die Anzahl der Testläufe in diesem Projekt.
Erstellt	Das Datum, an dem das Projekt erstellt wurde.

Aktionen

Die ausführbaren Aktionen.

Der Eintrag Beschreibung lässt sich mit einem Klick direkt in den Tabellenzeilen bearbeiten.

6.1.1.1. Ein neues Projekt erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun kann die *Beschreibung* festgelegt werden.

Mit Speichern wird das neue Projekt erzeugt und gespeichert. Die ID des Projekts (P00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *P00001* gelangen Sie auf die Detailseite des Projekts.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs haben Änderungen und sind noch nicht gespeichert!

6.1.1.2. Aktionen

In der Aktionsspalte ganz rechts lassen sich folgende Aktionen vornehmen:

- Aktivieren
- 🛯 Bearbeiten
- 🔒 Druckansicht öffnen
- 🗊 Löschen

Wurde ein Projekt gelöscht, ist es zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Projekte stehen dann statt *Aktivieren* und *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ☆ Das Projekt unwiederbringlich aus der Datenbank entfernen

6.1.1.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- 公 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.1.1.4. Schnellauswahl

Um schnell zwischen Projekten zu wechseln, steht in der Kopfzeile auch eine Schnellauswahl zur Verfügung (siehe <u>Abbildung 6.2, "Schnelle Auswahl von Projekten"</u>).

≡ 🧶	ITR00001 - 1.0.x-Sprint 01-Grundfunktionen 🗸	DE Finanz-Tracker	✓ 🗉 🗏	Q 🚱 v	
- ~					

Abbildung 6.2. Schnelle Auswahl von Projekten

6.1.2. Detailseite

Jedes Projekt verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID des betreffenden Projekts oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Eigenschaften, Benutzerdefiniert, Objekte kopieren, Zugang, Integration, Ergebnisse und Änderungen.

= 👱 κιαr	o s test managemen				DE Finanz-Tracker 📑 📑		Q X	0 ∼ ≗ ∼
📝 Definieren	P00005 - DE Finanz-Tracker							👌 😋 😜
Planen	Eigenschaften Benutzerdefiniert (6)	Objekte kopieren	Zugang Integration	Ergebnisse	Änderungen	Speichern	Verwerfen	Zurück
💠 Ausführen	ID P00005							
욙 Auswerten	Beschreibung DE Finanz-Tracker Textfeldformat HTML V (2)							
🔑 Konfigurieren	Erstellt vor 4 Jahren von Felix Mustermann					Zuletzt geänder	t vor 10 Monaten von F	elix Mustermann
						Speichern	Verwerfen	Zurück

Abbildung 6.3. Die Seite "Projekte - Eigenschaften"

6.1.2.1. Aktionen

Auf den Detailseiten lassen sich jeweils folgende Aktionen vornehmen:

읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Projekts erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Druckansichten sind ausführlich in Abschnitt 5 2 3 2 1 Druck-

Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.



Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Projekte gewechselt werden.

6.1.2.2. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.3</u>) werden die folgenden Attribute des Projekts angezeigt und können bearbeitet werden:

Beschreibung	Die Beschreibung des Projekts.
Textfeldformat	Mehrzeilige Textfelder können HTML-Inhalte oder Klartext ent- halten. Das Format wird hier für das gesamte Projekt festge- legt.

6.1.2.3. Benutzerdefiniert

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Ein Projekt kann an spezifische Anforderungen angepasst werden, indem für Objekte wie z.B. Testfälle zusätzliche Felder definiert werden. Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.2.4, "Benutzerdefinierte Eigenschaften".

= 👱 K L A R (DS TEST MANAGEMENT	DE Finanz-Tracker 🖬 📑	५ x 0 v ≛ v
📝 Definieren	P00005 - DE Finanz-Tracker		👌 😏
🚢 Planen	Eigenschaften Benutzerdefiniert (6) Objekte kopieren Zugang Integration	Speichern Ver Ergebnisse Änderungen	werfen Zurück
🌣 Ausführen	Anforderung Aufgabe Iteration Testfall (1) Testfallergebnis Testlauf	Testschritt Testsegment Testsuite Testsystem (1)	stumgebung (4)
Auswerten	Name Geprüft von Zusätzliche Informationen bzw. Bemerkungen	Typ Werte Voreinst Text - Textfeld -	ellung Aktion 1 面
🖌 Konfigurieren	Laborexperiment	Boolescher Wert - V Aufzählung (5) 🗹 Temperatur < 0°C	1 创 1 创 1 创
	Erstellt vor 3 Jahren von Felix Mustermann	Zuletzt geänder Speichern Ver	t vor 2 Jahren von Selen Deutsch werfen Zurück

Abbildung 6.4. Die Ansicht "Benutzerdefinierte Eigenschaften"

6.1.2.4. Objekte kopieren



Objekte können zwischen Projekten kopiert oder synchronisiert werden. In der Ansicht *Objekte kopieren* wird eine Auswahlliste aller Quell-Projekte angezeigt, auf die der Benutzer Zugriff hat und die mindestens ein Objekt zum Kopieren enthalten. Die Checkbox *Benutzerdefinierte Felder übertragen* bestimmt, ob benutzerdefinierte Felder mit kopiert werden sollen.



Abbildung 6.5. Die Ansicht "Objekte kopieren"

Sobald ein Quell-Projekt ausgewählt wurde, erscheinen die Listen der Objekte, die kopiert werden können (Anforderungen, Testumgebungen, Testsysteme, Testfälle, Testsegmente und Testsuiten), sowie eine Auswahlmöglichkeit für den Modus (Kopieren oder Synchronisieren).

Es stehen zwei Modi zur Verfügung.

- Kopieren
- Synchronisieren

Kopierte Objekte haben keinen Verweis auf ihr Quell-Objekt, daher werden spätere Änderungen am Quell-Objekt nicht in die Kopien übertragen.

Synchronisierte Objekte haben einen Verweis auf ihr Quell-Objekt und können *nicht* bearbeitet werden. Änderungen an den Quell-Objekten können jederzeit in die Kopien übertragen werden.



Wozu werden synchronisierte Objekte benötigt?

Mit synchronisierten Objekte können Sie oft benötigte Testfälle in einem Referenzprojekt verwalten und in anderen Projekten wiederverwenden.

Bereits synchronisierte Objekte sind ausgeblendet und können mit Klick auf den Button Zeige alle angezeigt werden. Wenn für ein synchronisiertes Objekt Änderungen am Quell-Objekt vorliegen, wird das Icon () in der Spalte *Zusätzliche Hinweise* angezeigt. Mit Klick darauf öffnet sich ein Dialog, über den es synchronisiert werden kann.



Synchronisierungsmodus

Derzeit werden nur Testfälle und Testsuiten im Synchronisierungsmodus unterstützt.

Nachdem ein oder mehrere Objekte ausgewählt wurden, öffnet sich mit einem Klick auf den Button Kopieren (Synchronisieren) ein Dialog, in dem nochmals alle Objekte aufgelistet werden. Ein Klick auf Ok startet den Kopiervorgang (siehe <u>Abbildung 6.6, "Der Dialog "Objekte kopieren""</u>).



Abbildung 6.6. Der Dialog "Objekte kopieren"

Die Checkbox *Testfallstatus beibehalten* bestimmt, ob die gesetzten Testfallstatus mit kopiert oder stattdessen für alle Testfälle der Status *Entwurf* gesetzt werden soll.



Anmerkung

Wenn eine Testsuite kopiert wird, werden automatisch auch alle enthaltenen Testfälle kopiert.

6.1.2.5. Zugang

Ein Projekt kann vor dem Zugriff von außen geschützt werden, indem die Option Anonymen Projektzugriff deaktivieren gewählt wird.

Wenn diese Option aktiviert ist, muss ein Benutzer immer bei Klaros-Testmanagement angemeldet sein, um über Lesezeichen referenzierte Seiten anzuzeigen. Weitere Informationen zum Setzen von Lesezeichen finden Sie unter <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u>.

Wenn diese Option deaktiviert ist, ist eine über Lesezeichen referenzierte Seite für jedermann auch ohne Anmeldung erreichbar. Sobald der Benutzer dann versucht auf eine andere Seite zu wechseln, wird er auf den Login-Bildschirm umgeleitet. Ist ein Benutzer noch nicht angemeldet, wird dies im <u>Abschnitt 5.2.2.4</u>, <u>"Das Benutzermenü"</u> rechts oben durch ein entsprechendes Symbol angezeigt.

6.1.2.5.1. Projektspezifische Rollen zuweisen



E Y KLAROS TEST MANAGEMENT DE Finanz-Tracker E E	Q × Ø×	* ~
C Definieren P00003 - Printer Tester	a	Э Ө
Planen Eigenschaften Benutzerdefiniert Objekte kopieren Zugang (3) Integration Ergebnisse Änderungen	Verwerfen Zurü	ck
Ausführen Anonymer Zugang zum Projekt nicht erlaubt		
🕒 Auswerten 😫 — 3 Einträge-Seite 1 von 1 🔣 ┥ 💶 🕨 🕅 🔟 🕶	V d	\$ ≡
Benutzemane Voller Name Globale Rolle Proje	ektrolle 🗢	Aktion
Konfigurieren manager Max Mustermann lestmanager I Olluwe Krame Taster Terfer	~	_
Concernation Con Concernation Concernation Concernatio Concernation Concernation Concernation Co	· · · · · · · · · · · · · · · · · · ·	-
		Aktion
Erstellt vor 4 Jahren von Felix Mustermann Zuletzt geän Zuletzt geän Speichern Speichern	ndert vor 3 Jahren von selen2 : Verwerfen Zurü	selen2 ck



Klaros-Testmanagement Enterprise Edition verfügt über ein globales rollenbasiertes Zugriffssystem (siehe <u>Abschnitt 10.3, "Benutzer"</u>). Auf der Ansicht *Zugang* können Administratoren und Testmanager die globale Rolle eines Benutzers in eine individuelle Projektrolle ändern. Zum Beispiel kann ein Benutzer mit der globalen Rolle Tester einem Projekt als Tester und einem anderen Projekt als Testmanager zugewiesen werden.

Zu Beginn ist jedes Projekt für jeden Benutzer zugänglich. Wenn einem Projekt mindestens ein Testmanager zugewiesen wurde, ist der Zugriff auf dieses Projekt für alle anderen nicht explizit zugewiesenen Benutzer eingeschränkt.

Mit Klick auf den Button Zuweisen öffnet sich ein Dialog (siehe <u>Abbildung 6.8, "Der Dialog "Projektrolle zuweisen""</u>), in dem neue Benutzer zu diesem Projekt hinzugefügt werden können.

= 📌 KLAR	0 S	TEST M	A N A G E M E	ΝT				DE Finanz-1	'racker 🖬 🗮		Q	× Ø×	* ~
📝 Definieren	P0	0003 - Printer T	ester									Ð	GÐ
Planen	Eig	genschaften Be	enutzerdefiniert	Objekte kopiere	en Zugang (3)	Integration	Ergebniss	e Änderungen		Speichern	Verwerfen	Zuri	ück
🏟 Ausführen		nonymer Zugang zu	ım Projekt nicht erla	ubt									
🚯 Auswerten		Projektrolle zuweise	en nden Benutzern wird	in diesem Projekt	die unten ausgewäh	lte Rolle zugewi	esen.	×			ktrolle 🗘	₹ •	Aktion
<i>F</i> Roninguneren		8 von 8 ausgewählt		8	Einträge - Seite 1 von 1	H 🖣 🚺	▶ ₩ [10 🗸		7 & ≡		~	-
		≤	Benutzername	÷		Voller Name 🗘			Globale Rolle	¢			Aktion
	Ers	✓ gross			Günther Gross			Gast			ndert vor 2 Jah	ren von seler	12 selen2
		 Meyer 			Markus Meyer			Testmanager					
		🖌 sahra			Sahra Berger			Tester			Verwerfen	Zuri	ück
		✓ Tafel			Thomas Tafel			Tester					
		✓ tester			Erika Mustermann			Tester					
		✓ Thiel			Tim Thiel			Tester					
		 Toppler 			Tanja Toppler			Tester					
		 Tunklik 			Timo Tunklik			Tester					
		Benutzer hinzufü	gen als Testmanaç	er 🗸									
									ОК	Abbrechen			

Abbildung 6.8. Der Dialog "Projektrolle zuweisen"



Jedes Projekt benötigt mindestens einen Testmanager!

Ein Projekt erfordert mindestens einen zugewiesenen Testmanager. Wird versucht ein Projekt ohne zugewiesenen Testmanager zu speichern, wird im LogPanel eine Warnung angezeigt (siehe <u>Abschnitt 5.2.2.2, "Das Log-Panel"</u>).

Folgende Massenaktionen lassen sich hier vornehmen:

& Benutzerrolle im Projekt ändern

ff Löschen

6.1.2.6. Integration

In dieser Ansicht lassen sich externe Systeme wie Issue Management und Anforderungsmanagement mit Klaros-Testmanagement verknüpfen.

6.1.2.6.1. Issue Management

Hier können die im Projekt verwendeten Issue-Management-Systeme bearbeitet werden.



Ein Issue-Management-System anlegen

Bevor ein Issue-Management-System auf dieser Seite verfügbar ist, muss es zunächst von einem Administrator angelegt werden.

Eine Anleitung zum Anlegen von Issue-Management-Systemen finden Sie unter <u>Abschnitt 10.5.1, "Issue Management"</u>.

Die Ansicht zeigt zwei Tabellen: Die linke Tabelle enthält die verfügbaren Issue-Management-Systeme, während die rechte Tabelle zeigt die Issue-Management-Systeme, die dem Projekt bereits zugewiesen sind. Durch Drücken des Icons > wird das entsprechende Issue-Management-System dem Projekt hinzugefügt. Um alle Issue-Management-Systeme hinzuzufügen, drücken Sie das Icon ». Die Icons < und « werden auf die gleiche Weise verwendet, um die Zuordnung von Issue-Management-Systeme aufzuheben. Es ist auch möglich, ein Problemverwaltungssystem per Drag & Drop in eine Liste zu ziehen.

Mit den Icons in der Mitte können folgende Aktionen vorgenommen werden:

- > Das ausgewählte Issue-Management-System zuordnen.
- » Alle Issue-Management-Systeme zuordnen.
- « Die Zuordnung für alle Issue-Management-Systeme aufheben.
- < Die Zuordnung des ausgewählten Issue-Management-Systems aufheben.



Abbildung 6.9. Die Ansicht "Integration/Issue Management"

6.1.2.6.2. Anforderungsmanagement



Weitere Informationen zum Verbinden entfernter Anforderungsmanagementsysteme mit Klaros-Testmanagement finden Sie unter <u>Abschnitt 10.5.2</u>, <u>"Anforderungsmanagement"</u>. Ein Anforderungsmanagement-System anbinden

≡ 👱 K L A R	os test managem	ENT	DE Finanz-Tracker 👅 🗮	५ × 0 × ≗×
📝 Definieren	P00003 - Printer Tester			🔁 😌 🗲
nter Planen	Eigenschaften Benutzerdefinier	: Objekte kopieren	Zugang (3) Integration Ergebnisse Änderungen	Verwerfen Zurück
🌣 Ausführen	Issue Management Anforderu	ngsmanagement (1)		
🛟 Auswerten 🌽 Konfigurieren	Anforderungen synchronisieren mit	RM00001 (SYNC) V Jira (SYNC) Task X Sub-task	X v	
	Zuordnungen			
	Remote-Feld ≜	Typ 🌲	11 Einträge - Seite 1 von 2 N 4 1 2 N 10 V	Y 选 ≡
	Attachment	Enum		✓
	BUGTESt	String		~
	Description	Text		~
	Fix Version/s	Enum		~
	Issue Type	Text		~
	KASSEL_TEST	Text		~
	Parent	Text		~
	Priority	Enum Prioritä	t	✓
	Project	Text		~
	Status	String		~
	Erstellt vor 4 Jahren von Felix Musterman	1	Zuletzt geän Speichern	lert vor 3 Jahren von selen2 selen2 Verwerfen Zurück

Abbildung 6.10. Die Ansicht "Integration/Anforderungsmanagement"

Ein Projekt kann mit einem Anforderungsmanagementsystem über die Ansicht Anforderungsmanagement auf der Projektdetailseite verbunden werden. Auf dieser Ansicht wird eine Dropdown-Liste mit allen verfügbaren Anforderungsmanagementsystemen angezeigt.



Wichtig

Um ein Anforderungsmanagementsystem mit einem Projekt zu verbinden, muss das System im Bereich *Konfigurieren* angelegt werden (siehe <u>Abschnitt 10.5.2</u>, <u>"Anforderungsmanagement"</u>).

Mit Klick auf den Button <u>Speichern</u> wird eine Synchronisation im Hintergrund gestartet. Dabei werden alle Anforderungen aus dem konfigurierten Anforderungsmanagementsystemen geladen und lokal gespeichert.



Anmerkung

Das Synchronisieren von Anforderungen kann je nach Anzahl der Anforderungen auf der externen Seite sehr viel Zeit in Anspruch nehmen. Ein eingebauter Scheduler synchronisiert alle Anforderungen aller angeschlossenen Anforderungsmanagementsysteme für alle freigegebenen Projekte in festgelegten Intervallen.



Warnung

Beim Aktivieren eines Remote-RMS werden alle bereits erstellten lokalen Anforderungen deaktiviert. Beim Zurückschalten auf die lokale Anforderungsverwaltung werden diese Anforderungen reaktiviert.

Issue-Typen verlinken

Nach der Auswahl eines Anforderungsmanagementsystems aus der Dropdown-Liste wird eine Liste der Anforderungstypen angezeigt. Standardmäßig sind alle Typen ausgewählt, sodass bei einer Hintergrundsynchronisierung Anforderungen jeglichen Typs synchronisiert werden.

Eigenschaften	Benutz	zerdefiniert	Objekte kopiere	n Zugang ((3) Integration	Ergebnisse	Änderungen	
Issue Manage	ement	Anforderung	gsmanagement					
Anforderunger Zu syno	n synchro chronisier	nisieren mit rende Typen	RM00001 (SYNC) 🔪 Task 🗶 Sub-tas	* k X V				🔷 Jira (SYNC)

Abbildung 6.11. Die synchronisierten Typen des Anforderungsmanagementsystems

Felder verknüpfen

Administratoren und Projektleiter können einzelne Felder zwischen dem entfernten Anforderungsmanagementsystem und Klaros-Testmanagement verknüpfen. Bei Verwendung von Jira werden standardmäßig die Felder *Beschreibung*, *Priorität* und *Zusammenfassung*-Felder aus dem entfernten RMS (wenn sie verfügbar sind) mit dem *Beschreibung*, *Priorität* und *Name* Feldern in Klaros-Testmanagement verknüpft. Darüber hinaus wird das Jira-Feld *Status* automatisch einem internen *Status*-Feld zugeordnet, das nur angezeigt wird, wenn die RMS-Synchronisation aktiviert ist.

Felder können von einem der drei folgenden Typen sein: String, Aufzählung und Boolean.



Anmerkung

Ein Aufzählungsfeld in Klaros-Testmanagement kann nur mit einem Aufzählungsfeld des entfernten RMS verknüpft werden, aber ein Aufzählungsfeld des RMS kann entweder mit einem String- oder Boolean-Feld in Klaros-Testmanagement verknüpft werden.

Verknüpfen von Aufzählungswerten

Bei der Verknüpfung von Aufzählungen ist es notwendig anzugeben, welche Werte der Aufzählung miteinander verknüpft werden sollen. Andernfalls werden alle Werte des RMS auf den Standard-Wert des Klaros-Testmanagement Feldes abgebildet.

Durch Anklicken des Icons
☐ einer Aufzählungsverknüpfung öffnet sich ein Dialog, in dem einzelne Werte verknüpft werden können.

= 🞐 KLAR	o s test mana	GEMENT			DE Finanz-Tracker			৹ @৵ ≛৵
📝 Definieren	P00003 - Printer Tester							🔁 😋 🕤
						Speichern	Verwerfen	Zurück
Planen	Eigenschaften Benutzerd	lefiniert Objekte ko	pieren Zugang (3)	ntegration Ergebnisse	Änderungen			
🏟 Ausführen	Issue Management An	forderungsmanageme	nt					
 Auswerten Konfigurieren 	Anforderungen synchronisie Zu synchronisierend	eren mit RM00001 (SY e Typen Task X S	AC) V Jb-task X V					Jira (SYNC)
	Verknüpfen von Auswahlwe	rten - Priority/Priority						7 & ≡
	Remote Au	swahlwert 🗘		Lokaler Ausv	vahlwert 🗘		1	Auto Sync 🗢 Aktion
	Blocker		Hoch			~	~	
	Critical		Hoch			~	~	
	Major		Mittel			~	~	
	Minor		Niedrig			~	~	
	Trivial		Niedrig			~	~	
					Speichern	Abbrechen	~	
	Parent	Text					~	
	Priority	Enum	Priorität				~	ľ
	Project	Text					~	
	Status	String					~	
	Summary	Text					~	
	Erstellt vor 10 Monaten von Felix M	lustermann				Zuletzt g Speichern	eändert vor 30 1 Verwerfen	agen von selen2 selen2 Zurück

Abbildung 6.12. Verknüpfen von Aufzählungswerten



Anmerkung

Mehrere Enum-Werte eines Remote-Enum-Feldes können mit demselben Enum-Wert eines Feldes in Klaros-Testmanagement verknüpft werden.

Synchronisieren von Aufzählungswerten

Beim Verknüpfen von Aufzählungsfeldern ist es oft mühsam, die Liste der auswählbaren Werte synchron zu halten. Hierfür steht mit der Spalte *Auto Sync* ein automatischer Synchronisationsmechanismus für Aufzählungswerte zur Verfügung.

Die Verwendung von AutoSync erfordert, dass das Remote-Feld ein Aufzählungsfeld ist. Außerdem muss das lokale Feld ein benutzerdefiniertes Aufzählungsfeld sein. Sobald dies erfüllt ist, wird das Feld *Auto Sync* mit einem Häkchen in der Liste angezeigt.



Anmerkung

Es ist nicht notwendig, alle Werte der Aufzählungsfelder anzugeben. Sobald das Flag *Auto Sync* gesetzt ist, wird das entfernte System nach verfügbaren Werten abgefragt und die Werte für das lokale Feld werden automatisch entsprechend geändert.

Ändern des angeschlossenen Anforderungsmanagement-Systems

Das Anforderungsmanagementsystem (RMS) eines Projekts kann jederzeit gewechselt werden. Alle aktuell vorhandenen Anforderungen des Projekts werden deaktiviert, solange ein anderes RMS verwendet wird.

6.1.2.7. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei weitere Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Project zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.1.2.8. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Projekts an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

6.2. Iterationen



Eine *Iteration* stellt einen Testzyklus in einem Projekt dar und erlaubt es den Testprozess mit einem agilen Entwicklungsprozess (z.B. Scrum) zu synchronisieren. In diesem Kapitel wird gezeigt, wie *Iterationen* erstellt und bearbeitet werden.

6.2.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Iterationen in einer Tabelle an. Neue Iterationen werden hier erstellt.

≡ 👱 KLAR	os test m	ANAGEMENT	DE Finanz-Tracker 🔳 🗮	२ × 0 ≁ ≛≁
📝 Definieren	Iterationen			
📇 Planen	Neu			Speichern Verwerfen
		Sentrage - Sente i Von i Name	Start & Föllig &	Testläufe 🛔 🛛 Aktion
🏩 Ausführen	O ITR00009	2.1.x-Sprint 01-Optimierung für die Verwendung auf Smartwatches	30.06.2020 30.07.2020	
	O ITR00008	2.0.x-Sprint 02-Dashboard erweitern	16.06.2020 30.06.2020	5 0 C C C C
🕒 Auswerten	O ITR00007	2.0.x-Sprint 01-Dashboard hinzufügen	29.05.2020 18.06.2020	3 0 6 6 4 1
	O ITR00006	1.1.x-Sprint 04-Optimierung für Tablets	06.05.2020 21.05.2020	11 O 🗹 🖓 🛱 🛍
🔎 Konfigurieren	O ITR00005	1.1.x-Sprint 03-Annullierung von Einzugsermächtigungen	30.04.2020 07.05.2020	8 O 🗹 🖵 🛱 🛍
	O ITR00004	1.1.x-Sprint 02-Daueraufträge bearbeiten	31.03.2020 29.04.2020	6 O 🖸 🖵 🛱 🛍
	O ITR00003	1.1.x-Sprint 01 - Übersichten	10.03.2020 31.03.2020	8 O 🖸 🖨 🛍
	O ITR00002	1.0.x-Sprint 02-Erweiterung der Grundfunktionen	07.02.2020 29.02.2020	30 0 ピ 🗗 🖨 🛍
	O ITR00001	1.0.x-Sprint 01-Grundfunktionen	27.01.2020 12.02.2020	77 🛛 🖒 🖓 🖨 🛍
	ID	Name	Start Fällig	Testläufe Aktion
	Neu			Speichern Verwerfen

Abbildung 6.13. Die Übersichtsseite "Iterationen"

Die Tabelle zeigt die folgenden Werte an:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Name	Der editierbare Name der Iteration.
------------	---------------------------------------------
Startdatum	Das Datum, an dem die Iteration beginnt.
Fällig	Das Datum, an dem die Iteration enden soll.
Testläufe	Die Anzahl der Testläufe.
Aktionen	Die ausführbaren Aktionen.

Die Einträge *Name, Startdatum* und *Fällig* lassen sich mit einem Klick in die entsprechenden Felder direkt in den Tabellenzeilen bearbeiten.

6.2.1.1. Eine neue Iteration erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun können Name, Startdatum und Fällig festgelegt werden.

Mit Speichern wird die neue Iteration erzeugt und gespeichert. Die ID der Iteration (ITR00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *ITR00001* gelangen Sie auf die Detailseite der Iteration.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.



Rote IDs

Alle Zeilen mit roten IDs haben Änderungen und sind noch nicht gespeichert!

Mit Verwerfen werden alle Änderungen rückgängig gemacht.

6.2.1.2. Aktionen

In der Aktionsspalte ganz rechts lassen sich folgende Aktionen vornehmen:

- •; Aktivieren
- 🛯 Bearbeiten
- Duplizieren
- 🔒 Druckansicht öffnen
- 🗊 Löschen

Wurde eine Iteration gelöscht, ist sie zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Iterationen stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ☆ Die Iteration unwiederbringlich aus der Datenbank entfernen

6.2.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Iterationen gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Iterationen aus, auf die die Aktion angewendet werden soll.

Für Iterationen werden folgende Massenaktionen unterstützt:

- 🛯 Bearbeiten
- Duplizieren
- 🔒 Druckansicht öffnen
- ff Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Die Iteration unwiederbringlich aus der Datenbank entfernen (nur Administrator).
- 🗀 Einer Kategorie zuordnen (erscheint erst, nachdem eine Kategorie angelegt wurde).

Massenaktionen sind ausführlich in Abschnitt 5.2.3.1.5, "Massenaktionen" beschrieben.

6.2.1.4. Tabellenoperationen

- In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:
 - 옮 Kategorisieren
 - Filter / Sortieren
 - Zeige alle / Nur aktive (Nur Administrator)
 - Q Suchen
 - & Exportieren
 - \equiv Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

6.2.1.5. Schnellauswahl

Um schnell zwischen Iterationen zu wechseln, steht in der Kopfzeile auch eine Schnellauswahl zur Verfügung (siehe <u>Abbildung 6.14, "Schnelle Auswahl von Projekten und Iterationen"</u>).



Abbildung 6.14. Schnelle Auswahl von Projekten und Iterationen

Wenn eine Iteration aktiv ist, sind nur die Testobjekte sichtbar, die für den durch die Iteration repräsentierten Testzyklus relevant sind. Wenn ein Testlauf ausgeführt wird, während eine Iteration aktiv ist, wird er automatisch mit der Iteration verknüpft. Dies hilft dabei, die Testergebnisse eines bestimmten Testzyklus von anderen Aktivitäten zu trennen.

6.2.2. Detailseite

Jede Iteration verfügt über eine eigene Detailseite mit mit mehreren weiteren Unteransichten. Mit Klick auf die ID der betreffenden Iteration oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Anhänge, Testumgebungen, Testsysteme, Anforderungen, Aufgaben, Ergebnisse und Änderungen.





6.2.2.1. Aktionen

Auf den Detailseiten lassen sich jeweils folgende Aktionen vornehmen:

읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der Iteration erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.	
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.	
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.	
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.	
G 🕤 Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Iterationen gewechselt werden.	

6.2.2.2. Übersicht

Alle hier angezeigten Daten beziehen sich immer auf die momentan angezeigte Iteration.

Auf der Ansicht Übersicht werden folgende Werte angezeigt:

ErfolgDie Gesamterfolgsrate aller Testfälle. Eine Erfolgsrate von 100
% bedeutet, dass der letzte Testlauf für jeden Testfall erfolg-
reich war. Auch wenn ein Testfall in der Vergangenheit bereits
erfolgreich ausgeführt wurde, wird nur das letzte Ergebnis für
die Erfolgsrate berücksichtigt.



Abbildung 6.16. Die Erfolgsrate der Iteration

Fortschritt	Die Fortschrittsrate zeigt an, wie viele Testfälle ausgeführt wurden. Im Gegensatz zur Erfolgsrate wird bei der Fortschritts- rate nur berücksichtigt, ob ein Testfall mindestens einmal in der Iteration ausgeführt wurde, unabhängig von dessen Ergeb- nis.	
Konformität	Die Konformitätsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle erfolgreich ausgeführt wurden.	
Abdeckung	Die Abdeckungsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle mindestens einmal ausgeführt wurden. Im Gegensatz zur Konformitätsrate berücksichtigt die Überde- ckungsrate nur, ob ein Testfall mindestens einmal in der Itera- tion ausgeführt wurde, unabhängig von dessen Ergebnis.	
Startdatum	Das Datum, an dem die Iteration beginnt.	
Fällig	Das Datum, an dem die Iteration enden soll.	
Ausführungszeit manuell und automatisiert	Die Gesamtausführungszeiten aus manuellen und automati- sierten Testausführungen der Iteration. Die Werte können für weitere Untersuchungen verwendet werden. Eine extrem hohe manuelle Ausführungszeit könnte z. B. auf eine hohe Rate an Wiederholungstests oder auf einen Bedarf an mehr automati- sierten Testfälle hinweisen.	
Testsystem Radarchart	Dieses Diagramm zeigt die Erfolgs- und Fortschrittsrate dieser Iteration für jede Kombination von Testumgebung und zu tes- tendem System, die für diese Iteration konfiguriert wurden.	



Testsystem Finance Tracker 1.1.0 Projekt DE Finanz-Tracker , Iteration 1.1.x-Sprint 01 - Übersichten





Anmerkung

Alle Testfälle, die durch Anforderungen Teil der Iteration sind, tragen zu den Metriken dieser Ansicht bei. Sind keine Testfälle für diese Iteration ausgewählt, werden alle Testfälle des Projekts berücksichtigt.

6.2.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.18</u>) werden die folgenden Attribute der Iteration angezeigt und können bearbeitet werden:

Beschreibung	Eine Beschreibung der Iteration.
Erfolgskriterien	Eine Beschreibung der zu erfüllenden Bedingungen, damit diese Iteration abgeschlossen ist.
Startdatum	Das Datum, an dem die Iteration beginnt.
Fälligkeit	Das Datum, an dem die Iteration enden soll.



Abbildung 6.18. Die Seite "Iterationen - Eigenschaften"

6.2.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.2.2.5. Anhänge

Einer Iteration können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.2.2.6. Testumgebungen

Möchten Sie Testumgebungen mit der Iteration verknüpfen, klicken Sie auf den Button Zuordnen und wählen Sie ein oder mehrere Testumgebungen aus.

6.2.2.7. Testsysteme

Die Seite listet alle der Iteration zugeordneten Testsysteme mit ihren jeweiligen Stati auf:

ID	Wird automatisch zugewiesen.	
Version	Die Version des Testsystems.	
Erfolg	Der Prozentsatz der bestandenen Tests für dieses Testsys- tems, die in dieser Iteration ausgeführt wurden.	
Fortschritt	Der Prozentsatz der in dieser Iteration für dieses Testsystem ausgeführten Tests.	
Konformität *	Der Prozentsatz der für dieses Testsystem ausgeführten Tests, die die dieser Iteration zugeordnete Anforderungen abdecken.	
Abdeckung*	Der Prozentsatz der für dieses Testsystem ausgeführten Tests, die die dieser Iteration zugeordnete Anforderungen abdecken und zuletzt das Ergebnis <i>Bestanden</i> hatten.	

* Nur angezeigt, wenn mindestens eine Anforderung mit der Iteration verknüpft ist.

Möchten Sie die Testsysteme mit der Iteration verknüpfen, klicken Sie auf den Button Zuordnen und wählen Sie ein oder mehrere Testsysteme aus.

6.2.2.8. Anforderungen

Die Ansicht Anforderungen zeigt die Anforderungen, die dieser Iteration zugeordnet wurden sowie ihre Konformitäts- und Abdeckungs-Rate. Diese Metriken beziehen sich auf die mit den Anforderungen verknüpften Testfälle und die letzten Ergebnisse dieser Ausführungen. Es werden die folgenden Werte angezeigt:

ID	Wird automatisch zugewiesen.
Revision	Die Revision der Anforderung.
Name	Der Name der Anforderung
Konformität	Der Prozentsatz der ausgeführten Testfälle, die die Anforde- rungen dieser Iteration abdecken.
Abdeckung	Der Prozentsatz der ausgeführten Testfälle, die die Anforde- rungen dieser Iteration abdecken und zuletzt das Ergebnis <i>Bestanden</i> hatten.

Möchten Sie die Anforderungen mit der Iteration verknüpfen, klicken Sie auf den Button Zuordnen und wählen Sie ein oder mehrere Anforderungen aus.

6.2.2.9. Aufgaben

Die Seite zeigt eine Liste der zur ausgewählten Iteration zugeordneten Aufgaben an.

In der Tabelle werden die folgenden Werte angezeigt:

ID	Wird automatisch zugewiesen.	
Zusammenfassung	Die Zusammenfassung der Aufgabe	
Priorität	Die Dringlichkeit der Aufgabe. Mögliche Werte sind Trivial, Niedrig, Hoch, Kritisch, Blocker	
Status	Der Status der Aufgabe. Mögliche Werte sind Neu, Wieder eröffnet, In Arbeit, Erledigt, Geschlossen, Abgelehnt.	
Fortschritt	Der Prozentsatz der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.	
Erfolg	Die Erfolgsrate der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.	
Fällig	Das Datum, bis zu dem diese Aufgabe beendet sein soll.	
Zugewiesen	Der Benutzer, dem die Aufgabe zugewiesen ist.	

Aktion

Zur Verfügung stehen die Aktionen 😰 Bearbeiten, 🖨 Druckansicht öffnen und 🏚 Ausführen. Die für Ausführen möglichen Darstellungen finden Sie unter <u>Abschnitt F.2.2, "Ausfüh-</u> rungsaktionen".

6.2.2.10. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei weitere Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Iteration zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.2.2.11. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieser Iteration an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

6.3. Anforderungen



Eine *Anforderung* ist eine vorgegebene, zu erfüllende Bedingung, die Eigenschaften oder zu erbringende Leistung eines Produktes, eines Prozesses oder eines Systems vorgibt und beschreibt. In diesem Kapitel wird gezeigt, wie Anforderungen erstellt und bearbeitet werden.

6.3.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Anforderungen in einer Tabelle an. Neue Anforderungen werden hier erstellt.

= 👱 K L A R (OS TEST MANAGEMENT DE Finanz-Tracker		۹ ،	< 0× ±×
📝 Definieren	Anforderungen			
• Dianon	Neu		Speichern	Verwerfen
	+ 🖸 🗋 👂 🛍 6 Einträge - Seite 1 von 1 🕅 ┥ 🧻 🕨 🕅 🔽	유 \ Alle anzeig	en	Q × & ≡
📩 Ausführen	ID ◆ Q Revision Name ◆	Priorität 🖨 S	Status 🗘 Testfälle 🖨	Aktion
Ausiunien	R00006 🔗 1.0 Das Dashboard wird korrekt aktualisiert, nachdem etwas geändert wurde.	Mittel E	Entwurf 4	╔╻ᇢ╓
	R00005 🔗 1.2 Die Anwendung ist mit Android-Smartphones, Tablets und Smartwatches kompatibel	Hoch E	Entwurf 5	┏┍₽₫
C Auswerten	R00004 1.0 Es ist erforderlich, dass sich die Datenbank immer in einem konsistenten Zustand befindet.	Hoch E	Entwurf 0	₡₽₽₫
	🗌 R00003 🔗 1.3 Mindestens 100 Benutzer sollten die Anwendung gleichzeitig ohne Leistungsverluste nutzen können.	Hoch E	Entwurf 8	₡ 🛛 🖨 🛍
🎤 Konfigurieren	R00002 🔗 1.0 Kein Verbindungsfehler.	Niedrig E	Intwurf 2	嫦健Ә₫
	R00001 🔗 1.0 Die Verarbeitung der Ergebnisse der Benutzereingaben dauert weniger als 4 Sekunden.	Mittel E	Intwurf 2	嫦健Әѿ
	ID Revision Name			Aktion
	Neu		Speichern	Verwerten

Abbildung 6.19. Die Übersichtsseite "Anforderungen"

Die Tabelle zeigt die folgenden Werte:

ID

Wird automatisch zugewiesen.

Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision der Anforderung.
Name	Der Name der Anforderung.
Priorität	Die Priorität der Anforderung. Mögliche Werte sind <i>Niedrig</i> , <i>Mittel</i> und <i>Hoch</i> .
Status	Der Status der Anforderung. Mögliche Werte sind Gesperrt, Genehmigt, Entwurf und Auslassen.
Testfälle	Die Anzahl der Testfälle, die dieser Anforderung zugeordnet sind.
Aktion	Die ausführbaren Aktionen.

Die Einträge *Name, Priorität* und *Status* lassen sich mit einem Klick in die entsprechenden Felder direkt in den Tabellenzeilen bearbeiten.

6.3.1.1. Eine neue Anforderung erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun können Name, Priorität und Status festgelegt werden.

Mit Speichern wird die neue Anforderung erzeugt und gespeichert. Die ID der Anforderung (R00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *R00001* gelangen Sie auf die Detailseite der Anforderung.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst mit Klick auf Speichern werden die Daten in die Datenbank übernommen.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Ungespeicherte Änderungen

In allen mit roten IDs versehenden Zeilen befinden sich Änderungen, die noch nicht gespeichert sind.

6.3.1.2. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

🖉 Bearbeiten

Duplizieren

🔒 Druckansicht öffnen

🗊 Löschen

Wurde eine Anforderung gelöscht, ist sie zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Anforderungen stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ⊘ Die Anforderung unwiederbringlich aus der Datenbank entfernen
- 6.3.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Anforderungen gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Anforderungen aus, auf die die Aktion angewendet werden soll.

Für Anforderungen werden folgende Massenaktionen unterstützt:

- 🛯 Bearbeiten
- Duplizieren
- Neue Revision erzeugen
- 🔒 Druckansicht öffnen
- fi Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Die Anforderung unwiederbringlich aus der Datenbank entfernen (nur Administrator).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

6.3.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 品 Kategorisieren
- √ Filter / Sortieren
- Zeige alle / Nur aktive (Nur Administrator)
- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.3.1.5. Remote-Synchronisation

Ist das aktive Projekt mit einem Anforderungsmanagementsystem verbunden, ist das Bearbeiten von Anforderungen für alle Benutzer deaktiviert. Oben rechts auf dem Bildschirm der Anforde-

rungsliste erscheint das Icon \mathcal{C} . Mit Klick auf das Icon wird eine Hintergrundsynchronisation der Anforderungen vom Anforderungsmanagementsystem zum Projekt durchgeführt.

🔳 📌 KLAR(S TEST MANAGEMENT	Printer Tester 🖬 🗮	۹× 0√ ≛۰
📝 Definieren	🖗 11:52:39 Das zur Zeit ausgewählte Projekt ist: Printer Tester (P00003)		+ ×
n Planen	Anforderungen		S
🏟 Ausführen	+ ⊖ 4 Einträge - Seite 1 von 1 K ◀ 1	▶ N 10 ▼ e ✿ Priorit	옮♡ Q x & =
-	R00004 1.0 SYNC-4 It is required that the database is always in a consi	tent state Hoch	
🕒 Auswerten	R00003 1.0 SYNC-3 At least 100 users should be able to use the applications in a column set.	tion simultaneously without any performance Mittel	0 To Do Q 🖨
	R00002 1.0 SYNC-2 No connection error.	Mittel	о то Do 🔍 🖨
差 Konfigurieren	R00001 1.0 SYNC-1 User input results takes less than 4 seconds to pro-	cess. Mittel	о то До 📿 🖨
	ID Revision Externe ID Na	ne Prior	ität Testfälle Externer Status Aktion

Abbildung 6.20. Die Seite "Anforderungen" mit Remote-Synchronisation

6.3.2. Detailseite

Jede Anforderung verfügt über eine eigene Detailseite mit weitergehenden Unteransichten. Mit Klick auf die ID der betreffenden Anforderung oder auf das Icon
rechts in der Aktionsspalte gelangen Sie zu der als letztes ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Anhänge, Iterationen, Testfälle, Revisionen, Ergebnisse und Änderungen.

6.3.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

Aktualisieren	Synchronisiert die lokale Anforderung mit der externen Quelle.				
🛃 Externen Link öffnen	Zeigt die Anforderung in der externer Quelle.				
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der Anforderung erstellt werden. Mit Klick auf das Icon 合 öffnet sich diese in einem neuen Browsertab.				
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.1, "Druckansichten"</u> .				
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hy link erreicht werden. Mit Klick auf das Icon 口 wird dieser in die Zwischenablage kopiert.				
	Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.2.2, "Lesezeichen".				



Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Anforderungen gewechselt werden.

6.3.2.2. Übersicht

≡ 📌 K L A R) S TEST MANAGEMENT DE Finanz-Tracker 🖀 🗮 🔍 Q 🗙 (0× ± ×
C Definieren	R00005 - 1.2 - Die Anwendung ist mit Android-Smartphones, Tablets und Smartwatches kompatibel	
😂 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Iterationen (9) Testfälle (5) Revisionen (3) Ergebnisse Änderungen	Zurück
🏟 Ausführen	Konformität 75% Priorität Hoch	
🕒 Auswerten	Abdeckung 60% Status Entwurf Automatisierungsgrad 0%	
🖌 Konfigurieren	Revision 1.2 V Testsystem Finance Tracker 2.0.0 V Testumgebung Android 10 Smartphone V	
	Testüberdeckungsverlauf (Anforderung Die Anwendung ist mit Android-Smartphones, Tablets und Smartwatches kompatibel, Letzte 30 Tage) Projekt DE Finanz-Tracker Testsystem Finance Tracker 2.0.0, Testumgebung Android 10 Smartphone	
	100 Konformität Abdeckung	
	90	
	80	
	70	
	50 50	
	40	
	30	
	20	
	2020 2021 2022	2023
	Zuletzt ausgeführte Testfälle	
	4 Einträge- Seite 1 von 1 🕅 🖣 🎦 🕨 M 10 🗸 🖓 🔍	× & ≡
	ID 🗣 🔍 Name 🗢 Start 🗣 Ausgeführt von 🕈 Testlauf 🕈 🛛 Testungebung 🗢 🛛 Testsystem 🗢 Dauer 🗣 Ergebn	iis 🗢 Aktion
	TC00015 Fügen Sie einen Dauerauftrag hinzu. vor 3 Jahren Felix Mustermann TRU0001721 Android 10 Smartphone Finance Tracker 2.0.0 00:00:08 🛕	Q
	TC00021 Aktualisieren Sie das Dashboard durch Stornierung vor 3 Jahren Felix Mustermann TRU0001697 Android 10 Smartphone Finance Tracker 2.0.0 00:00:09 🔮	Q
	TC00016 Daueraufträge löschen. vor 3 Jahren Talal Artf TRU0000226 Android 10 Smartphone Finance Tracker 2.0.0 00:00:27 🤗	Q
	TC00017 Lastschriftmandat löschen. vor 3 Jahren Talal Arif TRU0000226 Android 10 Smartphone Finance Tracker 2.0.0 00:00:16 🦿	Q
	Alle anzeigen	

Abbildung 6.21. Die Ansicht "Übersicht"

Auf der Ansicht Übersicht werden folgende Werte angezeigt:

Konformität	Die Konformitätsrate zeigt an, wie viele dieser Anforderung zugeordneten Testfälle mit dem Ergebnis <i>Bestanden</i> durchge- führt wurden.
Abdeckung	Die Abdeckungsrate zeigt an, wie viele dieser Anforderung zugeordneten Testfälle bereits durchgeführt wurden.
Automatisierungsgrad	Der Automatisierungsgrad zeigt an, wie viel Prozent der Test- fälle automatisiert durchgeführt werden können.
Priorität	Die Priorität der Anforderung. Mögliche Werte sind <i>Niedrig, Mittel</i> und <i>Hoch</i>
Status / Externer Status	Der Status der Anforderung. Wenn die Anforderung von einem externen System synchronisiert wird, wird hier der extern ver- waltete Status angezeigt.
	Andernfalls sind die möglichen Werte Gesperrt, Genehmigt, Entwurf und Auslassen.

Testüberdeckungsverlauf ChartDieses Diagramm zeigt die Konformitäts- und Abdeckungsra-
te dieser Anforderung für die hier eingestellte Kombination aus
Revision, Testsystem und Testumgebung.

Diese Grafik zeigt, wie sich die Konformität und der Abdeckungsgrad für diese Anforderung seit ihrer Definition verändert haben. Die Zeitleiste wird automatisch an die vorhandenen Informationen angepasst.

Im unteren Bereich befindet sich eine Tabelle mit der Übersicht über die zuletzt ausgeführten Testfälle.

6.3.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.22</u>) werden die folgenden Attribute der Anforderungen angezeigt und können bearbeitet werden, wenn der Benutzer die entsprechende Berechtigung hat:

ID	Wird automatisch zugewiesen.
Name	Der Name der Anforderung.
Priorität	Die Priorität der Anforderung. Mögliche Werte sind <i>Niedrig</i> , <i>Mittel</i> und <i>Hoch</i>
Status / Externer Status	Der Status der Anforderung. Wenn die Anforderung von einem externen System synchronisiert wird, wird hier der extern ver- waltete Status angezeigt.
	Andernfalls sind die möglichen Werte Gesperrt, Genehmigt, Entwurf und Auslassen.
Zusammenfassung	Die kurze Zusammenfassung der Anforderung.
Beschreibung	Die ausführliche Beschreibung der Anforderung.



Abbildung 6.22. Die Ansicht "Eigenschaften"

6.3.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.3.2.5. Anhänge

Einer Anforderung können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.3.2.6. Iterationen

Möchten Sie die Anforderung mit einer Iteration verknüpfen, klicken Sie auf den Button Zuordnen und wählen Sie ein oder mehrere Iterationen aus.

6.3.2.7. Testfälle

Der Abdeckungsgrad von Anforderungen wird durch Testfälle getestet. Testfälle können mit Anforderungen verknüpft werden in der Ansicht *Testfälle* verknüpft werden. Die Testergebnisse der verknüpften Testfälle zählen für den Abdeckungs- und Erfüllungsgrad der Anforderung.

≡ 🞐 K L A R	OS TEST MANAGEMENT	DE Finanz-Tracker 🖀 🗮	५ × 0 × ≛×
📝 Definieren	R00006 - Das Dashboard wird korrekt aktualisiert, nachdem etwas geändert wurde.		🖨 🛛 🌀 🗲
Planen 😫	Übersicht Eigenschaften Benutzerdefiniert Anhänge Iterationen (1) Testfälle (4) Revisioner	Speichern V n Ergebnisse Änderungen	/erwerfen Zurück
🏟 Ausführen	- 4 Einträge-Seite 1 von 1 🕅 ┥ 1 🕨 🕅 10 🗸	V	२ × & ≡
욙 Auswerten	Q ID+ Revision Name + - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -	Verfolgbarkeit \$ Requirement 1.2.4 in testplan_16.txt Requirement 1.2.4 in testplan_16.txt	Ausführung 🗢 Aktion Manuell 🍰
🖌 Konfigurieren	TC00014 1.0 Aktualisierung des Dashboards nach einer Banküberweisung. 24 TC00020 1.0 Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Requirement 1.2.4 in testplan_16.txt Requirement 1.2.4 in testplan_16.txt	Manuell &
	Erstellt vor 4 Jahren von Sabrina Gidley Zuweisen	Zuletzt geändert vo Speichern V	Vasioniung Auston or 4 Jahren von Felix Mustermann /erwerfen Zurück



Mit Klick auf den Button Zuweisen werden Testfälle einer Anforderung zugewiesen. Es öffnet sich ein Dialog, in dem mehrere Testfälle auf einmal ausgewählt werden können.

Für Anforderungen werden folgende Massenaktionen unterstützt:

– Entfernen

Die Detailseite eines Testfalls zeigt eine Liste aller Anforderungen, die von ihm abgedeckt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 6.7.2.7, "Revisionen"</u>.

6.3.2.8. Revisionen

Anforderungen können in unterschiedlichen Revisionen vorliegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.5, "Revisionen"</u>.

Contraction R00005 - 1.2 - Die Anwendung ist mit Android-Smartphones, Tablets und Smartwatches kompatibel Image: Contraction of the contractio	
Planen Übersicht Eigenschaften Benutzerdefiniert Anhänge Iterationen (9) Testfälle (5) Revisionen (3) Ergebnisse Änderungen Zurd	30
Ausführen Ausgewählte Revision 1.2	:k
Revisionskommentar Devices means smartphones, tablets and smartwatches.	
🔥 Auswerten Revision 💠 Erstellt 🗢 Erstellt 🗢 Erstellt 🗢 Geändert 🕈 Geändert von 🗢 Revisionskommentar 🗢	
O 1.2 vor 4 Jahren Talal Arif vor etwa einem Jahr Felix Mustermann Devices means smartphones, tablets and smartwatches.	
Konfinitrieren O 1.1 vor 4 Jahren Talal Arif vor etwa einem Jahr Felix Mustermann Devices means smartphones and tablets.	
O 1.0 vor 4 Jahren Talal Artf vor etwa einem Jahr Felix Mustermann Devices means smartphones.	
Erstellt vor 4 Jahren von Talal Arif	rmann
Neue Revision Speichern Verwerten Zuru	;K

Abbildung 6.24. Die Ansicht "Revisionen"

6.3.2.9. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Anforderung zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.3.2.10. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieser Anforderung an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

6.4. Testumgebungen

Eine *Testumgebung* fasst die äußeren Umstände zusammen, die ein Testergebnis beeinflussen können. Dies kann das Betriebssystem sein, die physikalische Umgebung, oder die Version eines Kundensystems, Webbrowsers oder Anwendungsservers. In diesem Kapitel wird gezeigt, wie Testumgebungen erstellt und bearbeitet werden.

6.4.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Testumgebungen in einer Tabelle an. Neue Testumgebungen werden hier erstellt.

= 👱 K L A R (OS TEST MANA			DE Finanz-Tracker 🖀 🗮	५ × 0 × ≗ ×
📝 Definieren	Testumgebungen				
• Dianon	Neu				Speichern Verwerfen
	₫ 🖨 û	5 Einträge - Seite 1 von 1	M ┥ 1 🕨 M 10 🗸	Alle anzeigen	Q × & ≡
📅 Ausführen	ID≑ 🖓				Aktion
Ausiunien	ENV00005 Android	8 Smartphone			ピ 🖨 🛍
A • • •	ENV00004 Android	9 Smartphone			ピ 🖨 🛍
C Auswerten	ENV00003 Android	Smartwatch			ぱ 🖨 🛍
	ENV00002 Android	Tablet			ぱ 🖨 🛍
差 Konfigurieren	ENV00001 Android	10 Smartphone			ぱ 🖨 🛍
	ID				Aktion
	Neu				Speichern Verwerfen

Abbildung 6.25. Die Übersichtsseite "Testumgebungen"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Beschreibung	Die Beschreibung der Testumgebung.
Aktionen	Die ausführbaren Aktionen.

Der Eintrag *Beschreibung* lässt sich mit einem Klick in das entsprechende Feld direkt in den Tabellenzeilen bearbeiten.

6.4.1.1. Eine neue Testumgebung erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun kann eine Beschreibung festgelegt werden.

Mit <u>Speichern</u> wird die neue Testumgebung erzeugt und gespeichert. Die ID der Testumgebung ENV00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *ENV00001* gelangen Sie auf die Detailseite der Testumgebung.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs haben Änderungen und sind noch nicht gespeichert!

6.4.1.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- 🔒 Druckansicht öffnen
- 🛍 Löschen

Wurde eine Testumgebung gelöscht, ist sie zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testumgebungen stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ⊘ Die Testumgebung unwiederbringlich aus der Datenbank entfernen

6.4.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testumgebungen gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testumgebungen aus, auf die die Aktion angewendet werden soll.

Für Testumgebungen werden folgende Massenaktionen unterstützt:

- 🕜 Bearbeiten
- 🔒 Druckansicht öffnen
- 🗊 Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Die Testumgebung unwiederbringlich aus der Datenbank entfernen (nur Administrator).
- 🗀 Einer Kategorie zuweisen (erscheint erst, nachdem eine Kategorie angelegt wurde).

Massenaktionen sind ausführlich in <u>Abschnitt 5.2.3.1.5, "Massenaktionen"</u> beschrieben.

6.4.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 옮 Kategorisieren
- √ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- Q Suchen
- 凸 Exportieren
- Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.4.2. Detailseite

Jede Testumgebung verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID der betreffenden Testumgebung oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Anhänge, Aufgaben, Ergebnisse und Änderungen.

≡ 🖞 K L A R	o s test management O	DE Finanz-Tracker 🖀 🗮 🔍 🔍 🕹 🗸
Definieren	ENV00001 - Android 10 Smartphone	음 🗅 😋 🔊
n Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Iterationen (9) Aufgaben	Speichern Verwerfen Zurück Ergebnisse (234) Änderungen
Ausführen	Erfolg 03% Fortschritt 75% Konformität 00%	
Konfigurieren	Testsystem Finance Tracker 1.1.0	
	Erfolgsverlauf (Letz Projekt DE Finanz Testsystem Finance Tracker 1.1.0, Testur	zte 30 Tage) z-Tracker ngebung Android 10 Smartphone
	90	Konformität Abdeckung
	80	
	70	•
	2020 2021	2022
	Testsysteme	
	ID Testsystem	12 12% Fortschritt Konformitat Abdeckung Aktion
	SUT00003 Finance Tracker 1.1.1	13 61% 75% 50% 71% O

Abbildung 6.26. Die Ansicht "Übersicht"

6.4.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben in der Kopfzeile weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der Testumgebung erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.
G S Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testumgebungen gewechselt werden.

6.4.2.2. Übersicht



Abbildung 6.27. Die Ansicht "Übersicht"

Alle hier angezeigten Daten beziehen sich immer auf die momentan angezeigte Testumgebung.

Auf der Ansicht Übersicht werden folgende Werte angezeigt:

Erfolg	Die Gesamterfolgsrate aller Testfälle. Eine Erfolgsrate von 100 % bedeutet, dass der letzte Testlauf für jeden Testfall erfolg- reich war. Auch wenn ein Testfall in der Vergangenheit bereits erfolgreich ausgeführt wurde, wird nur das letzte Ergebnis für die Erfolgsrate berücksichtigt.
Fortschritt	Die Fortschrittsrate zeigt an, wie viele Testfälle ausgeführt wurden. Im Gegensatz zur Erfolgsrate wird bei der Fortschritts- rate nur berücksichtigt, ob ein Testfall mindestens einmal aus- geführt wurde, unabhängig von dessen Ergebnis.
Konformität	Die Konformitätsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle erfolgreich ausgeführt wurden.
Abdeckung	Die Abdeckungsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle mindestens einmal ausgeführt wurden. Im Gegensatz zur Konformitätsrate berücksichtigt die Überde- ckungsrate nur, ob ein Testfall mindestens einmal in der Itera- tion ausgeführt wurde, unabhängig von dessen Ergebnis.
Automatisierungsgrad	Der Automatisierungsgrad zeigt an, wie viel Prozent der Test- fälle automatisiert durchgeführt werden können.
Diagramm Erfolgsverlauf	Dieses Diagramm zeigt die historische Entwicklung der Erfolgs-, Fortschritts-, Konformitäts- und Abdeckungsrate die- ser Testumgebung für das ausgewählte Testsystem an.

Die Zeitleiste wird automatisch an die vorhandenen Daten angepasst.

Im unteren Bereich befindet sich eine Tabelle mit den Kennzahlen dieser Testumgebung aufgeschlüsselt nach Testsystem.

6.4.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.28</u>) werden die folgenden Attribute der Testumgebung angezeigt und können bearbeitet werden:

Beschreibung

Eine Beschreibung der Testumgebung.

Abbildung 6.28. Die Ansicht "Eigenschaften"

6.4.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.4.2.5. Anhänge

Einer Testumgebung können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.4.2.6. Iterationen

Diese Seite zeigt alle Iterationen, denen diese Testumgebung zugeordnet ist.

= 👱 K L A R	o s test management	DE Finanz-Tracker 🗃 🗮 🔍 🔍 🗙 🖓 V 💄 V
📝 Definieren	ENV00005 - Android 8 Smartphone	음 🏿 😋 😜
📇 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Iterationen (5) Aufgaben	Speichern Verwerfen Zurück Ergebnisse (112) Änderungen
🏟 Ausführen	- 5 Einträge - Seite 1 von 1 🕅 ┥ 🚺 🕨 🕅	10 V Q X 选 三
 Auswerten Konfigurieren 	iD ◆ Q Name ◆ ITR00003 1.1.x-Sprint 01 - Übersichten ITR00005 1.1.x-Sprint 03-Annullierung von Einzugsermächtigungen ITR00005 1.0.x-Sprint 02-Enweiterung der Grundfunktionen ITR00002 1.0.x-Sprint 02-Enweiterung der Grundfunktionen ITR00001 1.1.x-Sprint 02-Daureartiräge bearbeiten ITR00001 1.0.x-Sprint 01-Grundfunktionen	Start € Failig € 10.03.2020 31.03.2020 30.04.2020 07.05.2020 07.02.2020 29.02.2020 31.03.2020 29.02.2020 27.01.2020 29.04.2020
	ID Name	Start Fällig
	Erstellt vor 4 Jahren von Sabrina Gidley	Zuletzt geändert vor 3 Jahren von Felix Mustermann
	Zuweisen	Speichern Verwerfen Zurück

Abbildung 6.29. Die Ansicht "Iterationen"

Mit Klick auf den Button Zuweisen kann diese Testumgebung einer oder mehreren Iterationen zugewiesen werden. Es öffnet sich ein Dialog, in dem mehrere Iterationen auf einmal ausgewählt werden können.

Für Iterationen werden folgende Massenaktionen unterstützt:

- Entfernen

Die Detailseite einer Iteration zeigt eine Liste aller Testumgebungen, die ihr zugeordnet sind. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 6.2.2.6</u>, <u>"Testumgebungen"</u>.

6.4.2.7. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieser Testumgebung an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

6.5. Testsysteme

Mit *Testsystem* wird das System/Objekt bezeichnet, welches getestet werden soll. Andere gebräuchliche Begriffe hierfür sind *Zu testendes System*, *Prüfsystem*, *System under Test*, *Testobjekt*. Dies kann beispielsweise durch die Versionsnummer der getesteten Software identifiziert werden. In diesem Kapitel wird gezeigt, wie Testsysteme erstellt und bearbeitet werden.

6.5.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Testsysteme in einer Tabelle an. Neue Testsysteme werden hier erstellt.

. ≡ 📌 K L A R •	OS TEST	MANAGEMEN	ΙT			DE Finanz-Tracker 🔄 📑	Q ×	0 ~ ≗ ~
📝 Definieren	Testsysteme							
t¶t Dianen	Neu						Speichern	Verwerfen
	ư đ ũ		6 Einträge - Seite 1 von 1	₩ ◄	1 🕨 🕅 10 🗸	옯 V Alle anzeigen	Q	× & ≡
🏚 Ausführen	ID\$ €	0			Version 🗢		Issues 🗢	Aktion
	SUT00006	Finance Tracker 2.1.0					1/2	C 🖨 🛈
	SUT00005	Finance Tracker 2.0.1					1/0	
Auswerten	SUT00004	Finance Tracker 2.0.0					0 / 0	₡₿₫
	SUT00003	Finance Tracker 1.1.1					1/0	☞ 🖨 🛍
🖌 Konfigurieren	SUT00002	Finance Tracker 1.1.0					0 / 0	☞ 🖨 🛍
	SUT00001	Finance Tracker 1.0.0					0 / 0	☞ 🖨 🛍
	ID							Aktion
	Neu						Speichern	Verwerfen

Abbildung 6.30. Die Übersichtsseite "Testsysteme"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Beschreibung	Die Beschreibung des Testsystems.
Aktionen	Die ausführbaren Aktionen.

Der Eintrag *Beschreibung* lässt sich mit einem Klick in das entsprechende Feld direkt in den Tabellenzeilen bearbeiten.

6.5.1.1. Ein neues Testsystem erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun kann die *Version* des Testsystems festgelegt werden.

Mit Speichern wird das neue Testsystem erzeugt und gespeichert. Die ID des Testsystems (SUT00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *SUT00001* gelangen Sie auf die Detailseite des Testsystems.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs enthalten Änderungen und sind noch nicht gespeichert!

6.5.1.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🕜 Bearbeiten
- 🔒 Druckansicht öffnen

前 Löschen

Wurde ein Testsystem gelöscht, ist es zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testsysteme stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ☆ Das Testsystem unwiederbringlich aus der Datenbank entfernen

6.5.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testsysteme gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testsysteme aus, auf die die Aktion angewendet werden soll.

Für Testsysteme werden folgende Massenaktionen unterstützt:

- 🕜 Bearbeiten
- Druckansicht öffnen
- 🗊 Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Das Testsystem unwiederbringlich aus der Datenbank entfernen (nur Administrator).
- Einer Kategorie zuweisen (erscheint erst, nachdem eine Kategorie angelegt wurde).

Massenaktionen sind ausführlich in <u>Abschnitt 5.2.3.1.5, "Massenaktionen"</u> beschrieben.

6.5.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 品 Kategorisieren
- √ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.5.2. Detailseite

Jedes Testsystem verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID des betreffenden Testsystems oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Anhänge, Issues, Aufgaben, Ergebnisse und Änderungen.

≡ 📌 K L A R	o s test management	DE Finanz-Tracker 🖬 🗮 🔍 🔍 🗙 V
📝 Definieren	SUT00002 - Finance Tracker 1.1.0	🖨 🛛 😋 🔊
🔹 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Issues Iterationen (1)	Speichern Verwerfen Zurück Aufgaben Ergebnisse (50) Änderungen
💠 Ausführen	Erfolg 90%	
🔥 Auswerten	Fortschift 45% Konformität 83% Abdeckung 42%	
🖌 Konfigurieren	Testumgebung Android 10 Smartphone	
	Erfolgsverlauf (L Projekt DE Fin Testsystem Finance Tracker 1.1.0, Tes	etzte 30 Tage) anz-Tracker tumgebung Android 9 Smartphone
	100 Erfolg	Konformität Abdeckung
	90 -	
	80	
	70	
	50	
	40 -	
	30 -	
	10	
	0	
	2020 2021	2022 2023
	Testumgebungen	
	ID Testumgebung	Testläufe Erfolg Fortschritt Konformität Abdeckung Aktion
	ENV0002 Android Smartwatch	0 0% 0% 0% 0% 0
	ENV00001 Android 10 Smartphone	13 13% 62% 16% 42% O
	ENV00005 Android 8 Smartphone	5 75% 33% 100% 21% O
	ENVUUUU4 Anarola 9 Smartphone	/ 90% 45% 83% 42% O
	Alle anzeigen	
	Zuletzt ausgeführte Testfälle	
	5 Einträge - Seite 1 von 1 🕅 ┥ 1 🕨 🕅	10 ▼ Q × & Ξ

Abbildung 6.31. Die Ansicht "Übersicht"

6.5.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben in der Kopfzeile weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Testsystems erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon ☐ wird dieser Link in die Zwischenablage kopiert.
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.
Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testsystemen gewech- selt werden.

6.5.2.2. Übersicht

— . 		
E Y KLAR	OSIESIMANAGEMENI DEFinanz-tracker ■ ቛ Q× "Ø♥ ≚	*
📝 Definieren	SUT00002 - Finance Tracker 1.1.0	Ð
••• Planan	Speichern Verwerfen Zurück	
	Übersicht Eigenschaften Benutzerdefiniert Anhänge Issues Iterationen (1) Aufgaben Ergebnisse (50) Änderungen	
🌣 Ausführen	Erfolg 90%	
	Fortschritt 45%	
🕒 Auswerten	Abdeckung 42%	
🔎 Konfigurieren	Testumgebung Android 10 Smartphone	
	Erfolgsverlauf (Letzte 30 Tage) Projekt DF Finanz-Tracker	
	Testsystem Finance Tracker 1.1.0, Testumgebung Android 9 Smartphone	
	Erfolg Fortschritt Abdeckung	
		,
	70	
	50 50	
)
	10	
	2020 2021 2022 20	23
	Testungebungen	
	ID Testumgebung Testläufe Erfolg Fortschritt Konformität Abdeckung Akt	ion
	ENV00003 Android Smartwatch 0 0% 0% 0% 0%)
	ENV00002 Android Tablet 0 0% 0% 0% 0% 0)
	ENV00001 Android 10 Smartphone 13 13 62% 16% 42% 0)
	ENV00005 Android 8 Smartphone 5 75% 83% 100% 21% C	2
	Envouou4 Anorona 9 sinartynone / 90% 45% 83% 42% C	/
	Alle anzeigen	
	Zuletzt ausneführte Testfälle	
	5 Einträge-Seite 1 von 1 🕅 🚽 🚺 🕨 🛛 10 🗸	≡

Abbildung 6.32. Die Ansicht "Übersicht"

Alle hier angezeigten Daten beziehen sich immer auf das momentan angezeigte Testsystem.

Auf der Ansicht Übersicht werden folgende Werte angezeigt:

Erfolg	Die Gesamterfolgsrate aller Testfälle. Eine Erfolgsrate von 100 % bedeutet, dass der letzte Testlauf für jeden Testfall erfolg- reich war. Auch wenn ein Testfall in der Vergangenheit bereits erfolgreich ausgeführt wurde, wird nur das letzte Ergebnis für die Erfolgsrate berücksichtigt.
Fortschritt	Die Fortschrittsrate zeigt an, wie viele Testfälle ausgeführt wurden. Im Gegensatz zur Erfolgsrate wird bei der Fortschritts- rate nur berücksichtigt, ob ein Testfall mindestens einmal aus- geführt wurde, unabhängig von dessen Ergebnis.
Konformität	Die Konformitätsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle erfolgreich ausgeführt wurden.
Abdeckung	Die Abdeckungsrate zeigt an, wie viele Anforderungen zuge- ordnete Testfälle mindestens einmal ausgeführt wurden. Im Gegensatz zur Konformitätsrate berücksichtigt die Überde- ckungsrate nur, ob ein Testfall mindestens einmal in der Itera- tion ausgeführt wurde, unabhängig von dessen Ergebnis.

Automatisierungsgrad	Der Automatisierungsgrad zeigt an, wie viel Prozent der Test- fälle automatisiert durchgeführt werden können.
Diagramm Erfolgsverlauf	Dieses Diagramm zeigt die historische Entwicklung der Erfolgs-, Fortschritts-, Konformitäts- und Abdeckungsrate die- ses Testsystems für die ausgewählte Testumgebung an.
	Die Zeitleiste wird automatisch an die vorhandenen Daten angepasst.

Im unteren Bereich befindet sich eine Tabelle mit der Übersicht der Ergebnisse für dieses Testsystem aufgeschlüsselt nach Testumgebung. In der Spalte *Aktion* kann die Testumgebung für das obige Diagramm ausgewählt werden.

6.5.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.33</u>) werden die folgenden Attribute des Testsystems angezeigt und können bearbeitet werden:

Version

Die Version des Testsystems.

\equiv 🞐 KLAR	o s test management	DE Finanz-Track
📝 Definieren	SUT00002 - Finance Tracker 1.1.0	
📇 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Issues Iterationen (1) Aufgaben Ergebnisse (50) Änderungen	
🏟 Ausführen	ID SUT00002 Version Finance Tracker 1.1.0	
🕒 Auswerten	Erstellt vor 4 Jahren von Talal Arif	
✗ Konfigurieren		

Abbildung 6.33. Die Ansicht "Eigenschaften"

6.5.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.5.2.5. Anhänge

Einem Testsystem können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.5.2.6. Issues

In dieser Ansicht werden alle Issues aufgelistet, die mit diesem Testsystem verknüpft sind.



Abbildung 6.34. Die Ansicht "Issues"

Die Tabelle zeigt die folgenden Werte:

ID	Die ID des Issues im externen Issue-Management-System.
System	Das externe Issue-Management-System dieses Issues.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Zusammenfassung	Die Zusammenfassung des Issues.
Erstellt	Das Datum, an dem dieser Issue erstellt wurde.
Ersteller	Der Benutzer, der diesen Issue erstellt hat.
Zugewiesen	Der Benutzer, dem dieser Issue zugewiesen ist.
Priorität	Die Priorität des Issues. Die möglichen Werte werden durch das externe Issue-Management-System vorgegeben.
Status	Der Status des Issues. Die möglichen Werte werden durch das externe Issue-Management-System vorgegeben.
Aktionen	Die ausführbaren Aktionen.

6.5.2.6.1. Ein neues Issue erstellen

Mit Klick auf <u>Neu</u> öffnet sich die Seite *Issue Details* (<u>Abschnitt 9.6.6, "Issue - Details</u>"), auf der neue Issues erstellt und mit dem Testsystem verknüpft werden können. Mit Klick auf <u>Link</u> werden bestehende Issues mit dem Testsystem verknüpft (<u>Abschnitt 9.6.7, "Einen Issue verlinken</u>").

6.5.2.6.2. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🕜 Bearbeiten
- 🖸 Öffnen
- fi Löschen

6.5.2.7. Iterationen

Diese Seite listet alle Iterationen, auf denen dieses Testsystem zugeordnet ist.

≡ 👱 K L A R C) S TEST MANAGEMENT DE	Finanz-Tracker 🖬 🗮 🚺 🔍 🕹 🗸
📝 Definieren	SUT00003 - Finance Tracker 1.1.1	🖶 🗋 😋 🔊
📇 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Issues (1) <u>Iterationen (3)</u> Aufgaben B	Speichern Verwerfen Zurück Ergebnisse (113) Änderungen
🔹 Ausführen	- 3 Einträge - Seite 1 von 1 🕅 ┥ 1 🕨 🕅 🔽	\[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\] \[\]
e Auswerten	ID \$\Phi\$ Q Name \$ ITR00006 1.1.x-Sprint 04-Optimierung für Tablets 1.1.x-Sprint 03-Annullierung von Einzugsermächtigungen	Start ◆ Fällig ◆ 06.05.2020 21.05.2020 30.04.2020 07.05.2020
🔑 Konfigurieren	ITR00004 1.1.x-Sprint 02-Daueraufträge bearbeiten ID Name	31.03.2020 29.04.2020 Start Fällig
	Erstellt vor 4 Jahren von Talal Arif Zuweisen	Zuletzt geändert vor 4 Jahren von Felix Mustermann Speichern Verwerfen Zurück



Mit Klick auf den Button Zuweisen kann dieses Testsystem einer oder mehreren Iterationen zugewiesen werden. Es öffnet sich ein Dialog, in dem mehrere Iterationen auf einmal ausgewählt werden können.

Für Iterationen werden folgende Massenaktionen unterstützt:

- Entfernen

Die Detailseite einer Iteration zeigt eine Liste aller Testsysteme, die ihr zugeordnet sind. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 6.2.2.7, "Testsysteme"</u>.

6.5.2.8. Aufgaben

Die Seite zeigt die Liste der dem ausgewählten Testsystem zugeordneten Aufgaben an.

In der Tabelle werden die folgenden Werte angezeigt:

ID	Wird automatisch zugewiesen.
Zusammenfassung	Die Zusammenfassung der Aufgabe
Priorität	Die Priorität der Aufgabe. Mögliche Werte sind Trivial, Niedrig, Hoch, Kritisch, Blocker
Status	Der Status der Aufgabe. Mögliche Werte sind Neu, Wieder eröffnet, In Arbeit, Erledigt, Geschlossen, Abgelehnt.
Fortschritt	Der Prozentsatz der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.
Erfolg	Die Erfolgsrate der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.
Fällig	Das Datum, bis zu dem diese Aufgabe beendet sein soll.
Zugewiesen	Der Benutzer, dem die Aufgabe zugewiesen ist.
Aktion	Zur Verfügung stehen die Aktionen 😰 Bearbeiten, 🖨 Druck- ansicht öffnen und Ausführen. Die für Ausführen möglichen Darstellungen finden Sie unter <u>Abschnitt F.2.2, "Ausführungs- aktionen"</u> .

6.5.2.9. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die diesem Testsystem zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.5.2.10. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Testsystems an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> <u>"Änderungshistorie"</u>.

6.6. Testsegmente



Ein *Testsegment* definiert und kapselt eine Sequenz von Testschritten. In Testfällen können diese Testsegmente als Testschritte eingeblendet werden. Der Inhalt eines Testsegments wird hier zentral bearbeitet und die Änderungen automatisch in den Testfällen übernommen. In diesem Kapitel wird gezeigt, wie Testsegmente erstellt und bearbeitet werden.

≡ 🧏	KLAR	OS TEST							DE Finanz-Tracker			Q	× 0 × 🛓	
📝 Def	finieren	Testsegmente												
📇 Pla	anen	Neu										Speichern	Verwerfen	
		rovat	Û		1 Einträge - Seite 1 von 1	М ┥	1 🕨	10 🗸		#L	Alle anzeigen		Q × 🖧 :	≡
📩 📩	sführen	🔲 id 🗘 🖸	Prevision				Name 🖨				Schritte	🕈 Testfälle 🗘	Aktion	
	ordinen	SEG00005	1.0 Lo	gin							1	17	C C A t	ÌÌ
		ID	Revision				Name				Schritte	Testfälle	Aktion	
C Au	Iswerten	Neu										Speichern	Verwerfen	
🔑 Kor	nfigurieren													

Abbildung 6.36. Die Seite "Testsegmente"

6.6.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Testsegmente in einer Tabelle an. Neue Testsegmente werden hier erstellt.

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision des Testsegments.
Name	Der Name des Testsegments.
Schritte	Die Anzahl der definierten Testschritte.

Testfälle

Die Anzahl der Testfälle in denen dieses Testsegment vorkommt.

Der Eintrag Name lässt sich mit einem Klick in das entsprechende Feld direkt in den Tabellenzeilen bearbeiten.

6.6.1.1. Ein neues Testsegment erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun kann der Name festgelegt werden.

Mit Speichern wird das neue Testsegment erzeugt und gespeichert. Die ID des Testsegments (SEG00001) wird dabei automatisch vergeben. Mit Klick auf die ID *SEG00001* gelangen Sie auf die Detailseite des Testsegments.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs haben Änderungen und sind noch nicht gespeichert!

6.6.1.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🖉 Bearbeiten
- Duplizieren
- 🔒 Druckansicht öffnen
- fi Löschen

Wurde ein Testsegment gelöscht, ist es zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testsegmente stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ☆ Das Testsegment unwiederbringlich aus der Datenbank entfernen

6.6.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testsegmente gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testsegmente aus, auf die die Aktion angewendet werden soll.

Für Testsegmente werden folgende Massenaktionen unterstützt:

- 🕜 Bearbeiten
- Duplizieren

D Neue Revision erzeugen

🔒 Druckansicht öffnen

- 🗊 Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Das Testsegment unwiederbringlich aus der Datenbank entfernen (nur Administrator).

Massenaktionen sind ausführlich in Abschnitt 5.2.3.1.5, "Massenaktionen" beschrieben.

6.6.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- Q Suchen
- 凸 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.6.2. Detailseite

Jedes Testsegment verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID des betreffenden Testsegments oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht *Eigenschaften*.

Es stehen die folgenden Ansichten zur Verfügung: Eigenschaften, Benutzerdefiniert, Schritte, Anhänge, Revisionen, Testfälle und Änderungen.

6.6.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben in der Kopfzeile weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

금 Druckansicht erstellen	Hiermit kann eine druckfreundliche Ansicht aller Einträge des betreffenden Testsegments erstellt werden. Mit Klick auf das Icon 🛱 öffnet sich diese in einem neuen Browsertab.				
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.				
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.				
	Das Anlegen von Lesezeichen ist ausführlich in Abschnitt 5.2.3.2.2, "Lesezeichen" beschrieben.				



Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testsegmenten gewechselt werden.

6.6.2.2. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.37</u>) können die folgenden Attribute des Testsegments angezeigt und können bearbeitet werden:

= 坐 K L A R (o s test mai	NAGEMENT	DE Finanz-Tracker 🖀 🗮 🔍 🔍 🕹 🗸
📝 Definieren	SEG00005 - Login		🖨 🛛 🔇 🔊
🏖 Planen	Eigenschaften Ben	utzerdefiniert Schritte (1) Anhänge Revisionen Änderungen	Speichern Verwerfen Zurück
🌣 Ausführen	ID	SEG00005	
🕒 Auswerten	Name Beschreibung	Login Diese Schritte beschreiben, wie Sie sich bei der App anmelden.	
🖌 Konfigurieren			
	Vorbedingung	Stellen Sie sicher, dass die Anwendung vor dem Testen installiert ist • Die Anwendung ist auf dem Gerät installiert. Falls nicht, folgen Sie den Anweisung • Verbindung zum Server besteht.	per: https://finanztracker/install
	Erwartetes Ergebnis	Der Benutzer kann sich in die App einloggen.	
	Nachbedingung	Sie gelangen auf das Startbild.	
	Anmerkung	Stellen Sie sicher, dass eine Internetverbindung zur Verfügung steht.	
	Erstellt vor 3 Jahren von Feli	x Mustermann	Zuletzt geändert vor 3 Jahren von Felix Mustermann

Abbildung 6.37. Die Ansicht "Eigenschaften"

ID	Wird automatisch zugewiesen.
Name	Der Name des Testsegments.
Beschreibung	Die textuelle Beschreibung des Testsegments.
Vorbedingung	Die Vorbedingung des Testsegments.
Erwartetes Ergebnis	Die erwartetes Ergebnis des Testsegments.
Nachbedingung	Die Nachbedingung des Testsegments.
Anmerkung	Eine optionale Anmerkung.



Editor-Einstellungen

Unter Definieren --> Projekt --> Eigenschaften kann eingestellt werden, ob der Editor nur Text oder HTML darstellt. Bei der Einstellung HTML erscheint beim Klick in das Textfeld eine Leiste mit Formatierungsoptionen. Mit Klick auf den Button Speichern werden die ausgewählten Änderungen in der Datenbank gespeichert, Verwerfen verwirft die Änderungen des Benutzers.

6.6.2.3. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.6.2.4. Testschritte

Die Testschritte enthalten die folgenden Angaben:

Aktionen	Die Beschreibung der auszuführenden Aktion, z.B. Namen und Passwort eingeben oder Klicken Sie auf den Login-Button
Erwartetes Ergebnis	Das (sichtbare) Verhalten des Systems, das bei der Durchfüh- rung des Testschritts vorhergesagt wird, z. B. Der Benutzer erhält eine Meldung, dass die Anmeldung erfolgreich war.
Vorbedingung	Die Bedingung, die vor der Ausführung des Testschritts erfüllt sein muss, z.B. Der Benutzer ist noch nicht eingeloggt.
Nachbedingung	Die Umgebungs- und Zustandsbedingungen, die nach der Aus- führung des Testfalls erfüllt sein müssen, z.B. Der Benutzer ist authentifiziert und hat Zugriff auf das System

≡ 📌 K L A R	OS TEST MANAGEMENT DEFmanz-Trac	ker 🖬 🗮 🔍 🔍 🔍 🕹 ∨
📝 Definieren	SEG00005 - Login	음 🏾 😌 🗲
🏝 Planen	Elgenschaften Benutzerdefiniert Schritte (1) Anhänge Revisionen Änderungen	Speichern Verwerfen Zurück
🌣 Ausführen	Neu	
🕒 Auswerten	1 Einträge - Seite 1 von 1 🖌 ┥ 🚺 🕨 🕅 🔟 🗸	田 井 ()
🖌 Konfiguriaran	Schritt 1: Starten Sie die Anwendung durch Drücken der blauen Symbolschaltfläche .	: ⊲ ⊳ 🗋 🗕
Koniiguneren	Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung Benutzerdefiniert Anhänge	
	Starten Sie die Anwendung durch Drücken der blauen Symbolschaltfläche	B
	Neu	
	Erstellt vor 4 Jahren von Felix Mustermann	
		Zuletzt geändert vor 4 Jahren von Felix Mustermann
		Speichern Verwerfen Zurück

Abbildung 6.38. Die Ansicht "Schritte"

6.6.2.5. Anhänge

Einem Testsegment können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.6.2.6. Revisionen

Testsegmente können in unterschiedlichen Revisionen vorliegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.5, "Revisionen"</u>.

6.6.2.7. Testfälle

Die Seite zeigt eine Liste der zum ausgewählten Testsegment zugeordneten Testfälle an.

In der Tabelle werden die folgenden Werte angezeigt:

ID	Wird automatisch zugewiesen.
Revision	Die Revision des Testsegments.
Name	Der Name des Testsegments.
Verfolgbarkeit	Ein Verweis auf die entsprechende Anforderung, den Anwen- dungsfall oder das Arbeitspaket.
Status	Der Status des Testfalls. Mögliche Werte sind Gesperrt, Geneh- migt, Entwurf und Auslassen.
Schritte	Die Anzahl der Testschritte.
Weitere Informationen zu Testfälle	en finden Sie unter A <u>bschnitt 6.7, "Testfälle"</u> .

6.6.2.8. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Testsegments an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> <u>"Änderungshistorie"</u>.

6.7. Testfälle

Ein Testfall ist ein Satz von Eingabewerten, Ausführungsvoraussetzungen, erwarteten Ergebnissen und Ausführung-Nachbedingungen, die für ein bestimmtes Ziel oder eine Testbedingung entwickelt wurden, um zu bestimmen, ob eine Anwendung oder ein Softwaresystem seine Spezifikationen erfüllt. In diesem Kapitel wird gezeigt, wie Testfälle erstellt und bearbeitet werden.

6.7.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Testfälle in einer Tabelle an. Neue Testfälle werden hier erstellt.

= 👱 KLAR	OS TEST	МА	N A G	EMENT		DE Finanz-Tracke	er 🖬 📑			Q ;	< @~ _ ~
📝 Definieren	Testfälle										
📇 Planen	Neu		-			3	-		Speic	hern	Verwerfen
	+ 6 6 %		6	24 Einträge - Seite 1 von 3		1		Alle anzeigen			<u> </u>
🏚 Ausführen	L ID≑ ♡	÷ I	Revision	Name 🗢	Verfolgbarkeit 🗢	Priorität 🗢 S	tatus 🗢 🛛	Ausführung 🗢 🛛 Sch	iritte 🗢 🛛 I:	ssues ≑	Aktion
	TC00024	A	1.0	Überprüfen Sie die Verbindung zum Server	Requirement 1.2.4 in testplan_16.txt	Hoch E	ntwurf	Automatisiert	0	2/1	0040
🕒 Auswerten	TC00023	A	1.0	Einen Dauerauftrag ändern.	Requirement 1.2.4 in testplan_16.txt	Mittel Er	ntwurf I	Manuell	5	0/2	₡ @ ₴ ₪
	TC00022	*	1.0	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	Mittel Er	ntwurf	Manuell	6	5/0	┏┖₽₫
🔎 Konfigurieren	TC00021	0	1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	Mittel Er	ntwurf	Manuell	6	0/2	ぴ₽₽₫
	TC00020	*	1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Requirement 1.2.4 in testplan_16.txt	Mittel Ei	ntwurf	Manuell	6	0/0	ぴ₽₿₫
	TC00019	0	1.0	Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/ testing/web/Connection.java	Hoch E	ntwurf	Automatisiert	0	0/0	ぴ₽₽₫
	TC00018	0	1.0	Testen der Verbindung zum Server der Bank.	/FinanceTracker/src/test/java/de/sabrina/ testing/web/Connection.java	Hoch E	ntwurf	Automatisiert	0	0/0	ぴ₽₿₫
	TC00017	A	1.0	Lastschriftmandat löschen.	Requirement 1.2.4 in testplan_16.txt	Mittel E	ntwurf	Manuell	5	0/0	┏┍₽ΰ
	TC00016	0	1.0	Daueraufträge löschen.	Requirement 1.2.4 in testplan_16.txt	Mittel E	ntwurf I	Manuell	5	0/0	CCAÓ
	TC00015	0	1.0	Fügen Sie einen Dauerauftrag hinzu.	Requirement 1.2.4 in testplan_16.txt	Mittel E	ntwurf I	Manuell	5	0/0	┏┏₽₫
	ID										Aktion
	Neu								Speic	hern	Verwerfen

Abbildung 6.39. Die Übersichtsseite "Testfälle"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision des Testfalles.
Name	Der Name des Testfalls.
Verfolgbarkeit	Die Anforderung, die diesem Testfall zugeordnet ist.
Priorität	Die Priorität des Testfalls. Mögliche Werte sind Niedrig, Mittel und Hoch.
Status	Der Status des Testfalls bestimmt, ob er bearbeitbar oder aus- führbar ist. Nur Testfälle mit dem Status "Entwurf" oder "Über- springen" sind bearbeitbar. Nur Testfälle mit dem Status "Ent- wurf" oder "Genehmigt" sind ausführbar.

	Status: Testfall									
	Entwurf Genehmigt Auslassen Gesperrt									
Editierbar	\bigcirc	8	0	8						
Ausführbar	\bigcirc	\bigcirc	8	8						

	Ausf	<u> </u>				
Ausführung	Manuell oder Automatisiert.					
Schritte	Die Anz	zahl der de	finierten Tests	chritte für die	sen Testfall.	

Issues

Die Anzahl der offenen und behobenen Issues.

Aktion Die ausführbaren Aktionen.

Die Einträge Name, Verfolgbarkeit, Priorität, Status und Ausführung lassen sich mit einem Klick in die entsprechenden Felder direkt in den Tabellenzeilen bearbeiten.

6.7.1.1. Einen neuen Testfall erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun können Name, Verfolgbarkeit und Priorität und die Ausführung festgelegt werden.

Mit Speichern wird der neue Testfall erzeugt und gespeichert. Die ID des Testfalls (TC00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *TC00001* gelangen Sie auf die Detailseite des Testfalls.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst mit Klick auf Speichern werden die Daten in die Datenbank übernommen.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Ungespeicherte Änderungen

In allen mit roten IDs versehenden Zeilen befinden sich Änderungen, die noch nicht gespeichert sind.

6.7.1.2. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- Duplizieren
- 🔒 Druckansicht öffnen
- 前 Löschen

Wurde ein Testfall gelöscht, ist er zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testfälle steht dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ⊘ Den Testfall unwiederbringlich aus der Datenbank entfernen

6.7.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testfälle gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testfälle aus, auf die die Aktion angewendet werden soll.
Für Testfälle werden folgende Massenaktionen unterstützt:

- + Einer neuen Testsuite zuweisen
- 🛯 Bearbeiten
- Duplizieren
- D Neue Revision erzeugen
- 🔒 Druckansicht öffnen
- 前 Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Den Testfall unwiederbringlich aus der Datenbank entfernen (nur Administrator).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

6.7.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

♣ Kategorisieren
 ♥ Filter / Sortieren
 Zeige alle / Nur aktive (Nur Administrator)
 Q Suchen
 ♣ Exportieren
 = Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

6.7.2. Detailseite

Jeder Testfall verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID des betreffenden Testfalls oder auf das Icon
rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Schritte, Anhänge, Revisionen, Issues, Aufgaben, Ergebnisse und Änderungen.



Abbildung 6.40. Die Ansicht "Übersicht"

6.7.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben in der Kopfzeile weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

Review	Legt eine Review-Aufgabe an.
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Testfalls erstellt werden. Mit Klick auf das Icon 🔒 öffnet sich diese in einem neuen Browsertab.
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon 🔲 wird dieser Link in die Zwischenablage kopiert.
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.
G ● Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testfällen gewechselt werden.

6.7.2.2. Übersicht

Auf der Ansicht Übersicht werden folgende Werte angezeigt:

Status	Der Status des Testfalls. Mögliche Werte sind Gesperrt, Geneh- migt, Entwurf und Auslassen.
Ausführung	Die Art der Testfallausführung: Manuell oder Automatisiert.
Priorität	Die Priorität des Testfalls: Niedrig, Mittel oder Hoch.
Letzter Ausführender	Das Benutzer, der diesen Testfall zuletzt ausgeführt hat.
Die Ausführungszeit dieses Testfalls.	Dieses Diagramm zeigt die minimale, maximale und durch- schnittliche Ausführungszeit für diesen Testfall. Dadurch kann der Zeitbedarf für zukünftige Ausführungen dieses Testfalls besser geschätzt werden.
Die erfolglosesten Testschritte	Dieses Diagramm zeigt alle Testschritte dieses Testfalls und welche Schritte während eines Testlaufs als fehlgeschlagen oder als Fehler markiert wurden. Das ist hilfreich bei der Ermittlung möglicher Fehlkonfigurationen, z. B. wenn an einem bestimmten Schritt häufiger Fehler auftreten.
Testsystem - Überblick	Diese Tabelle zeigt das letzte Ergebnis dieses Testfalls für jedes Testsystem und jede Testumgebung, in der dieser Test- fall ausgeführt wurde. Mit Klick auf das Ergebnis-Icon wird das zugehörige Testfallergebnis geöffnet.
Letzte Ergebnisse	Diese Tabelle zeigt das letzte Ergebnis dieses Testfalls. Mit Klick auf das Icon Q wird das zugehörige Testfallergebnis geöffnet.
Letzte Ergebnisse	Diese Liste zeigt die neuesten Issues, die mit diesem Testfall verknüpft sind. Mit Klick auf das Icon 🖉 wird das Issue direkt im Issue-Management-System angezeigt.

6.7.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.41</u>) werden die folgenden Attribute des Testfalls angezeigt und können bearbeitet werden:



Editor-Einstellungen

Unter Definieren --> Projekt --> Eigenschaften kann eingestellt werden, ob der Editor nur Text oder HTML darstellt. Bei der Einstellung HTML erscheint beim Klick in das Textfeld eine Leiste mit Formatierungsoptionen.

ID	Wird automatisch zugewiesen.
Name	Der Name des Testfalles.
Beschreibung	Die ausführliche Beschreibung des Testfalls.
Vorbedingung	Bedingung, die vor der Ausführung des Testschritts erfüllt sein muss, z. B. Es ist erforderlich, dass die Datenbank initialisiert wird.

Erwartetes Ergebnis	Das beobachtbare Verhalten des Systems, das bei der Durch- führung des Testschritts vorhergesagt wird, z. B. Der Benutzer erhält eine Meldung, dass die Anmeldung erfolgreich war.
Nachbedingung	Die Umgebungs- und Zustandsbedingungen, die nach der Aus- führung des Testfalls erfüllt sein müssen, z. B. Der Benutzer ist authentifiziert und hat Zugriff auf das System.
Geschätzte Dauer	Die geschätzte Dauer, die für die Durchführung dieses Test- falls benötigt wird.
Note	Eine optionale Bemerkung, beispielsweise Hinweise an den Tester.
Testart	Eine Gruppe von Testaktivitäten basierend auf spezifischen Testzielen mit dem Zweck, eine Komponente oder ein Sys- tem auf bestimmte Eigenschaften zu testen (Funktional, Nicht- funktional, Strukturell, Regression, Retest).
Ergebnisart	Der erwartete Ausgang des Tests: Positiv oder Negativ.
Priorität	Die Priorität des Testfalls: Niedrig, Mittel oder Hoch.
Team	Das Team, das für den Testfall verantwortlich ist.
Dokument	Ein Verweis auf das Dokument, auf dem dieser Testfall basiert, das z. B. die Anforderungen enthalten kann, auf die sich dieser Testfall bezieht.
Evaluierungsart	Die Art der Testergebnisauswertung: Manuell oder Automati- siert.

≡ 📌 KLAR	S TEST MANAGEMENT DE Finanz-Tracker 🖬 🗮 🔍 🔍 🖌	• •
📝 Definieren	TC00022 - Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags. 💿 🖨 🗍 🔇	Ð
🏝 Planen	Übersicht Eigenschaften Benutzerdefiniert Schritte (6) Anhänge Revisionen Issues (5) Aufgaben (1) Ergebnisse (25) Änderungen	
🏟 Ausführen	ID TC00022 Name Aktualizioren Sie des Dechbeerd durch Hiszuffigen eines Deurzuffrage	0
\rm e Auswerten	Beschreibung Testet die automatische Aktualisierung des Dashboards durch <i>Drücken</i> der Taste	0
🗲 Konfigurieren	Dauerauftrag hinzufügen zum Hinzufügen eines neuen Dauerauftrags.	
	Der Dauerauftrag, der nicht bearbeitet werden soll, sollte folgende Informationen enthalten:	
	Begünstigter: Max Mustermann IBAN: DE19 1234 1234 1234 1234 1234 12 Betrag: € 100,- Rechnung: 38745 Zeitintervall: monatlich	
	Vorbedingung Stellen Sie sicher, dass die Anwendung vor dem Testen installiert ist! Die Anwendung ist auf dem Gerät installiert. Falls nicht, folgen Sie den Anweisungen: https://finanztracker/install Verbindung zum Server besteht.	0
	Erwartetes Ergebnis Ein neuer Dauerauftrag ist auf dem Dashboard aufgeführt.	0
	Nachbedingung Das Dashboard wurde aktualisiert.	0



6.7.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.7.2.5. Testschritte

In der Ansicht *Schritte* werden die Testschritte erstellt. Testschritte bilden die Ablaufbeschreibung eines Testfalls. Beim ersten Öffnen der Ansicht sind folgende Elemente zu sehen:

Neu Erstellt einen neuen Testschritt.

Testsegment einfügen Fügt ein bestehendes Testsegment ein.

Ansichtsmodus Wechseln Wechselt die Ansicht zwischen einer Anzeige mit und ohne Tabs.

- לבי Minimiert die Darstellung aller Schritte.
- [] Maximiert die Darstellung aller Schritte.

6.7.2.5.1. Einen neuen Testschritt erstellen

Mit Klick auf den Button Neu öffnet sich das Panel zum Erstellen eines neuen Testschrittes.

‡ Verschiebt den Eintrag an eine andere Position.

- ▶ Fügt einen neuen Schritt davor ein.
- KX Fügt ein neues Testsegment davor ein.
- ► Fügt ein neues Testsegment danach ein.
- Fügt einen neuen Schritt danach ein.
- Dupliziert den Schritt.
- Discht den Schritt.
- und + Öffnet und schließt die Feldansicht.

Sobald das Eingabefeld aktiviert wird, erscheint im Textfeld eine Symbolleiste. Die Texte können hiermit formatiert sowie mit Links und Bildern versehen werden.

Aktionen	Die Beschreibung der auszuführenden Aktion, z. B. Namen und Passwort eingeben. oder Klicken Sie auf den Login-Button.
Erwartetes Ergebnis	Das (sichtbare) Verhalten des Systems, das bei der Durchfüh- rung des Testschritts vorhergesagt wird, z. B. Der Benutzer erhält eine Meldung, dass die Anmeldung erfolgreich war.
Vorbedingung	Bedingung, die vor der Ausführung des Testschritts erfüllt sein muss, z. B. Der Benutzer ist noch nicht eingeloggt.
Nachbedingung	Die Umgebungs- und Zustandsbedingungen, die nach der Aus- führung des Testfalls erfüllt sein müssen, z. B. Der Benutzer ist authentifiziert und hat Zugriff auf das System.

= 👱 K L A R	OS TEST MANAGEMENT DE Finanz-Tracker 🖬 🗮 🔍 🔍 🗙 🔞	⊳~ ≞ ~
📝 Definieren	TC00022 - Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	
😂 Planen	Ubersicht Eigenschaften Benutzerdefiniert Schritte (6) Anhänge Revisionen Issues (5) Aufgaben (1) Ergebnisse (25) Änderungen	Zurück
🏟 Ausführen	Neu Testsegment einfügen	
욙 Auswerten	□ + 国 ⑪ 6 Einträge-Seite 1 von 1 🖌 < 1 ▶ ▶ 10 ▼	⊞ ⇔ ∷
🗲 Konfigurieren	Schritt 1: Starten Sie die Anwendung durch das drücken der blauen Symbolschaltfläche. I d K3 [X] b Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung Benutzerdefiniert Anhänge	┎╜┛╸
	Starten Sie die Anwendung durch das drücken der blauen Symbolschaltfläche.	
	Schritt 2: Wählen Sie die Taste , die sich im Hauptmenü befindet.	0 🛍 –
	Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung Benutzerdefiniert Anhänge	
	Wählen Sie die Taste Deuerauftrag	
	, die sich im Hauptmenu beimoer.	

Abbildung 6.42. Die Ansicht "Schritte"

Jeweils rechts neben den Eingabefeldern *Beschreibung*, *Erwartetes Ergebnis*, *Vorbedingung* und *Nachbedingung* befindet sich der Button Anhang einfügen (siehe <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>).

≡ 📌 K L A R	OS TEST MANAGEMENT DEFinanz-Tracker ॼ ☴ Q × @ × ▲ ×
📝 Definieren	TC00021 - Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags. 💿 🖨 🗌 😋 😏
🏝 Planen	Speichern Verwerfen Zurück Übersicht Eigenschaften Benutzerdefiniert Schritte (6) Anhänge (1) Revisionen Issues (2) Aufgaben Ergebnisse Änderungen
🏟 Ausführen	Neu Testsegment einfügen
욙 Auswerten	+ 図 前 6 Einträge-Seite 1 von 1 M 1 ► M 10 田 11 12
🖌 Konfigurieren	Schritt 1: Testsegment I de Anwendung durch Drücken der blauen Symbolschaltfläche .
	Name * Größe * Dateityp * Version * Erstellt von * Aktion. Keine Einträge vorhanden
	Schritt 2: Wählen Sie die Taste , die sich im Hauptmenü befindet.
	Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung Benutzerdefiniert Anhänge Wählen Sie die Taste Deuerauftragi , die sich im Hauptmenü befindet. 4
	Schritt 3: Wählen Sie die Schaltfläche Löschen. 🏌 🕴 KKI (XX) (V) 👔 🛑
	Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung Benutzerdefiniert Anhänge
	Wählen Sie die Schaltfläche Löschen.

Abbildung 6.43. Der Dialog "Anhang einfügen"



Anmerkung

Die auszuwählenden Anhänge müssen zuvor in der Ansicht Anhänge hochgeladen sein.

Wurde ein Anhang ausgewählt, wird mit Klick auf den Button Hinzufügen eine Referenz in Form von %*inline-att:[attachment-identifier]*%, z. B. %*inline-att:*9e3df155-359f-40e7-8c78-7ae9aa-c6a8ab% angelegt.

≡ 坐 к L A R	OS TEST MANAGEMENT DE Finanz-Tracker 🖬 🗮 🔍 🤇	0° 1°
📝 Definieren	TC00022 - Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags. 💿 🚍	
🏝 Planen	Ubersicht Eigenschaften Benutzerdefiniert Schritte (6) Anhänge Revisionen Issues (5) Aufgaben (1) Ergebnisse (25) Änderungen	Zurück
🏟 Ausführen	Neu Testsegment einfügen	
e Auswerten	+ 🖾 III 6 Einträge - Seite 1 von 1 + 4 1 > 10 - Schritt 1: Starten Sie die Anwendung durch das drücken der blauen Symbolschaltfläche. 1 4 10 -	
🔑 Konfigurieren	Aktion Starten Sie die Anwendung durch das drücken der blauen Symbolschaltfläche.	-8
	Erwartetes Ergebnis Sie gelangen auf das Startbild. %INLINE-ATT:ea752534-18da-4f56-9bd4-c35bd3e703e9%	÷.
	Vorbedingung Die Anwendung ist auf dem Gerät installiert. Wenn nicht, folgen Sie den Anweisungen: https://finanztracker/install	-e
	Nachbedingung Die Anwendung ist erfolgreich gestartet.	÷
	Anhänge	£
	Schritt 2: Wählen Sie die Taste , die sich im Hauptmenü befindet.	0 🛍 –
	Aktion Wählen Sie die Taste Douerauftrag , die sich im Hauptmenü befindet.	÷
	Erwartetes Ergebnis Es wird eine Liste aller Daueraufträge angezeigt.	¢

Abbildung 6.44. Ein Verweis auf einen Anhang im Feld Erwartetes Ergebnis

Handelt es sich bei dem Anhang um ein Bild, wird diese Referenz bei der Ausführung des Testfalls durch ein skalierbares Vorschaubild während der Ausführung des Testfalls ersetzt. Andernfalls wird der Verweis durch einen Hyperlink ersetzt, der den Anhang in einem neuen Browser-Tab öffnet.



Abbildung 6.45. Ein Verweis auf einen Anhang wird durch eine Vorschau des Anhangs ersetzt



Abbildung 6.46. Ein Verweis auf einen Anhang wird durch einen Hyperlink zum Anhang ersetzt

6.7.2.6. Anhänge

Einem Testfall können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.7.2.7. Revisionen

Testfälle können in unterschiedlichen Revisionen vorliegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.5, "Revisionen"</u>.

6.7.2.8. Issues

In dieser Ansicht werden alle Issues aufgelistet, die mit diesem Testfall verknüpft sind.



Abbildung 6.47. Die Ansicht "Issues"

Die Tabelle zeigt die folgenden Werte:

ID	Die ID des Issues im externen Issue-Management-System.
System	Das externe Issue-Management-System dieses Issues.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Zusammenfassung	Die Zusammenfassung des Issues.
Erstellt	Das Datum, an dem dieser Issue erstellt wurde.
Ersteller	Der Benutzer, der dieser Issue erstellt hat.
Zugewiesen	Der Benutzer, dem dieser Issue zugewiesen ist.
Priorität	Die Priorität des Issues. Die möglichen Werte werden durch das externe Issue-Management-System vorgegeben.
Status	Der Status des Issues. Die möglichen Werte werden durch das externe Issue-Management-System vorgegeben.
Aktionen	Die ausführbaren Aktionen.

6.7.2.8.1. Ein neues Issue erstellen

Mit Klick auf <u>Neu</u> öffnet sich die Seite *Issue Details* (<u>Abschnitt 9.6.6, "Issue - Details</u>"), auf der neue Issues erstellt und mit dem Testfall verknüpft werden können. Mit Klick auf <u>Link</u> werden bestehende Issues mit dem Testfall verknüpft (<u>Abschnitt 9.6.7, "Einen Issue verlinken"</u>).

6.7.2.8.2. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- 🖸 Öffnen
- 前 Löschen

6.7.2.9. Aufgaben



Abbildung 6.48. Die Ansicht "Aufgaben"

In dieser Ansicht werden alle Aufgaben aufgelistet, in denen dieser Testfall enthalten ist. Diese Aufgaben können direkt aus dieser Ansicht heraus ausgeführt werden. Weitere Informationen zur Verwaltung von Aufgaben finden Sie unter <u>Abschnitt 7.1, "Aufgaben"</u>.

6.7.2.10. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Testsuite zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.7.2.11. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Testfalles an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

6.8. Testsuiten

Eine *Testsuite* ist eine Zusammenstellung von Testfällen, die gemeinsam ausgeführt werden sollen. Die Ergebnisse der Ausführung einer Testsuite können verwendet werden, um zu überprüfen und sicherzustellen, dass ein System bestimmte Spezifikationen oder Anforderungen erfüllt.

🔳 👱 K L A R (OS TEST	MANA	GEMENT	C	E Finanz-Tracker 🛅 🗮	۵ ک	× @~ ± ~
📝 Definieren	Testsuiten						
Planen	Neu					Speichern	Verwerfen
	+ 6 6 %	00	7 Einträge - Seite 1 von 1 🛛 🕅		Alle anzeigen		Q × ⅔ ≡
🏚 Ausführen	L ID≑ ♀	🕈 🗘 🕈	Name 🗢		Testsystem 🗢	Testfälle 🗢	Aktion
	TS00008	8 1.0	Überprüfen Sie die Verbindung zum Server		Finance Tracker 2.1.0	1	嫦健₿₫
4	TS00007	>\$ 1.0	Daueraufträge bearbeiten			3	₡₽₽₫
C Auswerten	TS00006	1.0	Übersicht			7	┏┏₽₫
	TS00004	8 1.0	Verbingungen überprüfen		Finance Tracker 1.0.0	2	₡₽₽₫
🔑 Konfigurieren	TS00003	🛕 1.0	Aktivitäten verschiedener Zeiträume anzeigen			4	┏┖╘╓
	TS00002	3 1.0	Überweisungen testen			8	嫦健ᇢ◍
	TS00001	8 1.0	Dashboard aktualisieren			4	┏┖₴₫
	ID						Aktion
	Neu					Speichern	Verwerfen

Abbildung 6.49. Die Seite "Testsuiten"

6.8.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Testsuiten in einer Tabelle an. Neue Testsuiten werden hier erstellt.

Die Tabelle zeigt die folgenden Werte an:

ID	Wird automatisch zugewiesen.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision der Testsuite.
Name	Der Name der Testsuite.
Testsystem	Das Testsystem, das dieser Testsuite zugeordnet sind.
Testfälle	Die Anzahl der Testfälle, die diese Testsuite enthält.
Aktion	Die ausführbaren Aktionen.

Die Einträge *Name* und *Testsystem* lassen sich mit einem Klick in die entsprechenden Felder direkt in den Tabellenzeilen bearbeiten.

6.8.1.1. Eine neue Testsuite erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun können der Name und das Testsystem festgelegt werden.

Mit <u>Speichern</u> wird die neue Testsuite erzeugt und gespeichert. Die ID der Testsuite (TS00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *TS00001* gelangen Sie auf die Detailseite des Testsystems.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst mit Klick auf Speichern werden die Daten in die Datenbank übernommen.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Ungespeicherte Änderungen

In allen mit roten IDs versehenden Zeilen befinden sich Änderungen, die noch nicht gespeichert sind.

6.8.1.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- Duplizieren
- Druckansicht öffnen
- 前 Löschen

Wurde eine Testsuite gelöscht, ist sie zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testsuiten stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ⊘ Die Testsuite unwiederbringlich aus der Datenbank entfernen

6.8.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testsuiten gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testsuiten aus, auf die die Aktion angewendet werden soll.

Für Testsuiten werden folgende Massenaktionen unterstützt:

- + Testfälle zuweisen
- 🕜 Bearbeiten
- Duplizieren
- Neue Revision erzeugen
- 🔒 Druckansicht öffnen
- ff Löschen
- Wiederherstellen (nur Administrator)
- ⊘ Die Testsuite unwiederbringlich aus der Datenbank entfernen (nur Administrator).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

6.8.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 옮 Kategorisieren
- √ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

Q Suchen

- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

6.8.2. Detailseite

Jede Testsuite verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID der betreffenden Testsuite oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der als letztes ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Anhänge, Revisionen, Aufgaben, Ergebnisse und Änderungen.



Warnzeichen in der ID-Spalte

Ein manueller Testfall der keine Schritte enthält, ist nicht ausführbar. Um dies anzuzeigen, erscheint ein Warnzeichen (\blacktriangle) in der Spalte ID.

6.8.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

Review	Legt eine Review-Aufgabe an.
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der betreffenden Testsuite erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.1, "Druckansichten"</u> .
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.
G D Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testsuiten gewechselt werden.

6.8.2.2. Übersicht



Auf der Ansicht Übersicht werden folgende Werte angezeigt:





Ein Testsystem zuordnen

Das Zuordnen einer Testsuite zu einem Testsystem ist nur ein Hinweis für den Benutzer. Zugeordnete Testsuiten können weiterhin mit anderen Testsystemen ausgeführt werden.

Aktuelle Testläufe	Diese Grafik zeigt die Anzahl der bestandenen, fehlgeschlage- nen, fehlerhaften, unklaren und übersprungenen Testfallergeb- nisse für die letzten fünf Testläufe dieser Testsuite.
Ausführungsdauer dieser Test- suite	Dieses Diagramm zeigt die minimale, maximale und durch- schnittliche Ausführungszeit für diese Testsuite an.
Testsystem - Übersicht	Diese Tabelle zeigt das neueste Ergebnis dieser Testsuite für jedes Testsystem und jede Testumgebung, in der diese Test- suite ausgeführt wurde. Wenn Sie mit dem Mauszeiger über ein einzelnes Ergebnis fahren, erhalten Sie eine detailliertere Übersicht über die einzelnen Testfallergebnisse dieser Test- suite. Mit Klick auf ein Ergebnis gelangt man zur Detailseite des Testfallergebnisses.

6.8.2.3. Eigenschaften

In dieser Ansicht (<u>Abbildung 6.22</u>) werden die folgenden Attribute der Testsuite angezeigt:

≡	Y KLAR	OS TEST MANAGEMENT DE Finanz-Tracker ቛ		۹ ،	< 0× ±×
ľ	Definieren	TS00007 - Daueraufträge bearbeiten		۲	⊖ C C C
*2:	Planen	Sp Übersicht Eigenschaften Benutzerdefiniert Anhänge Revisionen Aufgaben (11) Ergebnisse (21) Änderungen	veichern V	'erwerfen	Zurück
\$	Ausführen	ID TS0007 Name Daueraufträge bearbeiten Testsystem	~		
¢	Auswerten	Z → 3 Entráge- Seite 1 von 1 K ≤ 1 ► N 10 ▼ Verfolsbarkeit	Ausführu	ng Status	Schritte Aktion
عر	Konfigurieren	Image:	Manuell Manuell Manuell	Entwurf Entwurf Entwurf	5 1 - 5 1 - 5 1 -
		+ 24 Einträge-Seite 1 von 3 🕅 ◀ 1 2 3 ▶ 🕅 10 ✔ ♀ ID ♦ Revision Name ♦ Verfolgbarkeit ♦	品 Ausführung \$	Status 🗘 S	Q × & Ξ chritte \$ Aktion
		TC00024 1.0 Überprüfen Sie die Verbindung zum Server Requirement 1.2.4 in testplan_16.txt TC00023 1.0 Einen Dauerauftrag ändern. Requirement 1.2.4 in testplan_16.txt TC00022 1.0 Aktualisieren Sie das Dashboard durch Hinzufügen eines Requirement 1.2.4 in testplan_16.txt	Automatisiert Manuell Manuell	Entwurf Entwurf Entwurf	0 + 5 + 6 +
		Dauerauftrags. TC00021 1.0 Aktualisieren Sie das Dashboard durch Stornierung eines Requirement 1.2.4 in testplan_16.txt Dauerauftrags. TC0002 1.0 Hundling die der beschend durch Stornierung eine	Manuell	Entwurf	6 +
		COUD20 1.0 Axtualisefert set das Usainbard durch stormerung eines requirement 1.2.4 in testipar_16.xt Lastschriftmandats. TC00019 1.0 Testen der Verbindung zur Datenbank. //FinanceTracker/arc/test/java/de/sabrina/testing/web/Connection.java	Automatisiert	Entwurf	° +
		TC00018 1.0 Testen der Verbindung zum Server der Bank. /FinanceTracker/src/test/java/de/sabrina/testing/web/Co nnection.java	Automatisiert	Entwurf	0 +
		TC00017 1.0 Lastschriftmandat löschen. Requirement 1.2.4 in testplan_16.txt	Manuell	Entwurf	5 + 5 +
		© TC00015 1.0 Fügen Sie einen Dauerauftrag hinzu. Requirement 1.2.4 in testplan_16.txt	Manuell	Entwurf	5 +
		Erstellt vor 4 Jahren von Talal Arif	Zuletzt geändert vo	or 4 Jahren von	Felix Mustermann
		sp	elchern V	'erwerfen	Zurück

Abbildung 6.51. Die Ansicht "Eigenschaften"

In dieser Ansicht können der Testsuite Testfälle hinzugefügt oder wieder entfernt werden. Die obere Tabelle zeigt die in der Testsuite enthaltenen Testfälle an, sie unterstützt das Umordnen durch Ziehen und Ablegen. Die untere Tabelle zeigt die verfügbaren Testfälle an.

Für die Testfälle der Testsuite werden folgende Massenaktionen unterstützt:

- E Mehrfachvorkommen entfernen
- Testfälle entfernen

In dieser Ansicht werden die folgenden Attribute der Testfälle angezeigt:

ID	Wird automatisch zugewiesen.
Name	Der Name des Testfalls.
Verfolgbarkeit	Die Anforderung, die dieser Testsuite zugeordnet ist.
Ausführung	Manuell oder Automatisiert.
Status	Der Status des Testfalls. Mögliche Werte sind Gesperrt, Geneh- migt, Entwurf und Auslassen.
Schritte	Die Anzahl der zugehörigen Testschritte.
Aktion	Entfernt oder fügt den Testfall zur Testsuite hinzu.

6.8.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

6.8.2.5. Anhänge

Einer Testsuite können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

6.8.2.6. Revisionen

Testsuiten können in unterschiedlichen Revisionen vorliegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.5, "Revisionen"</u>.

6.8.2.7. Aufgaben

≡ 📌 K L A R	O S TEST MANAGEMENT	DE Finanz-Tracker 🛛 🔤	५ × 0∕ ≛ र
📝 Definieren	TS00007 - Daueraufträge bearbeiten		④ 🖨 🎵 😌 🕄
🏝 Planen	Übersicht Eigenschaften Benutzerdefiniert Anhänge Revisionen	Aufgaben (11) Ergebnisse (21) Änderungen	hern Verwerfen Zurück
🏟 Ausführen	11 Einträge - Seite 1 von 2 K ◀	1 2 ► N 10 ▼ Priorität ♦ Status ♦ Fortschritt ♦ Erfola	▼ Q × A ≡ ◆ Fälling ◆ Zugewiesen ◆ Aktion
 Auswerten Konfigurieren 	10.0 Zusammenfassung ● →09009345 Edit standing orders →09009356 Edit standing orders →0900936 Edit standing orders →0900936 Edit standing orders →0900937 Edit standing orders →0900938 Edit standing orders →0900939 Edit standing orders →0900939 Edit standing orders →0900939 Edit standing orders →0900940 Edit standing orders →0900940 Daueraufträge bearbeiten →0900940 Daueraufträge bearbeiten	Priorital * Status * Fortschrift * Erfolg A Hoch Erledigt 100% 100% A Hoch Erledigt 100% 66% A Hoch Erledigt 100% 0% A Hoch Erledigt 100% 100% Zul Speicl Speicl	Ealling Zugewissen Aktion 25.10.2020 Tanja Toppler Image To



In dieser Ansicht werden alle Aufgaben aufgelistet, in denen diese Testsuite enthalten ist. Diese Aufgaben können direkt aus dieser Ansicht heraus ausgeführt werden. Weitere Informationen zur Verwaltung von Aufgaben finden Sie unter <u>Abschnitt 7.1, "Aufgaben"</u>.

6.8.2.8. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Testsuite zugeordneten Testergebnisse an. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u>.

6.8.2.9. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieser Testsuite an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> <u>"Änderungshistorie"</u>.

Kapitel 7. Planen

enterprise edition Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Dieses Kapitel beschreibt den Bereich *Planen* in Klaros-Testmanagement. Hier können <u>Aufgaben</u> erstellt und bearbeitet werden.

7.1. Aufgaben verwalten

Eine *Aufgabe* dient der Planung, Organisation und Verfolgung einer Testaktivität. Eine Aufgabe kann aus der Ausführung oder Überprüfung eines Testfalls oder einer Testsuite bestehen oder die textuelle Beschreibung einer beliebigen anderen möglichen Aktivität enthalten. Dieses Kapitel zeigt, wie Sie Aufgaben erstellen und bearbeiten.

7.1.1. Übersichtsseite

Die Übersichtsseite zeigt alle im ausgewählten Projekt vorhandenen Aufgaben in einer Tabelle an. Neue Aufgaben werden hier erstellt.

≡	Y KLAR	OS TEST M	МА	NAGEMENT					DE Finanz-Ti	racker 🔳	0		٩	×	6 ~	* ~
ľ	Definieren	Aufgaben verwa	alter	1												
**	Planen	Neu										Speich	ern		Verwe	rfen
		▷ Ư C A Ó	0	22 Einträg	e - Seite 1 von 3	M 🔺 🚺	2 3	► N 10	*		Alle a	inzeigen		Q	×	& ≡
•	Ausführen	ID 🗢	0	Zusammenfassung 🖨	Testsystem 🗘	Priorität 🖨	Status 🖨	Testfälle 🗘 I	Fortschritt 🗢	Erfolg 🖨	Fällig 🖨	Zugewiesen 🗢		A	ction	
Ģ	Auswerten	JOB00360		Ausführung TC00022 - Aktualisierung des Dashboards durch Hinzufügen eines Dauerauftrags.	Finance Tracker 2.0.0	▲ Hoch	Neu	1	9%		26.10.2020	Markus Meyer	D	C () 🖶 1	1
ų	Konfigurieren	JOB00359		Ausführung TC00021 - Aktualisierung des Dashboards durch Stornierung eines Dauerauftrags.	Finance Tracker 2.0.0	▲ Hoch	Neu	1	0 %		26.10.2020	Markus Meyer	D	C () 🖨 1	<u>0</u> -5
		JOB00358		Ausführung TC00020 - Aktualisierung des Dashboards durch Stornierung einer Belastungsermächtigung.	Finance Tracker 2.0.0	▲ Hoch	Neu	1	0 %		26.10.2020	Felix Mustermann	D	C () 🗗 I	1
		JOB00357		Ausführung TC00017 - Lastschriftmandat löschen.	Finance Tracker 2.0.0	 Hoch 	Neu	1	0 %		26.10.2020	Felix Mustermann	Þ	ľ	9	û 🎝
		JOB00356		Ausführung TC00016 - Daueraufträge löschen.	Finance Tracker 2.0.0	▲ Hoch	Neu	1	0 %		26.10.2020	Markus Meyer	Þ	ľ) (û 🏞
		JOB00355	24	Ausführung TC00015 - Einen Dauerauftrag hinzufügen.	Finance Tracker 2.0.0	 Hoch 	Erledigt	1	100%		26.10.2020	Felix Mustermann	Þ	ľ) 🖯 I	â 🖞
		JOB00354	0	Review TC00017 Lastschriftmandat löschen.	Finance Tracker 2.1.0	ጰ Kritisch	Erledigt	1	0 %	100%	26.10.2020	Sabrina Gidley	Þ	ľ) (û よ
		JOB00240 (65)		Testen Sie alle Funktionen für realese 2.1.0	Finance Tracker 2.1.0	ጵ Kritisch	In Arbeit	79	84%	76%	26.10.2020	Markus Meyer	Þ	ľ	181	â 🌣
_		JOB00218 (21)		Testen Sie alle Funktionen auf einer Smartwatch	Finance Tracker 2.1.0	Xiedrig	In Arbeit	29	86%	80%	26.10.2020	Markus Meyer	Þ	ľ) (Û 🌣
		JOB00172 (43)		Testen Sie alle Grundfunktionen für	Finance	ጵ Kritisch	In Arbeit	54	90%	80%	26.10.2020	Markus Meyer	Þ	C () 👌 1	ŵ 🌣

Abbildung 7.1. Die Seite "Aufgaben verwalten"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Zusammenfassung	Die Zusammenfassung der Aufgabe.
Testsystem	Das dieser Aufgabe zugeordnete Testsystem.
Priorität	Die Priorität der Aufgabe. Mögliche Werte sind <i>Trivial</i> , <i>Niedrig</i> , <i>Hoch</i> , <i>Kritisch</i> und <i>Blocker</i> .
Status	Der Status der Aufgabe. Mögliche Werte sind Neu, Wieder eröffnet, In Arbeit, Erledigt, Geschlossen und Abgelehnt.

Testfälle	Die Anzahl der Testfälle, die dieser Aufgabe und ihrer Unter- aufgaben zugeordnet sind.
Fortschritt	Der Prozentsatz der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.
Erfolg	Die Erfolgsrate der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben in Prozent.
Fällig	Das Datum, bis zu dem diese Aufgabe beendet sein soll.
Zugewiesen	Der Benutzer, dem die Aufgabe zugewiesen ist.
Aktion	Die ausführbaren Aktionen.

7.1.1.1. Eine neue Aufgabe erstellen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Nun kann eine Zusammenfassung definiert werden.

Mit <u>Speichern</u> wird die neue Aufgabe erzeugt und gespeichert. Die ID der Aufgabe (JOB00001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Klick auf die ID *JOB00001* gelangen Sie auf die Detailseite der Aufgabe.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs haben Änderungen und sind noch nicht gespeichert!

7.1.1.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- Aufgabe Unterordnen
- Bearbeiten, (siehe <u>Abschnitt 7.1.2, "Detailseite"</u>)
- Duplizieren
- Druckansicht öffnen
- fi Löschen

✿ Ausführen, ♣ Manuell ausführen, ♣ Review durchführen, ♣ Testergebnisse importieren, ♣ Neu öffnen oder aktualisieren (siehe <u>Abschnitt 7.1.1.6, "Aufgaben ausführen"</u>).

Wurde eine Aufgabe gelöscht, ist sie zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Aufgaben stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen (nur Administrator)
- ☆ Die Aufgabe unwiederbringlich aus der Datenbank entfernen

7.1.1.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Aufgaben gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Aufgaben aus, auf die die Aktion angewendet werden soll.

Für Aufgaben werden folgende Massenaktionen unterstützt:

- ▷ Aufgaben einer anderen Aufgabe unterordnen
- 🛯 Bearbeiten
- Duplizieren
- 🔒 Druckansicht öffnen
- 前 Löschen
- Wiederherstellen (nur Administrator).
- ⊘ Die Aufgabe unwiederbringlich aus der Datenbank entfernen (nur Administrator).
- O Aufgaben einer Iteration zuweisen.

Massenaktionen sind ausführlich in <u>Abschnitt 5.2.3.1.5, "Massenaktionen"</u> beschrieben.

7.1.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- √ Filter / Sortieren
- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

7.1.1.5. Untergeordnete Aufgaben anordnen

Untergeordnete Aufgaben können mit den Pfeil-Icons in der Spalte *Aktion* neu angeordnet werden. Das Icon \ddagger öffnet einen Dialog, in dem der Benutzer die Aufgabe an eine neue Position verschieben kann. Die Icons \triangleleft und \triangleright ändern die Aufgabenhierarchie. Das Icon \triangleleft verschiebt die Aufgabe in die Hierarchieebene der Elternaufgabe. Das Icon \triangleright öffnet einen Dialog, in dem die neue übergeordnete Aufgabe ausgewählt werden kann.

Übergeordnete Aufgaben besitzen die folgenden Eigenschaften:

• Einer übergeordnete Aufgabe kann kein Testfall oder Testsuite zugewiesen werden.

- Eine übergeordnete Aufgabe kann andere übergeordnete Aufgaben mit weiteren untergeordneten Aufgaben enthalten.
- Ist das Icon <a>> einer Aufgabe ausgegraut dargestellt, ist keine Aufgabe zur Unterordnung verfügbar.

Um eine Aufgabe einer anderen Aufgabe unterzuordnen, klicken Sie auf das Icon Nun können Sie auswählen, welcher Aufgabe die Aufgabe untergeordnet werden soll. Neben der ID der übergeordneten Aufgabe wird dann die Anzahl der untergeordneten Aufgaben angezeigt. Die angezeigte Erfolgs- und Fortschrittsrate der übergeordneten Aufgabe wird automatisch aus den Testergebnissen der ihr untergeordneten Aufgaben abgeleitet.



Warnung

Aufgaben, denen bereits ein Testfall oder eine Testsuite zugewiesen wurde, können keine Unteraufgaben enthalten und werden daher nicht als mögliche Auswahl im Dialog Übergeordnete Aufgabe dargestellt.



Änderungen an übergeordneten Aufgaben

Änderungen an übergeordneten Aufgaben betreffend Testumgebung, Testsystem und Iteration werden automatisch an deren untergeordneten Aufgaben weitergeleitet.

7.1.1.6. Aufgaben ausführen

Eine detaillierte Beschreibung, wie Aufgaben ausgeführt werden können, finden Sie unter Abschnitt 8.1.3, "Eine Aufgabe ausführen".

7.1.2. Detailseite

Jede Aufgabe verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf die ID der betreffenden Aufgabe oder auf das Icon
rechts in der Aktionsspalte gelangen Sie zu der als letztes ausgewählte Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

Es stehen die folgenden Ansichten zur Verfügung: Übersicht, Eigenschaften, Benutzerdefiniert, Abhängigkeiten, Anhänge, Ergebnisse, Kommentare, Arbeitsprotokoll und Änderungen.

7.1.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Symbole. Folgende Aktionen lassen sich hier vornehmen:

🗘 🎝 🕹 - 🔁 🕹 Ausfüh-	Mit Klick auf das angezeigte Icon wird die Aufgabe ausgeführt.
ren	Das Ausführen von Aufgaben ist ausführlich in <u>Abschnitt 7.1.1.6, "Aufgaben ausführen"</u>) beschrieben.
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der Aufgabe erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.

	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.
G 🕤 Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Aufgaben gewechselt werden.

7.1.2.2. Übersicht



Abbildung 7.2. Die Ansicht "Übersicht"

Тур	Der Aufgabentyp (<i>Testausführung - Automatisch, Testausfüh- rung - Manuell, Review</i> oder <i>Auftrag</i> (wenn der Aufgabe weder ein Testfall noch eine Testsuite zugeordnet ist).
Status	Der Status der Aufgabe.
Priorität	Die Priorität der Aufgabe.
Zugewiesen	Der Zuständige für diese Aufgabe.
Startdatum	Der früheste Zeitpunkt, an dem diese Aufgabe ausgeführt wer- den soll.
Fällig	Der Zeitpunkt, zu dem die Ausführung der Aufgabe spätestens abgeschlossen sein muss.

Testfälle	Die Anzahl der Testfälle in dieser Aufgabe und oder ihren Unteraufgaben.
Testfall / Testsuite	Der dieser Aufgabe zugeordnete Testfall oder Testsuite.
Testsystem	Das Testsystem, das bei der Ausführung dieser Aufgabe ver- wendet werden soll.
Testumgebung	Die Testumgebung, die bei der Ausführung dieser Aufgabe ver- wendet werden soll.
Geschätzte Dauer	Die geschätzte Dauer, die für die Durchführung dieser Aufgabe benötigt wird.
Fortschritt	Die Fortschrittsrate zeigt an, wie viele Testfälle dieser Aufgabe bereits ausgeführt wurden, unabhängig von ihrem Ergebnis.
Erfolg	Die Erfolgsquote zeigt an, wie viele der Testfälle, die von die- ser Aufgabe abgedeckt werden, im letzten Testlauf erfolgreich ausgeführt wurden.
Gesamte Arbeitszeit	Die für die Testausführung aufgewendete Zeit. Dieser Wert ergibt sich aus der Summe der Ausführungszeiten aus dem Arbeitsprotokoll.
Letzter Ausführender	Der Benutzer, der die Aufgabe zuletzt ausgeführt hat.
Aktuelle Testläufe	Dieses Diagramm zeigt die kumulierten Testfallergebnisse der letzten Testläufe für diese Aufgabe in absteigender Reihenfol- ge an.
Ausführungszeit	Dieses Diagramm zeigt die minimale, durchschnittliche und maximale Ausführungszeit für diese Aufgabe an.
Aktuelle Testläufe	Diese Tabelle listet die letzten abgeschlossenen Testläufe für diese Aufgabe auf.



Anmerkung

Diese Tabelle zeigt standardmäßig bis zu fünf Testläufe an. Durch Klicken auf das Icon *Alle anzeigen* wird man auf die Ansicht *Ergebnisse* weitergeleitet, wo alle Testläufe aufgelistet sind.

7.1.2.3. Eigenschaften

In der Ansicht Eigenschaften können folgende Attribute geändert werden:

Тур	Der Aufgabentyp (<i>Testausführung - Automatisch, Testausfüh- rung - Manuell, Review</i> oder <i>Auftrag</i> (wenn der Aufgabe weder ein Testfall noch eine Testsuite zugeordnet ist).
Zusammenfassung	Die Zusammenfassung der Aufgabe.

Beschreibung Die ausführliche Beschreibung der Aufgabe.

Priorität

Status

Die Priorität der Aufgabe. Mögliche Werte, in aufsteigender Reihenfolge ihrer Schwere sind *Trivial, Niedrig, Hoch, Kritisch* und *Blocker*.

> Der Status der Aufgabe. Mögliche Werte sind Neu, In Arbeit, Erledigt, Geschlossen, Wieder eröffnet und Abgelehnt.



Abbildung 7.3. Statusgraph einer Aufgabe

Fortschritt	Die Fertigstellung der Aufgabe in Prozent.
Geschätzte Dauer	Die geschätzte Dauer in Tagen, Stunden, Minuten und Sekun- den, die die Aufgabe voraussichtlich in Anspruch nehmen wird (Tage HH:MM:SS).
Startdatum	Der früheste Zeitpunkt, zu dem diese Aufgabe ausgeführt wer- den soll.
Fällig	Der Zeitpunkt, zu dem die Ausführung der Aufgabe spätestens abgeschlossen sein muss.
Zugewiesen	Der Benutzer, der für die Aufgabe verantwortlich ist.
Iteration	Die Iteration, der diese Aufgabe zugeordnet ist. Eine Aufgabe kann sich zu einem Zeitpunkt nur in einer Iteration befinden. Ein Wechsel der Iteration ist jederzeit möglich.

Testfall	Der Testfall, der in dieser Aufgabe ausgeführt werden soll. Eine Aufgabe kann entweder einen Testfall oder eine Testsui- te enthalten. Dieses Feld ist nicht verfügbar, wenn die Aufgabe Unteraufgaben enthält oder vom Typ <i>Auftrag</i> ist.
Testsuite	Die Testsuite, die mit dieser Aufgabe ausgeführt werden soll. Eine Aufgabe kann entweder einen Testfall oder eine Testsuite enthalten. Dieses Feld ist nicht verfügbar, wenn die Aufgabe Unteraufgaben enthält oder vom Typ <i>Auftrag</i> ist.
Testsystem	Das Testsystem, für das diese Aufgabe ausgeführt werden soll.
Testumgebung	Die Testumgebung, in der diese Aufgabe ausgeführt werden soll.

≡	Y KLAR	OS TEST M	MANAGEMENT DE Finanz-Tracker 🖬 🗮	२ × छ ≁ ≛×
Ľ	Definieren	JOB00360 - Aust	sführung TC00022 - Aktualisierung des Dashboards durch Hinzufügen eines Dauerauftrags.	♣; 🔁 🗋 🔤 😜
*	Planen	Übersicht Eigen	Speichern enschaften Benutzerdefiniert Abhängigkeiten Anhänge Ergebnisse Kommentare Arbeitsprotokoll Änderungen	Verwerfen Zurück
\$	Ausführen	ID Typ	ID JOB00360 /₽ Testausführung ❤	
Ģ	Auswerten	Ausführung Zusammenfassung	1g Manuell 1g Ausführung TCD0022 - Aktualisierung des Dashboards durch Hinzuffigen eines Dauerauftrags	
<i>بر</i>	Konfigurieren	Beschreibung	g g Testet die automatische Aktualisierung des Dashboards durch Drücken der	
			Dauerauftrag hinzufügen	
			Der Dauerauftrag sollte folgende informationen enthalten:	
			Begünstigter der Zahlung: Max Mustermann	
			IBAN: DE19 1234 1234 1234 1234 1234 1234 1234 1234	
			Rechnung:38745	
			Zeitintervall:monatlich	
			Stellen Sie sicher, dass die Anwendung vor dem Testen installiert ist!	

Abbildung 7.4. Die Ansicht "Eigenschaften"

7.1.2.4. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

7.1.2.5. Abhängigkeiten

In der Ansicht *Abhängigkeiten* können Abhängigkeiten zwischen Aufgaben definiert werden, damit bestimmte Aufgaben erst dann ausgeführt werden können, wenn zuvor festgelegte Bedingungen erfüllt sind. Für jede Aufgabe kann eine beliebige Anzahl an Abhängigkeiten zu einer oder mehreren anderen Aufgaben definiert werden, die erfüllt werden müssen bevor die Aufgabe ausgeführt werden kann. In dieser Ansicht finden Sie eine Übersicht aller vorausgesetzten und abhängigen Aufgaben der aktuell ausgewählten Aufgabe.

≡	👱 KLAR	OS TES	ST MANAG	GEMEN	١T					DE Finanz-Tracker 📑 📑	० छ∽ ≛ ∽
ľ	Definieren	JOB00360 -	Ausführung T	C00022 - A	Aktualisierung de	s Dashbo	oards durch H	inzufügen eine	s Dauera	auftrags.	♣ 🖨 🏾 😋 🗲
*	Planen	Übersicht	Eigenschaften	Benutzer	definiert Abhän	gigkeiten (5) Anhänge	Ergebnisse	Kommen	Speichern tare Arbeitsprotokoll Änderunge	Verwerfen Zurück
±	Ausführen	Vorausge	setzte Aufgabe	n							
		Aufgabe 🗢			Fortschritt 🖨	Erfolg 🖨		Ergebnis 🖨		Status \$	Aktion
	Auguartan	JOB00356	00		100%	75%	Bestanden			Geschlossen	d
	Auswerten	JOB00355	00		55%	10%	Fehler, Überspru	ingen, Unbekannt, F	Fehlschlag		C D
		JOB00358	00		0%	0%	Übersprungen			Abgelehnt	C ū
×	Konfigurieren	JOB00359	00		30%	20%	Bestanden			Erledigt	
		JOB00357			0%	0%				Neu, Erledigt, Abgelehnt, In Arbeit, Wiede	r eröffnet, Geschlossen 🛛 🖉 🔟
		Abhängig	e Aufgaben								
		Aufgabe 🗘	Portschritt	🗧 Erfolg 🖨			Ergebnis 🖨			Status 🖨	
		JOB00241	95%	85%	Fehler, Übersprung	en, Bestand	en			Erledigt, In Arbeit, Geschlossen	
		JOB00284	50%	15%	Fehler, Unbekannt,	Bestanden				Neu, Abgelehnt, Wieder eröffnet	
		Abhän	gigkeit hinzufügen								
		Erstellt vor 10 N	fonaten von Felix Mu	stermann						Zuletzt geänd	lert vor 8 Monaten von Sandra Selen
										Speichern	Verwerfen Zurück

Abbildung 7.5. Die Ansicht "Abhängigkeiten"

= 坐 K L A R	os test management	DE Fir	nanz-Tracker 🗃 🚆 🛃 🖉 🖉 V	* ~
📝 Definieren	JOB00360 - Ausführung TC00022 - Aktualisie	rung des Dashboards durch Hinzufügen eines Dauera	uftrags. 🛃 🕹 🗋 💭	3 🕤
😤 Planen	Übersicht Eigenschaften Benutzerdefiniert	Abhängigkeiten Anhänge Frgebnisse Kommentare	Speichern Verwerfen Zurü Arbeitsprotokoll Änderungen	ck
Ausführen	Abhängigkeit hinzufügen	JOB00359 - JOB00359 Ausführung TC0002 🗸		
G Auswerten	Erstellt vor 3 Jahren von Felix Mustermann	0%	Zuletzt geändert vor 3 Jahren von Sand	ra Selen
🖌 Konfigurieren	Erfol	0% ()	Speichern Verwerfen Zurü	ck
	Letztes Ergebni	Unbekannt Bestanden Ubersprungen Fehlschlag Fehler Unklar Neu Wieder eröffnet In Arbeit Erledigt Geschlossen		
		Abgelehnt Hinzufügen Abbrecher		

Abbildung 7.6. Der "Abhängigkeit hinzufügen" Dialog

Abhängigkeitskriterien

Aufgabe	Die vorausgesetzte Aufgabe.								
Fortschritt	Die minimale Fortschrittsrate der gewählten Aufgabe (0-100%).								
Erfolg	Die Mindest-Erfolgsrate der gewählten Aufgabe (0-100%).								
Letztes Ergebnis	Das letzte Ergebnis der gewählten Aufgabe (Bestanden, Fehl- schlag, Fehler, Unklar, Übersprungen oder Unbekannt). Es kön- nen mehrere Werte ausgewählt werden.								

Status

Der Status der gewählten Aufgabe. Mögliche Werte sind Neu, In Arbeit, Erledigt, Geschlossen, Wieder eröffnet und Abgelehnt.

Wenn eine Aufgabe nicht ausgeführt werden kann, weil die Abhängigkeitskriterien ihrer vorausgesetzten Aufgaben nicht erfüllt sind, wird anstelle des Ausführungs-Icons das Icon 2. angezeigt.

7.1.2.6. Anhänge

Einer Aufgabe können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6</u>, <u>"Anhänge"</u>.

7.1.2.7. Ergebnisse

Die Ansicht *Ergebnisse* enthält drei weitere Ansichten: *Testfallergebnisse*, *Testsuiteergebnisse* und *Testläufe*. Jede Ansicht zeigt die dieser Aufgabe zugeordneten Testergebnisse an. Weitere Informationen sind in <u>Abschnitt 5.2.3.2.7</u>, <u>"Testläufe und Ergebnisse"</u> zu finden.

7.1.2.8. Kommentare

Nach Klick auf den Button Kommentieren können hier Bemerkungen hinzugefügt werden.

≡ 👱 K L A R	o s test management	DE Finanz-Tracker 🖬 🗮 🚺 🖉 🖉 🕹 🗸
📝 Definieren	JOB00360 - Ausführung TC00022 - Aktualisierung des Dashboards durch Hinzufügen ei	ines Dauerauftrags. 🕹 🖨 🗋 😋 📀
🐴 Planen	Übersicht Eigenschaften Benutzerdefiniert Abhängigkeiten Anhänge Ergebnisse K	Speichem Verwerfen Zurück Kommentare Arbeitsprotokoll Änderungen
🏚 Ausführen	0 Einträge - Seite 1 von 1 🔣 🗸 🕨	
🕒 Auswerten	Keine Einträge vorhanden	
🖋 Konfigurieren	Kommentieren	
	Erstellt vor 4 Jahren von Felix Mustermann	Zuletzt geändert vor 4 Jahren von Sandra Selen
		Speichern Verwerfen Zurück
	Einen Kommentar hinzufügen	
	Kommentar Diese Aufgabe muss aktualisiert werden.	
	ОК	Abbrechen

Abbildung 7.7. Der Dialog "Einen Kommentar hinzufügen"

Der in diesem Dialog eingegebene Text wird als Kommentar zu der ausgewählten Aufgabe hinzugefügt, sobald der Button OK angeklickt und gespeichert wird. Um den Kommentar zu ändern, verwenden Sie das Icon 😰 um ihn zu entfernen, klicken Sie auf das Icon 前 .

7.1.2.9. Arbeitsprotokoll

Jeder Testlauf, der sich auf diese Aufgabe bezieht, wird in der Ansicht *Arbeitsprotokoll* angezeigt. Es ist auch möglich, der Aufgabe eigene zusätzliche Arbeitsprotokolle hinzuzufügen. Klicken Sie dazu auf den Button Zeit buchen . Anschließend wird ein Dialog für die Eingabe der Details angezeigt. Planen

≡ 👱 K L A R	OS TEST MANAGEMENT DE Finanz-Tracker ॼ
📝 Definieren	JOB00360 - Ausführung TC00022 - Aktualisierung des Dashboards durch Hinzufügen eines Dauerauftrags. 🎥 🛱 🗍 🤆 ᅌ
😫 Planen	Speichern Verwerfen Zurück Übersicht Eigenschaften Benutzerdefiniert Abhängigkeiten Anhänge Ergebnisse Kommentare Arbeitsprotokoll Änderungen
🏟 Ausführen	Gesamte Arbeitszeit 0 ms 0 Einträge-Seite 1 von 1 M ◀ ▶ M 10 ♥
욙 Auswerten	Erstellt von 🕈 Erledigt am 🕈 Dauer 🕈 Geschätzte Restzeit 🕈 Testlauf 🕈 🛛 Tätigkeitsbeschreibung 🕈 Geändert 💠 Äktion Keine Einträge vorhanden
🖌 Konfigurieren	Zeit buchen
	Erstellt vor 4 Jahren von Felix Mustermann Zuletzt geändert vor 4 Jahren von Sandra Selen
	Arbeit protokollieren Zuletz geandert vor 4 Janten von sandra Selen Tätigkeitsbeschreibung Vorbereiten des Test-Hosta Erfeldigt am 07.05.2024 Dauer 0000:00 Geschätzte Restzeit 00:00 Fortschritt 9% Speichern Abbrechen

Abbildung 7.8. Der Dialog "Arbeit protokollieren"

In das Feld *Dauer* muss eine Zeitspanne eingetragen werden, damit die Tätigkeit protokolliert werden kann. Klicken Sie dann auf Speichern , um Ihre protokollierte Arbeit zu speichern. Um einen Arbeitsprotokolleintrag zu ändern, verwenden Sie das Icon ② , um ihn zu entfernen, klicken Sie auf das Icon 面.

7.1.2.10. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie für diese Aufgabe an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

7.2. Aufgaben aus Testfällen

Auf dieser Seite können Aufgaben zur Ausführung oder Überprüfung ausgewählter Testfälle erzeugt werden.

= 👱 κιαr	OS TEST MA	A N A G E M E N T	DE Finanz-Tracker 🛛 🔤		२ ४ 0 ४ ≗ ४
📝 Definieren	Aufgaben aus Test	fällen			
📇 Planen		24 Einhänn Colle Lung 2 M 🛋 🧎	Ausführung	g vorsehen	Review vorsehen
	ID A Bevision	Name	Verfoldbarkeit ੈ	Rriorität ≜ Status 1	Ausführung 🖨 Schritte 🚖
🌣 Ausführen	TC00024_1.0	Übernrüfen Sie die Verbindung zum Server	Requirement 1 2 4 in testolan 16 txt	Hoch Entwurf	Automatisiert 0
	TC00023 1.0	Einen Dauerauftrag ändern.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 5
욙 Auswerten	TC00022 1.0	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 6
🖌 Konfigurieren	TC00021 1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 6
	TC00020 1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 6
	TC00019 1.0	Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/testing/web/Conrection.java	Hoch Entwurf	Automatisiert 0
	TC00018 1.0	Testen der Verbindung zum Server der Bank.	/FinanceTracker/src/test/java/de/sabrina/testing/web/Conrection.java	Hoch Entwurf	Automatisiert 0
	TC00017 1.0	Lastschriftmandat löschen.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 5
	TC00016 1.0	Daueraufträge löschen.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 5
	TC00015 1.0	Fügen Sie einen Dauerauftrag hinzu.	Requirement 1.2.4 in testplan_16.txt	Mittel Entwurf	Manuell 5
	ID Revision	n Name	Verfolgbarkeit	Priorität Status	Ausführung Schritte
			Ausführung	g vorsehen	Review vorsehen

Abbildung 7.9. Die Seite "Aufgaben aus Testfällen"

Die Aufgaben für die ausgewählten Testfälle können mit Klick auf den Button Ausführung vorsehen oder Review vorsehen erzeugt werden. Es öffnet sich ein Dialog, in dem eine Beschreibung angegeben werden kann. Außerdem kann ausgewählt werden, wem diese Aufgabe zugewiesen wird und Angaben zur Priorität, dem zu verwendenden Testsystem, der Testumgebung und dem Start- und Fälligkeitsdatum gemacht werden.

≡	🞐 KLAR	0				ENT				DE Finanz-Tracker				Q X (∂ ~ ≗ ~
ľ	Definieren	1	Aufgaben	aus Tes	tfällen										
-20	Planen	١.	24	we wählt		24 Einträge - Se	ita 1 yoo 2	4 1 2	3 N 10	~	Ausführung	vorsehen	∇	Review vorse	hen
		1.6		Revision		Name 🖨	Re I VOIT 5 PA		Verfolgha	rkeit ≎		Priorität 🖨	รtatus ≜	Ausführung 🖨	▲ Main Annual Annua
*	Ausführen		TC0002	4 1.0	Überprüfen Sie	die Verbindung zum Server		Requireme	ent 1.2.4 in testplan_1	6.txt		Hoch	Entwurf	Automatisiert	0
		1	TC0002	3 1.0	, Einen Dauerauf	trag ändern.		Requireme	nt 1.2.4 in testplan_1	6.txt		Mittel	Entwurf	Manuell	5
Ģ		ł	✓ TC0002	2 1.0	Aktualisieren Si Dauerauftrags.	ie das Dashboard durch Hinzu	ıfügen eines	Requireme	nt 1.2.4 in testplan_1	6.txt		Mittel	Entwurf	Manuell	6
ىر	Konfigurieren	ł	✓ TC0002	1 1.0	Aktualisieren S Dauerauftrags	Ausführung als Aufgabe vors	ehen	Description	ant 1 0 4 in Acada in a 1	C 44		e dina al	Cotourf	Manuell	6
		ł	✓ TC0002	0 1.0	Aktualisieren S Lastschriftma	Beschreibung							urf	Manuell	6
		ł	✓ TC0001	9 1.0	Testen der Ver								urf	Automatisiert	0
		ł	✓ TC0001	8 1.0	Testen der Ver	Zugewiesen	Selen Deutsch		~				urf	Automatisiert	0
		ł	тсооо1	7 1.0	Lastschriftma	Priorität	Hoch 🗸						urf	Manuell	5
		ł	✓ TC0001	6 1.0	Daueraufträge	Testsystem		×					urf	Manuell	5
		ł	TC0001	5 1.0	Fügen Sie eine	,							urf	Manuell	5
			ID	Revisior	۱ –	Testumgebung		~					tus	Ausführung	Schritte
						Start	ŧ	8						Review vorse	hen
						Fällig	ŧ	ŧ.							
						Übergeordnete Aufgabe							~		
											ок	Abbrecher	•		

Abbildung 7.10. Der "Ausführung als Aufgabe vorsehen" Dialog

Planen

≡	学 K L A R	0 S T	est m	ANAGEM	ENT					DE Finar	nz-Tracker 🖬 🚆			Q × (∂× ≛×
ľ	Definieren	Aufgab	en aus Te	stfällen											
	Planen								•		Ausführung	vorsehen		Review vorse	hen
		24 von 24	ausgewählt		24	Einträge - Seite 1 von 3	M	4 1 2 3	► H	10 🗸		1	47	Q	× & ≡
D	Ausführen	ID 🗹	Revision	in	Name	¢			Ve	folgbarkeit 🗢		Priorität 4	🗧 Status 🖨	Ausführung 🖨	Schritte 🗘
		TC00	024 1.0	Überprüfen Sie	die Verbindung zur	n Server		Requirement '	1.2.4 in tes	plan_16.txt		Hoch	Entwurf	Automatisiert	0
	Auswerten	✓ TC00	023 1.0	Einen Dauerau	ftrag ändern.			Requirement *	1.2.4 in tes	plan_16.txt		Mittel	Entwurf	Manuell	5
	Auswerten	✓ TC00	022 1.0	Aktualisieren S Dauerauftrags	ie das Dashboard d	Requirement '	1.2.4 in tes	plan_16.txt		Mittel	Entwurf	Manuell	6		
ېر	Konfigurieren	🖌 тсос	021 1.0	Aktualisieren S Dauerauftrags	ie das Dashboard d	urch Stornierung eir	nes	Requirement '	1.2.4 in tes	tplan_16.txt		Mittel	Entwurf	Manuell	6
		✓ тсос	020 1.0	Aktualisieren S Lastschriftma	Beschreibung	llegen						Mittel	Entwurf	Manuell	6
		✓ тсоо	019 1.0	Testen der Ver	5						esting/web/Con	Hoch	Entwurf	Automatisiert	0
		🖌 тсос	018 1.0	Testen der Ver							esting/web/Con	Hoch	Entwurf	Automatisiert	0
		🖌 тсос	017 1.0	Lastschriftma								Mittel	Entwurf	Manuell	5
		🖌 тсос	016 1.0	Daueraufträge	Zugewiesen	Selen Deutsch			~			Mittel	Entwurf	Manuell	5
		🖌 тсос	015 1.0	Fügen Sie eine	Priorität	Hoch 🗸						Mittel	Entwurf	Manuell	5
		10) Revisio	in	01-11							Priorität	Status	Ausführung	Schritte
					Start						Auoführung	vorashan		Poviow vorac	hon
					Fallig						Austuniung	vorsenen		Review voi se	ileii
					Testrali sperren	•									
								ок		Abbrechen					

Abbildung 7.11. Der "Review-Aufgabe anlegen" Dialog

Sollen die Aufgaben einer anderen Aufgabe untergeordnet werden, kann diese hier per Dropdown-Menü ausgewählt werden.

Die neu erzeugten Aufgaben erscheinen auf den Seiten Aufgaben verwalten und Meine Aufgaben und können dort ausgeführt werden.

7.3. Aufgaben aus Testsuiten

Auf dieser Seite kann der Benutzer Aufgaben zur Ausführung oder Überprüfung ausgewählter Testsuiten generieren.

≡	👱 KLAR	OS TEST MA	NAGEMENT	DE Finanz-Tracker 📔 📑	५ ४ 0 ४ ≗ ४
ľ	Definieren	Aufgaben aus Test	suiten		
	Planen			Ausführung vorsehen	Review vorsehen
~	Ausführen	ID 🗘 Revision	/ Eintrage - Seite 1 von 1 N Seschreibung 🗢	ਲੱਜ ਪੇ Testsystem ≎	C × ﷺ ≡ Testfälle ♦
Ĩ	Adoramen	TS00008 1.0	Überprüfen Sie die Verbindung zum Server	Finance Tracker 2.1.0	1
¢	Auswerten	TS00006 1.0	Übersicht		7
بو	Konfigurieren	TS00004 1.0 TS00003 1.0	Verbingungen überprüfen Aktivitäten verschiedener Zeiträume anzeigen	Finance Tracker 1.0.0	2
*	Ronnganeren	TS00002 1.0	Überweisungen testen		8
		ID Revision	Dashboard aktualisieren Beschreibung	Testsystem	4 Testfälle
				Ausführung vorsehen	Review vorsehen



Die Aufgaben für die ausgewählten Testsuiten können mit Klick auf den Button Ausführung vorsehen oder Review vorsehen erzeugt werden. Es öffnet sich ein Dialog, in dem eine Beschreibung angegeben werden kann. Außerdem kann ausgewählt werden, wem diese Aufgabe zugewiesen wird und Angaben zur Priorität, dem zu verwendenden Testsystem, der Testumgebung und dem Start- und Fälligkeitsdatum gemacht werden.

≡ 🞐 K L A R	o s test management	DE Finanz-Tracker 🗃 🗮	द x Øv ≛v
📝 Definieren	Aufgaben aus Testsuiten		
😤 Planen		Ausführung vorsehen	Review vorsehen
	7 von 7 ausgewählt 7 Einträge - Seite 1 von 1 🕅 ┥ 1 🕨 🕅 10 🗸	· * **	Q × & ≡
📩 Δusführen	✓ ID	Testsystem 🗢	Testfälle 🗢
- Adoramen	✓ TS00008 1.0 Überprüfen Sie die Verbindung zum Server	Finance Tracker 2.1.0	1
	✓ TS00007 1.0 Daueraufträge bearbeiten		3
C Auswerten	TS00006 1.0 Übersicht		7
	✓ TS00004 1.0 Ausführung als Aufgabe vorsehen	r 1.0.0	2
🔑 Konfigurieren	TS00003 1.0 Beschreibung		4
	▼ TS00002 1.0		8
		Testousters	4 Testfälle
	Zugewiesen Selen Deutsch	Testsystem	restraile
	Priorität Hoch 🗸	orsehen	Review vorsehen
	Testsystem		
	Testumgebung		
	Start		
	Fällig		
	Übergeordnete Aufgabe	~	
	Eine Aufgabe pro Testfall		
		OK Abbrechen	

Abbildung 7.13. Der "Ausführung als Aufgabe vorsehen" Dialog

≡	🞐 KLAR	OS TEST MA				DE Finanz-Tracker 🖀 📑	द × 0° ≛°
ľ	Definieren	Aufgaben aus Tes	tsuiten				
	Planen					Ausführung vorsehen	Review vorsehen
		7 von 7 ausgewählt ✓ ID ♦ Revision	n	7 Einträge - Seite 1 von 1		品V Testsystem	Q × ಔ ≡
•	Ausführen	✓ TS00008 1.0	Überprüfen Sie die Verbindung z	um Server		Finance Tracker 2.1.0	1
		✓ TS00007 1.0	Daueraufträge bearbeiten				3
Ģ		✓ TS00006 1.0	Übersicht				7
		✓ TS00004 1.0	Verbingungen überprüfen			Finance Tracker 1.0.0	2
¥	Konfigurieren	✓ TS00003 1.0	Aktivitäten verschiedener Zeiträ	ume anzeigen			4
		✓ TS00002 1.0	Überweisunge Review-Aufgabe	anlegen			8
		✓ TS00001 1.0	Dashboard akt				4
		ID Revisior	n Beschleibung			Testsysten	n Testfälle
						Ausführung vorsehen	Review vorsehen
			Zuaewiesen	Selen Deutsch	~		
			5				
			Priorität	Hoch 🗸			
			Start	曲			
			Fällig	曲			
					OK Abbre	chen	
-							

Abbildung 7.14. Der "Review-Aufgabe anlegen" Dialog

Mit der Auswahl *Eine Aufgabe pro Testfall* kann festgelegt werden, ob für jeden Testfall der Testsuite eine eigene Aufgabe erstellt werden soll. Diese werden dann unter einer Oberaufgabe zusammengefasst. Die Aufspaltung in einzelne Aufgaben ermöglicht es, die Durchführung der Testfälle individuell auf mehrere Benutzer zu verteilen.

Sollen die Aufgaben einer anderen Aufgabe untergeordnet werden, so kann diese hier per Dropdown-Menü ausgewählt werden.

Die neu erzeugten Aufgaben erscheinen auf den Seiten Aufgaben verwalten und Meine Aufgaben und können dort ausgeführt werden.

7.4. Aufgaben nach Benutzer

Dieser Abschnitt dient dazu, die Arbeitsauslastung und den Fortschritt der einzelnen Benutzer im ausgewähltem Projekt zu verfolgen.

Auf der Seite Aufgaben nach Benutzer wird die Arbeitsauslastung, die Arbeitsdauer und der Fortschritt jedes einzelnen Benutzers für das aktuell ausgewählte Projekt angezeigt. Durch Klicken auf das Icon Q wird die Detailseite (<u>Abbildung 7.16</u>) für den entsprechenden Benutzer angezeigt.

= 👱 K L A R (o s test managem	ENT				DE Finanz-Tracker			Q X 🕼	× ≛×
🕑 Definieren	Aufgaben nach Benutzer									
🐣 Planen		Name 🗘	19 Einträge - Seite 1 von 2	M	1 2 ▶ Aufgaben	▶ 10 ✔ Geschätzte Dauer ≑	Beschäftigt von 🕯	Beschäftigt bis 🗘	Fortschrit	7 & ≡ t ♦ Aktion
🏟 Ausführen	Felix Mustermann Claudia Könnnecke Max Mustermann				3 0	00:30:00	01.08.20, 07:00	26.10.20, 00:00	1/3 0/0	Q
🕒 Auswerten	Markus Meyer Oliver Krams				40 0	5 Tage 13:52:00	01.08.20, 07:00	26.10.20, 00:00	31/40 0/0	Q
🔑 Konfigurieren	Patrick Reilly Sabrina Gidley Sahra Berger				0 18 0	18:00:00	01.08.20, 07:00	26.10.20, 00:00	0/0 18/18 0/0	Q
	Sandra Selen Selen Deutsch	Name			1 0 Aufgaben	Geschätzte Dauer	01.08.20, 07:00 Beschäftigt von	26.10.20, 00:00 Beschäftigt bis	0/1 0/0 Fortschr	Q itt Aktion
							, second s			

Abbildung 7.15. Die Seite "Aufgaben nach Benutzer"

7.4.1. Aufgaben nach Benutzer - Details

Diese Seite zeigt die Arbeitsauslastung, die Arbeitsdauer und den Fortschritt für einen Benutzer im aktuell ausgewählten Projekt an. Die Aufgaben des Benutzers können entweder in einer Tabelle aufgelistet angezeigt werden oder in einer Kalenderansicht visualisiert werden. Mit Klick auf das Icon
wird die Detailseite für die entsprechende Aufgabe geöffnet. Weitere Informationen finden Sie in <u>Abschnitt 7.1.2, "Detailseite"</u>.

Planen

≡ 📌 K L A R	os test	MANAGE	EMENT			DE Finanz-Tracker		۹ 🚱	∕ ≜ ×
📝 Definieren	Aufgaben nach	Benutzer: Ma	rkus Meyer						8
😤 Planen	Aufgaben Geschätzte Dauer Beschäftigt von	40 5 Tage 13:52:00 01.08.2020			Ne Wieder eröffne	u 🖌 et 🖌			
🏟 Ausführen	Beschäftigt bis Fortschritt	12.05.2021 31 / 40			Erledig Geschlosse				
e Auswerten	Terminnlan 💧	ufashen			Abgelehr	it			
🔑 Konfigurieren	Zurück Weite	er Aktuelles D	latum		April 2021		Monat	Woche Tag	Monat
	Мо		Di	Мі	Do	Fr	Sa	So	
	KW13	29	30	31	1	2	3		4
	KW14	5	6	7	8	9	10		11
	KW15	12	13	14	15	16	17		18
				00 Uhr JOB00359 - Ausführung	TC00021 - Aktualisierung des Da	shboards durch Stornierung eines	Dauerauftrags.		
	KW16	19	20	21	22	23	24		25
	JOB00359 - Ausfüh	rung TC00021 - Aktu	alisierung des Dashboards	durch Stornierung eines Dauerau	itrags.				
	KW17	26	27	28	29	30	1		2
	0C JOB00359 - Ausf 00 Uhr JOB00360	führung TCO() - Ausführung TCOOO	22 - Aktualisierung des Das	hboards durch Hinzufügen eines	Dauerauftrags.	00 Uhr JOB00356 - Ausführung	TC00016 - Daueraufträge löscher	1.	
	KW19	-							
	JOB00356 - Ausfüh	3 rung TC00016 - Dau	4 eraufträge löschen.	5	6	7	8		9

Abbildung 7.16. Die Detailseite "Aufgaben nach Benutzer"

Kapitel 8. Ausführen

In diesem Kapitel wird beschrieben, wie Testfälle, Testsuiten und Aufgaben ausgeführt, unterbrochene Testausführungen wieder aufgenommen und Issues für fehlgeschlagene Tests erzeugt werden.

8.1. Meine Aufgaben



Auf der Seite *Meine Aufgaben* können die Aufgaben des aktiven Benutzers sowie alle nicht zugewiesenen Aufgaben ausgeführt werden. (<u>Abbildung 8.1</u>).

≡ 🧏	≡ 📌 KLAROS TEST MANAGEMENT									DE Fir	anz-Tracker 🔲		Q 🚱 Y	* ~
📝 Def	finieren	Meine Aufgal	ben											
🚢 Pla	anen	Projekt ੈ	ID 🖨	3 Eint	räge - Seite 1 von 1		1	M 20 V]	Priorität 🚔	Status 🚔	Alle Projekte	Q X	Aktion
🌼 Aus	sführen	DE Finanz- Tracker	JOB00241	Test editing standir	ng orders	lassung 🗸			Finance Tracker 2.1.0	▲ Hoch	Wieder eröffnet	0%	26.10.2020	
		DE Finanz- Tracker	JOB00357	Ausführung TC000	17 - Lastschriftma	ndat lösche	n.		Finance Tracker 2.0.0	O Blocker	Neu	0 %	26.10.2020	2 🎝
🕒 🐣 Au	swerten	DE Finanz- Tracker	JOB00358	Ausführung TC000 Stornierung einer B	20 - Aktualisierung elastungsermäch	i des Dashb igung.	oards durc	h	Finance Tracker 2.0.0	S Blocker	Neu	0 %	26.10.2020	2 🎝
🗲 Kor	nfigurieren	Aufgaben o	hne Veran	twortlichen										
					1 Einträge - Seite	1 von 1 🛛 🖌	I 🔺 📘	► H	20 🗸			V	Q X	≗≡
		Pro	jekt 🖨	ID 🗘	Zu	sammenfas	sung 🖨		Testsystem 🕏	Priorität 🖨	Status 🖨	Fortschritt 🗘 Erfolg 🖨	Fällig 🖨	Aktion
		DE Finanz-Track	er	JOB00284	Test the dashboar	d			Finance Tracker 2.0.1	 Hoch 	Wieder eröffnet	100%	26.10.2020	2

Abbildung 8.1. Die Seite "Meine Aufgaben"

Die Tabelle Meine Aufgaben and Aufgaben ohne Verantwortlichen zeigen die folgenden Werte:

Projekt	Das Projekt, zu dem die Aufgabe gehört.
ID	Die automatisch zugewiesene ID.
Zusammenfassung	Eine Zusammenfassung der Aufgabe.
Testsystem	Das dieser Aufgabe zugeordnete Testsystem.
Priorität	Die Priorität der Aufgabe. Mögliche Werte sind <i>Trivial, Niedrig, Hoch, Kritisch</i> und <i>Blocker</i> .
Status	Der Status der Aufgabe. Mögliche Werte sind Neu, Wieder eröffnet, In Arbeit, Erledigt, Geschlossen und Abgelehnt.
Fortschritt	Der Prozentsatz der ausgeführten Testfälle dieser Aufgabe und ihrer Unteraufgaben.
Erfolg	Die Erfolgsrate dieser Aufgabe in Prozent.

Aktion

Die ausführbaren Aktionen.

Das Datum, bis zu dem diese Aufgabe beendet sein soll.

8.1.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in den Tabellen. Folgende Aktionen lassen sich hier vornehmen:

- 🕜 Bearbeiten
- Ausführen
- Le Manuell ausführen
- Lo Neueröffnen oder aktualisieren
- ▲ Review ausführen
- Ergebnisse importieren

8.1.2. Tabellenoperationen

In der Zeile rechts oberhalb der Tabellen lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Alle Projekte / Nur aktuelles Projekt

- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

8.1.3. Eine Aufgabe ausführen

Die zur Verfügung stehende Aktion für jede Aufgabe in den Tabellen richtet sich nach dem Typ der Aufgabe, ob sie manuelle oder automatisierte Testfälle enthält und ob sie Abhängigkeiten zu anderen Aufgaben hat.



Nicht ausführbare Aufgaben

Falls eine Aufgabe nicht ausgeführt werden kann, ist das entsprechende Icon in der Aktionen-Spalte ausgegraut. Befindet sich der Mauszeiger über dem Icon, wird die Begründung als Tooltip angezeigt.

Ausführen

Die Aufgabe ist ein Auftrag und sie enthält keinen Testfall und keine Testsuite.
La Manuell ausführen	Die Aufgabe enthält mindestens einen manuell ausführbaren Testfall.
Ausführen oder aktualisie- ren	Die Aufgabe wurde bereits abgeschlossen.
▲ Review durchführen	Die Aufgabe ist eine Review-Aufgabe.
-🔄 Ergebnisse importieren	Die Aufgabe enthält nur automatisiert ausführbare Testfälle.

8.1.3.1. Ein Review durchführen

Aufgaben vom Typ *Review* sind dazu da, Testfälle oder Testsuiten zu überprüfen. Mit Klick auf den Button Review abschließen öffnet sich ein Dialog, in dem der Testfall bzw. die Testsuite freigegeben oder abgelehnt werden kann.

8.2. Testfall ausführen

Das Ausführen eines Testfalls bedeutet entweder, die angegebenen Testschritte mit einer bestimmten Version des Testsystems in einer ausgewählten Testumgebung nacheinander auszuführen (nur bei Testfällen mit mindestens einem Schritt) oder die zugehörigen Testergebnis-Dateien zu importieren.

≡	👱 KLAR	0 S T	EST	MAN	AGEMENT		DE Finanz-Tracker 🖬 🗮		۹	× 0×	* ~
ľ	Definieren	Testfäl	le ausf	ühren							
:2:	Planen				24 Einträge - Seite 1 von 1	∢ 💶 ► →	30 🗸	# 7		QX	& ≡
		ID 🖨									Aktion
	A	TC00024	L 🗛	1.0	Überprüfen Sie die Verbindung zum Server	Automatisiert	Requirement 1.2.4 in testplan_16.txt	0	2/1	3	-5
*	Austunren	TC00023	. 📀	1.0	Einen Dauerauftrag ändern.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/2	31	20
e	Auswerten	TC00022	2 ×	1.0	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Manuell	Requirement 1.2.4 in testplan_16.txt	6	5/0	25	2 0
		TC00021	0	1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Manuell	Requirement 1.2.4 in testplan_16.txt	6	0/2	24	2 ¢
ىر	Konfigurieren	TC00020) 🛪	1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Manuell	Requirement 1.2.4 in testplan_16.txt	6	0/0	19	.
		TC00019	0 8	1.0	Testen der Verbindung zur Datenbank.	Automatisiert	/FinanceTracker/src/test/java/de/sabrina/testing/w eb/Connection.java	0	0/0	3	-5
		TC00018	8 🕄	1.0	Testen der Verbindung zum Server der Bank.	Automatisiert	/FinanceTracker/src/test/java/de/sabrina/testing/w eb/Connection.java	0	0/0	3	-5
		TC00017	7 🔺	1.0	Lastschriftmandat löschen.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/0	31	20
		TC00016	5 📀	1.0	Daueraufträge löschen.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/0	41	20
		TC00015	5 📀	1.0	Fügen Sie einen Dauerauftrag hinzu.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/0	49	20
		TC00014	4	1.0	Aktualisierung des Dashboards nach einer Banküberweisung.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/0	14	2 ¢
		TC00013	3 🛛 🔿	1.0	Überweisen Sie Geld ohne Angabe einer Referenz.	Manuell	Requirement 1.2.4 in testplan_16.txt	5	0/0	46	20
		TC00012	2 📀	1.0	Überweisen Sie Geld, ohne einen Betrag anzugeben.	Manuell	Requirement 1.2.4 in testplan_16.txt	4	0/0	44	20
		TC00011	•	1.0	Überweisen Sie Geld ohne Angabe der IBAN.	Manuell	Requirement 1.2.4 in testplan_16.txt	4	0/0	31	20
		TC00010) 🔺	1.0	Überweisen Sie Geld, ohne einen Empfänger anzugeben.	Manuell	Requirement 1.2.4 in testplan_16.txt	4	0/0	38	.
		TC00009) 🔺	1.0	Zeigen Sie die Aktivitäten der letzten drei Monate an.	Manuell	Requirement 1.2.4 in testplan_16.txt	3	0/0	24	2 0
		TC00008	8 🔺	1.0	Aktivitäten innerhalb eines vom Benutzer	Manuell	Requirement 1.2.4 in testplan_16.txt	4	0/0	26	20

Abbildung 8.2. Die Seite "Testfälle ausführen"

Die Seite Testfälle ausführen zeigt alle Testfälle des ausgewählten Projektes in einer Tabelle an.

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision des Testfalles.
Name	Der Name des Testfalls.

Ausführung	Manuell oder automatisiert.
Verfolgbarkeit	Die Anforderung, die diesem Testfall zugeordnet ist.
Schritte	Die Anzahl der definierten Testschritte für diesen Testfall.
Ergebnisse	Die Anzahl der Testfallergebnisse.
Aktion	Die ausführbaren Aktionen.

8.2.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:





8.2.2. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 옮 Kategorisieren
- √ Filter / Sortieren
- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

8.2.3. Einen Testfall ausführen



Manuelle oder automatisierte Ausführung

Wenn ein Testfall mindestens einen Testschritt hat, kann dieser manuell ausgeführt werden. Alternativ können für alle Testfälle Testergebnis-Dateien importiert werden (weitere Informationen hierzu finden Sie unter <u>Abschnitt 8.5, "Testergebnisse importieren"</u>.



Anmerkung

Nur Testfälle mit dem Status Entwurf oder Genehmigt können ausgeführt werden.

Mit Klick auf das Icon and wird ein Dialog mit Informationen zum Testfall angezeigt. (<u>Abbildung</u> <u>8.3</u>).

≡	🞐 KLAR	os test ma	NAGEMENT	DE Finanz-Tracker 🛛 📑		५ × 0∕	* ~
ľ	Definieren	Testfälle ausführen					
_							
***	Planen	Testfall ausführen				QX	& ≡
		1.0 - TC00021, A	tualisieren Sie das Dashboard durch Stornierung eines Dauerau	uftrags.	▲ Jes ♥	Ergebnisse 🗢	Aktion
*	Ausführen	— Testfalldetails			/2	31	20
		restrandetano	T000001		/0	25	.
Ģ	Auswerten	IL	10		/2	24	С
		Name	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags				
يكر	Konfigurieren	Beschreibung	Testet die automatische Aktualisierung des Dashboards durch Stonierung eines	s Dauerauftrags.	/0	19	
					/0	3	-5
					10	2	F 2
						3	-12
		Vorbedingung	Stellen Sie sicher, dass die Anwendung vor dem Testen installiert ist!		/0	31	20
					/0	41	20
			Die Anwendung ist auf dem Gerät installiert. Falls nicht folgen Sie den A	nweisungen: https://finanztracker/install	/0	49	
			Vorbindung zum Sonvor bestaht	anversungen. https://infanztracker/instan	/0	14	2 0
			• Verbindung zum server besteht.		/0	46	20
					/0	44	20
					/0	31	20
					/0	38	20
		Nachbedingung	Das Dashboard wurde aktualisiert.		(0)	04	•.
					70	24	
					/0	26	20
		Erwartetes Ergebnis	Das Dashboard hat einen Eintrag weniger für Daueraufträge als beim Starten de	ler Anwendung.	10	22	•.
			5555	5	/0	23	
					/0	23	
					/0	24	2.
		Team	Team Alpha		/0	17	20
		Status	Entwurf		/0	14	20
		Testar	Funktional		/0	14	20
		Entwurfsverfahren	Black-Box		sues		Aktion
		Ergebnisar	Positiv		-		
				Ausführen Abbre	chen		

Abbildung 8.3. Der Dialog "Testfall ausführen"

Bevor ein Testfall ausgeführt werden kann, müssen die Testumgebung, in der der Testfall ausgeführt wird und das Testsystem selbst ausgewählt werden. Außerdem können hier die benutzerdefinierten Eigenschaften des Testlaufs angegeben werden (weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>).

Mit Klick auf den Button Ausführen öffnet sich ein zweites Browserfenster, in dem die Testausführung stattfindet. Stellen Sie sicher, dass Pop-Ups in Ihrem Browser erlaubt sind oder fügen Sie eine Ausnahme für Klaros-Testmanagement hinzu. Standardmäßig wird eine Ansicht mit einer Schritt-für-Schritt-Anleitung angezeigt, dies kann aber während der Ausführung in eine tabellarische Ansicht geändert werden.

8.2.3.1. Die Ansicht "Schritt-für-Schritt-Anleitung"



Abbildung 8.4. Die Ansicht "Schritt-für-Schritt-Anleitung"

Die Ansicht Schritt-für-Schritt zeigt die folgenden Werte für den aktuellen Testschritt an: Aktion, Erwartetes Ergebnis, Vorbedingung und Nachbedingung.

Abhängig vom Ergebnis können folgende Werte vergeben werden:

\oslash	Bestanden	Der Testschritt wurde erfolgreich abgeschlossen.
⊗	Fehler	Der Testschritt wurde mit einem Fehler abgeschlossen.
⚠	Fehlschlag	Der Testschritt wurde mit einem Fehlschlag abgeschlossen.
*	Unklar	Der Testschritt wurde mit keinem eindeutigen Ergebnis abge- schlossen.
M	Überspringen	Den aktuellen Testschritt überspringen, ohne das Ergebnis des Testfalls zu verändern.
M	Alle Überspringen	Alle Testschritte überspringen, ohne das Ergebnis des Test- falls zu verändern.
ľ	Bearbeiten	Dieses Testschrittergebnis bearbeiten.
۲	Review	Ein Review für diesen Testfall anfordern.
0	Einen Issue erstellen	Einen Issue zu diesem Testfall in einem Issue-Manage- ment-System erstellen.
()	Einen Issue verknüpfen	Einen bestehenden Issue aus einem Issue-Management-System mit diesem Testfall verknüpfen.

G Zurück

Zurück zum letzten Testschritt, um ihn zu wiederholen oder zu bearbeiten.

Das gleiche Vorgehen wird für jeden Testschritt des Testfalls angewendet.

Sobald alle Schritt ein Ergebnis haben oder übersprungen wurden, kann über einen Dialog die Ausführung beendet werden.

Mit Klick auf den Button OK wird das Ergebnis des Testlaufs für den Testfall angezeigt (<u>Abbil-dung 8.7</u>).

8.2.3.2. Die Ansicht "Tabellarische Ansicht"



Abbildung 8.5. Die Ansicht "Tabellarische Ansicht"

Die Seite Tabellarische Ansicht zeigt die Beschreibung, das Erwartetes Ergebnis, die Vorbedingung, die Nachbedingung und das Ergebnis aller Schritte an. Das Standardergebnis ist Unbekannt.

Die folgenden Aktionen können bei jedem Schritt durchgeführt werden:

M	Überspringen	Den aktuellen Testschritt überspringen, ohne das Ergebnis des Testfalls zu verändern.
G	Zurück	Zum letzten Testschritt zurückgehen, um ihn erneut auszufüh- ren oder zu bearbeiten.
۲	Review	Fordert ein Review für diesen Testfall an.
0	Einen Issue erstellen	Erstellt einen Issue zu diesem Testfall in einem Issue-Manage- ment-System.

() Einen Issue verknüpfen	Verlinkt einen bestehenden Issue aus einem Issue-Manage- ment-System mit diesem Testfall.
Testausführung anhalten	Pausiert die Testausführung und stoppt die Zeitmessung.
Alle verbleibenden Schritte überspringen	Beendet den Test und markiert alle verbleibenden Schritte als übersprungen.



Was ist der Unterschied zwischen einem Fehlschlag und einem Fehler?

Ein Fehler ist eine Diskrepanz zwischen einem berechneten, beobachteten oder gemessenen Wert oder Zustand und dem wahren, spezifizierten oder theoretisch richtigen Wert oder Zustand.

Ein Fehler ist die Unfähigkeit des Systems, den Test korrekt durchzuführen.

Für jeden Fehler oder Fehlschlag sowie für jedes unklare Ergebnis wird der folgende Dialog angezeigt: (<u>Abbildung 8.6</u>).

Tabellena + TC00	nsicht?	das Dashboard durch Stornierung eines Dauerauftrags. Schritt 1 von 6	
	Schritt 1 von 6 Ergebnis 🔞		
Vorb	https://finan	ung ist auf dem Gerät installiert . Wenn nicht, folgen Sie den Anweisungen: iztracker/install	
	Das Ergebnis des Schr	ritts bearbeiten	
	Ergebnis	😢 🗸 Fehler	
	Zusammenfassung	App ist nicht auf dem Gerät installiert.	
	Beschreibung	Fügen Sie dem Testfall einen Hinweis hinzu, um sicherzustellen, dass die App vor dem Testen auf dem Gerät installiert ist.	
Erwartetes			
	1 Anhänge hochla	aden OK Abbrechen	
Nachb	edingung Die Anwende	ung ist erfolgreich gestartet .	

Abbildung 8.6. Der Dialog "Testschrittergebnis bearbeiten"

- Anhang hochladen öffnet einen Dialog zum Hochladen von Anhängen an den Schritt.
- Test beenden öffnet einen Dialog, der nachfragt, ob die Ausführung beendet werden soll.
- Ok schließt den Dialog und zeigt den nächsten Schritt an.
- Abbrechen schließt den Dialog.



Tipp

Die Felder Zusammenfassung und Beschreibung erscheinen in allen Kommentar-Dialogen.

8.2.3.2.1. Ausführung beenden

Nachdem das letzte Schritt-Ergebnis eines Testfalls eingegeben wurde, wird ein Dialog angezeigt der nachfragt, ob die Ausführung beendet werden soll.

Mit Klick auf den Button OK wird das Ergebnis des Testlaufs für diesen Testfall angezeigt (<u>Abbildung 8.7</u>).

In der Ansicht *Testlauf* - *Übersicht* können eine Zusammenfassung und eine Beschreibung für das Testfallergebnis sowie für jedes einzelne Testschrittergebnis hinterlegt werden. (<u>Abbildung 8.7</u>).

Testlauf - Übers	icht
Testfallergebnis Testlauf Start Ausführungsdauer	TCR0007369 TRU0005309 30.07.21, 11:55 00:00:12
Zusammenfassung	
Beschreibung	
Schritte	Anhang hochladen
Ergebnisse	1 2 3 4 5 6
Schritt	Schritt 1: Starten Sie die Anwendung durch Drücken der blaue 💙
Zusammenfassung	
Beschreibung	

Abbildung 8.7. Die Seite "Testlauf - Übersicht"



Anmerkung

Sofern für das aktive Projekt ein Issue-Management-System konfiguriert ist, werden die Buttons Issue verlinken und Issue anlegen aktiviert.

8.2.3.3. Issues erstellen

Während der Ausführung eines Testfalls können für diesen Testfall mit einem Klick auf den Button Issues in einem externen Issue-Management-System erstellt werden. Bereits angelegte Issues aus einem Issue-Management-System können mit einem Klick auf den Button ① mit dem Testfall verlinkt werden. Ausführliche Informationen zum Anlegen und Verlinken von Issues finden Sie unter Abschnitt 9.6.6, "Issue - Details" und Abschnitt 9.6.7, "Einen Issue verlinken".

8.2.3.4. Review-Aufgaben anlegen

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Wird während der Testausführung ein Fehler in einer Testfallbeschreibung gefunden, kann durch Klick auf den Button ④ eine Überprüfungsaufgabe angelegt werden. Es öffnet sich ein Dialogfenster, in dem die Details der Review-Aufgabe eingegeben werden können:

Tabellenansicht? Image: Comparison of the system of th								
	Review-Aufgabe ar	legen						
Frwartetes	Beschreibung Zugewiesen Priorität Start	Der Testfall ist veraltet. Felix Mustermann						
Liwartetea	Fällig							
	Testfall sperren							
Nachb		OK Abbrechen						

Abbildung 8.8. Eine Review-Aufgabe anlegen

8.3. Testsuiten ausführen

Das Ausführen einer *Testsuite* besteht aus der Ausführung einer definierten Menge von Testfällen auf einer bestimmten Version des Testsystems in einer ausgewählten Testumgebung.

≡	y KLARO)S TES	ЗT	MAN	AGEMENT	DE Finanz-Tracker 🧧 📑		<u>२ × </u> @≁	* ~
ľ	Definieren	Testsuiten	n aus	sführen					
<u>.</u>	Planen	ID 🚖 🛛 0		Revision	7 Einträge - Seite 1 von 1 🔣 🚽 1 🕨 🕅 🛛		# ▼	Q X	
٠	Ausführen	TS00008 A	0	1.0 1.0	Überprüfen Sie die Verbindung zum Server Daueraufträge bearbeiten		1 3	4 21	-5i 20
¢	Auswerten	TS00006 TS00004 A TS00003	0	1.0 1.0 1.0	Übersicht Verbingungen überprüfen Aktivitäten verschiedener Zeiträume anzeigen		7 2 4	19 2 3	20 -5
r	Konfigurieren	TS00002 TS00001	0	1.0 1.0	Überweisungen testen Dashboard aktualisieren		8 4	7	20
		ID		Revision	Beschreibun	3	Testfälle	Ergebnisse	Aktion

Abbildung 8.9. Die Seite "Testsuiten ausführen"

Die Seite Testsuiten ausführen zeigt alle Testsuiten des ausgewählten Projektes in einer Tabelle.

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Revision	Die Revision der Testsuite.
Beschreibung	Die Beschreibung der Testsuite.
Testfälle	Die Anzahl der Testfälle in der Testsuite.
Ergebnisse	Die Anzahl der Testfallergebnisse.
Aktion	Die ausführbaren Aktionen.

8.3.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Le Manuell ausführen



8.3.2. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- 品 Kategorisieren
- √ Filter / Sortieren

- Q Suchen
- 요 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

8.3.3. Eine Testsuite ausführen



Wichtig

Eine Testsuite mit mindestens einem Testschritt in einem ihrer Testfälle kann manuell ausgeführt werden. Für alle anderen Testsuiten können Testergebnis-Dateien importiert werden (weitere Informationen hierzu finden Sie unter <u>Abschnitt 8.5, "Testergebnisse importieren"</u>.



Warnung

Testfälle mit dem Zustand *Gesperrt* und *Überspringen* oder ohne Schritte werden beim Ausführen der Testsuite übersprungen.

Mit Klick auf den Button 2, wird ein Dialog mit Informationen zur Testsuite angezeigt. (<u>Abbildung</u> 8.10).

≡ 👱 K L A R	o s test management	DE Finanz-Tracker 🖀 📑	Q X ØY ≛Y
📝 Definieren	Testsuiten ausführen		
📇 Planen	7 Einträge - Seite 1 von 1 K 4 3 k H 10 v	品又 Testi	
🔹 Ausführen	TS00008 ▲ © 1.0 Überprüfen Sie die Verbindung zum Server TS00007 >24 1.0 Daueraufträge bearbeiten	1	4 -
🕒 Auswerten	Testsuite ausführen 1.0 - TS00007, Daueraufträge bearbeiten	7 2 4	19 2 3 4
🔑 Konfigurieren	- Testsuitedetails	8	7 2
	ID TS00007 Revision 1.0 Beschreibung Daueraufträge bearbeiten		
	ID 🕈 Revision 🛛	Ausführung 🕈 Status 🗢 Schritte 🕈 Aktion	
	TC00015 1.0 Image: Fügen Sie einen Dauerauftrag hinzu. TC00023 1.0 Image: Einen Dauerauftrag ändern. TC00016 1.0 Daueraufträge löschen.	Manuell Entwurf 5 🌲 Manuell Entwurf 5 🌲 Manuell Entwurf 5 🌲	
	Testumgebung Android 8 Smartphone Q Testsystem Finance Tracker 2.1.0 Q	Ausführen Abbrechen	



Bevor eine Testsuite ausgeführt werden kann, müssen die Testumgebung, in der die Testsuite ausgeführt wird und das Testsystem selbst ausgewählt werden. Außerdem können hier die benut-

zerdefinierten Eigenschaften des Testlaufs angegeben werden (weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>).

Mit Klick auf den Button Ausführen öffnet sich ein zweites Browserfenster, in dem die Testausführung stattfindet. Stellen Sie sicher, dass Pop-Ups in Ihrem Browser erlaubt sind oder fügen Sie eine Ausnahme für Klaros-Testmanagement hinzu.

- TS00007 - Dau	ueraufträge b	earbeiten Testfall 1 von 3	
Testsuite T	S00007		
Name D	aueraufträge	bearbeiten	
Revision 1	.0		
Testfälle 3			
Testsystem F	inance Track	er 2.1.0	
Testumgebung A	ndroid 8 Sma	artphone	
Anhänge			
— Testfallübersi	cht		
	Testfall	TC00015	
	Revision	1.0	
	Name	Fügen Sie einen Dauerauftrag hinzu.	
Be	eschreibung	Testet die automatische Aktualisierung des Dashboards durch Hinzufügen eines n euen Dauerauftrags.	
Vo	orbedingung	Stellen Sie sicher, dass die Anwendung vor dem Testen installiert ist!	
		- Die Anwendung ist auf dem Gerät installiert. Falle nicht felgen Sie den Anw	
		eisungen: https://fipanztracker/install	
		Verbindung zum Server besteht	
Nac	hbedingung	Das Dashboard wurde aktualisiert.	
	Schritte	5	
	Testsystem	Finance Tracker 2.1.0	
Tes	tumgebung	Android 8 Smartphone	
Testing internally	at company		
- /	Anhänge		
		Beginnen Testfall überspringen Abbrechen	

Abbildung 8.11. Die Ansicht "Testfallübersicht"

Diese Ansicht zeigt die Übersicht des nächsten auszuführenden Testfalls. Mit Klick auf den Button Beginnen wird die Ausführung des Testfalls gestartet (die Ausführung eines Testfalls ist ausführlich in <u>Abschnitt 8.2.3, "Einen Testfall ausführen"</u> beschrieben).

8.3.3.1. Testfälle überspringen

Während der Ausführung einer Testsuite können einzelne Testfälle zur späteren Ausführung übersprungen oder endgültig übersprungen werden.

8.3.3.1.1. Testfälle endgültig überspringen

Tabellenansicht?	alisieren Sie das Dashboard durc	n Stornierung eines [Dauerauftrags.	Schritt 1	Von 6
Schritt Ergebnis Vorbedingung Aktion	1 von 6 Die Anwendung ist auf dem Ger <u>s://finanztracker/install</u> Starten Sie die Anwendung dur sicher?	ät installiert . Wenn ch Drücken der blau	nicht, folgen Si en Symbolscha	ie den Anweist Ittfläche [©] .	ıngen: <u>http</u>
Nacht ?	Diese Aktion beendet die Ausführu fortfahren?	ng des aktuellen Tes	stfalls. Wollen S	Sie Abbrechen	
	° 🖉 👁 🕕 📜		ę		8 %

Abbildung 8.12. Der Dialog "Einen Testfall dauerhaft überspringen"

Nach dem endgültigen Überspringen eines Testfalls wird der Tester aufgefordert, eine Begründung einzugeben, warum der Testfall übersprungen wurde. Sind bereits Begründungsvorlagen definiert (siehe <u>Abschnitt 10.4.3, "Testausführung"</u>), kann eine dieser Vorlagen ausgewählt werden.

Mit Klick auf das Icon + wird die ausgewählte Vorlage in die Zusammenfassung des Testfallergebnisses kopiert.

Tabellenansicht?	das Dashboard durch Stornieru	ung eines Dauerauft	rags. Schritt	වූ 1 von 6
Vorbedingung Die Anwend https://final	lung ist auf dem Gerät installie nztracker/install	rt . Wenn nicht, folg	en Sie den Anweisun	gen:
Endgültig überspringe	en			
Begründen,	warum dieser Testfall vollständ	lig übersprungen wi	ird.	
Zusammenfassung	Test Server ist nicht erreichb	ar.		
Erwartetes				
		ОК	Abbrechen	
Nachbedingung Die Anwend	lung ist erfolgreich gestartet .			

Abbildung 8.13. Der Dialog "Eine Begründung einfügen"



Anmerkung

Wenn alle Schritte eines Testfalls überspringen wurden, wird der gesamte Testfall als dauerhaft übersprungen markiert.

Testlauf - Übers	icht
Testfallergebnis	TCR0007367
Testlauf	TRU0005307
Start	30.07.21, 11:54
Ausführungsdauer	00:00:10
Zusammenfassung	
Beschreibung	
Schritte	Anhang hochladen
oonnitte	
Ergebnisse	1 2 3 4 5 6
Schritt	Schritt 1: Starten Sie die Anwendung durch Drücken der blaue 💙
Zusammenfassung	
Beschreibung	

Abbildung 8.14. Die Ansicht "Testlauf - Übersicht"

Nach Beendigung eines Testlaufs kann der Tester auch die Begründungen für das Überspringen von Testfällen in der Testlaufübersicht bearbeiten (siehe <u>Abbildung 8.14</u>).

8.3.3.2. Ausführung beenden

Nachdem alle Testfälle ausgeführt wurden, wird die Ansicht *Testsuitelauf - Übersicht* angezeigt. Hier kann für jeden einzelnen Testfall und Testschritt des Testlaufs eine Zusammenfassung und eine Beschreibung eingegeben werden (<u>Abbildung 8.15</u>).

+ TS00007 - Dau	eraufträge bearbeiten Testfall 1 von 3
Testsuitelauf - Ü	bersicht
Testsuiteergebnis Testlauf Ausführungsdauer	TSR0000976 TRU0005314 00:00:12
Ergebnisse	1 2 3
Testfallergebnis	TC00015-TCR0007374 ¥
Testfall	TC00015 - Fügen Sie einen Dauerauftrag hinzu.
Zusammenfassung	
Beschreibung	
	Anhang hochladen
Ergebnisse	1 2 3 4 5
Schritt	Schritt 1: Starten Sie die Anwendung durch 💙
Zusammenfassung	
Beschreibung	

Abbildung 8.15. Die Ansicht "Testsuitelauf - Übersicht"

8.4. Testlauf fortsetzen

Wird ein Testlauf nicht vollständig durchgeführt und muss unterbrochen werden, kann die Ausführung später fortgesetzt werden. Die Ausführung wird mit dem ersten Testschritt, der noch kein Ergebnis besitzt, fortgesetzt.

Die Seite Ausführung fortsetzen zeigt alle noch nicht beendeten Testläufe des ausgewählten Projektes in einer Tabelle an.



Anmerkung

Die Einstellung *Tester dürfen nur eigene Testläufe wieder aufnehmen* regelt, ob Benutzer mit der Rolle *Tester* unterbrochene Testläufe von anderen Benutzern sehen und fortsetzen können (<u>Abschnitt 10.4.1, "Sonstiges"</u>).



Abbildung 8.16. Die Seite "Ausführung fortsetzen"

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Start	Der Startzeitpunkt des Testlaufs.
Ausführender	Der Name des Benutzers, der den Testlauf zuletzt ausgeführt hat.
Iteration	Die Iteration, in der der Testlauf ausgeführt wird.
Testfall/suite	Die ID des Testfalls oder der Testsuite des Testlaufs.
Testumgebung	Die Testumgebung, in der der Test ausgeführt wird.
Testsystem	Das Testsystem, das getestet wird.
Fortschritt	Die Anzahl der abgeschlossenen Testfälle.
Aktion	Die ausführbaren Aktionen.

8.4.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Q Die bisher erfassten Testfallergebnisse dieses Testlaufs anzeigen.

Die bisher erfassten Testfallergebnisse dieses Testlaufs anzeigen. Die Ausführung des Testlaufs fortsetzen

- Druckansicht öffnen
- Diesen Testlauf löschen.

Wurde ein Testlauf gelöscht, ist er zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testläufe steht dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Den Testlauf wiederherstellen.
- ⊘ Den Testlauf unwiederbringlich aus der Datenbank entfernen (nur Administrator).



Einen Testlauf löschen

Beim Löschen eines Testlaufs werden die zugehörigen Testfall- und Testsuiteergebnisse ebenfalls gelöscht.

Sind alle Testläufe für ein bestimmtes Testsystem oder eine bestimmte Testumgebung gelöscht, kann auch das jeweilige Testsystem oder die Testumgebung gelöscht werden. Andernfalls bleiben diese für das Löschen gesperrt.

8.4.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testläufe gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testläufe aus, auf die die Aktion angewendet werden soll.

Für Testläufe werden folgende Massenaktionen unterstützt:

- Die Attribute des Testlaufs editieren.
- Druckansicht öffnen
- fi Löschen
- Hereicher (nur Administrator)

☆ Die ausgewählten Testläufe unwiederbringlich aus der Datenbank entfernen (nur Administrator).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

8.4.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

Q Suchen

公 Exportieren

∃ Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

8.4.4. Testlauf fortsetzen

Mit Klick auf das Icon **L** Icon; in der Aktionsspalte gelangen Sie zu einer Übersichtsseite um den Testlauf fortzusetzen. Hier werden die Eigenschaften des Testlaufs und alle bisher ermittelten Testfallergebnisse angezeigt.



Abbildung 8.17. Die Seite "Ausführung fortsetzen"

Ein Klick auf den Button Fortsetzen nimmt die Ausführung des Testlaufs am zuletzt gespeicherten Testschritt wieder auf.

Ein Klick auf den Button Zurück führt zurück zur Seite Ausführung fortsetzen (Abbildung 8.16).

8.5. Testergebnisse importieren

In Klaros-Testmanagement können Testergebnisse aus verschiedenen Test-Frameworks importiert werden (eine vollständige Liste aller unterstützten Formate finden Sie unter <u>Abschnitt 8.5,</u> <u>"Testergebnisse importieren"</u>). Sind die zugehörigen Testfälle oder Testsuiten nicht bereits in der Klaros-Testmanagement definiert, können diese während des Imports automatisch erstellt werden.

Um den Import zu starten, muss eine XML-Datei mit Ergebnissen aus einem unterstützten Test-Framework hochgeladen werden. Wenn der Datei-Upload abgeschlossen ist, ist es zwingend erforderlich, das Testsystem und die Testumgebung auszuwählen, der diese Ergebnisse zugeordnet werden sollen. Das Format der Ergebnisdatei wird in der Regel automatisch erkannt.

≡ 👱 K L A I	ROS TEST MANAGEN		DE Finanz-Tracker 🗧 🚆	२ ४ 0 ४ ≗ ४
📝 Definieren	Ergebnisse importieren			
😤 Planen	Zum Fortsetzen muss eine Da	atei mit Testergebnissen hochgeladen werden		×
🔅 Ausführen	Ergebnisdatei hochlad	len		
	Format	Testumgebung	Testsystem	
Auswerten	AUnit	Android 8 Smartphone		
6 1 1 1 1	Boost Test	Android 9 Smartphone	Finance Tracker 2.0.1	
Kontigurieren	Check	Android Smartwatch	Finance Tracker 2.0.0	
	CppUnit	Android Tablet	Finance Tracker 1.1.1	
	ctest	Android 10 Smartphone	Finance Tracker 1.1.0	
	CUnit		Finance Tracker 1.0.0	
	EmbUnit			
	Fitnesse			
	Free Pascal Unit			
	GoogleTest			
	GLib/gtester			
	JBehave			
	JMeter			
	JsUnit			
	Jubula/GUIDancer			
	JUnit			
	MbUnit			
	MSToct			

Abbildung 8.18. Die Seite "Ergebnisse importieren"

Mit Klick auf dem Button Nächste öffnet die *Testergebnisse zu Testfällen zuordnen* Seite. Der Testfall, dem jedes Testergebnis zugeordnet werden soll, kann über die entsprechende Dropdown-Box ausgewählt werden. Jedes Ergebnis muss einem anderen Testfall zugeordnet werden.

Der anfängliche Testfallnamensabgleich wird durch den internen Namen des Testfalls oder der Testsuite in Klaros-Testmanagement gegenüber dem externen Namen des Testfalls oder der Testsuite in der Ergebnisdatei bestimmt. Diese Heuristik kann hier außer Kraft gesetzt werden. Einmal zugeordnet, wird der Abgleich eines Testfalls oder einer Testsuite mit einem externen Namen automatisch für weitere Importe angewendet, bis anders definiert. Dies gilt auch für Ergebnisimporte über REST API oder andere externe Quellen wie das Jenkins-Plugin.

Den Import einzelner Ergebnisse kann übersprungen werden, indem der Eintrag für dieses Ergebnis über die Checkbox auf der linken Seite abgewählt wird. Wird der Importprozess durch eine Aufgabe zur Testausführung angestoßen, wird die ursprüngliche Auswahl mit den in der Aufgabe enthaltenen Testfällen oder Testsuiten abgeglichen. Andernfalls werden alle Ergebnisse zunächst automatisch für den Import ausgewählt.

Ist die Option *Beim Import von Testergebnissen für unbekannte Tests automatisch neue Testfälle anlegen.* ausgewählt, wird beim Import automatisch ein neuer Testfall erstellt, wenn dieser nicht bereits vorhanden ist.

Ist die Option *Beim Import von Testergebnissen ein Testsuiteergebnis erzeugen und falls notwendig eine neue Testsuite anlegen* ausgewählt, werden automatisch auch Testsuiteergebnisse für die entsprechenden Testsuiteinformationen, die in der Ergebnisdatei enthalten sind, erstellt. Diese Informationen können je nach Importformat variieren. Zusätzlich wird eine entsprechende Testsuite für das Testsuiteergebnis erzeugt, falls es noch nicht vorhanden war.

≡ 👱 K L A R	ROS TEST MANAGEMENT	DE Finanz-Tracker 🔳 🔳 🔍 Q Q V 🛓 V
Definieren	Ergebnisse importieren	
📇 Planen	Beim Import von Testergebnissen für unbekannte T Beim Import von Testergebnissen ein Testsuiteergebnis erzeugen und fall	'ests automatisch neue Testfälle anlegen 🗹 s notwendig eine neue Testsuite anlegen 🗹
	Name der zu importierenden Testsuite 🗢	Zugeordnete Testsuite 🗢
Ausiumen	de.verit.klaros.arquillian.test.LoginTest	TS00008 - Überprüfen Sie die Verbindung zum Server 🗸 🗸 🗸
🕒 Auswerten	Name des zu importierenden Testfalls 🕈	Zugeordneter Testfall 🗘
	de.verit.klaros.arquillian.test.LoginTest.loginAdmin	TC00018 - Testen der Verbindung zum Server der Bank.
🔑 Konfigurieren	de.verit.klaros.arquillian.test.LoginTest.loginManager	✓
		Importieren Abbrechen Zurück

Abbildung 8.19. Die Seite "Ergebnisse importieren"

Mit Klick auf den Button Import werden alle Ergebnisse in Klaros-Testmanagement importiert.

8.5.1. Unterstützte Frameworks

Die folgenden Frameworks werden unterstützt:

AUnit	AUnit ist eine Anpassung der Java JUnit und C++ CppUnit Unit- Test-Frameworks für Ada Code.
Boost Test	Die Boost Test -Bibliothek bietet einen abgestimmten Satz von Komponenten zum Schreiben von Testprogrammen, zum Organisieren von Tests in einfache Testfälle und Testsuiten und zum Steuern ihrer Laufzeitausführung.
Check	Check ist ein Unit-Test-Framework für C.
CppTest	CppTest ist ein portables und leistungsfähiges, aber dennoch einfaches Unit-Test-Framework zum Durchführen automati- sierter Tests in C++. Der Fokus liegt auf Benutzerfreundlichkeit und Erweiterbarkeit.
CppUnit	CppUnit ist ein C++ Unit-Test-Framework.
ctest	ctest ist ein von CMake bereit gestellter Testtreiber.
CUnit	CUnit ist ein leichtgewichtiges System zum Schreiben, Verwal- ten und Ausführen von Unit-Tests in C.
embUnit	Embedded Unit ist ein Unit-Test-framework für eingebettete C- Systeme.
Fitnesse	Fitnesse ist ein Werkzeug zum Spezifizieren und Verifizieren von Anwendungsakzeptanzkriterien (Anforderungen).
Free Pascal Unit	Free Pascal Unit ist eine Portierung des JUnit-Core Frame- works nach Free Pascal.
GLib/gtester	GLib stellt die zentralen Anwendungsbausteine für in C geschriebene Bibliotheken und Anwendungen zur Verfügung. Es bietet das in GNOME verwendete Kernobjektsystem, die Implementierung der Hauptschleife und einen großen Satz von Hilfsfunktionen für Zeichenketten und allgemeine Daten-

	strukturen. gtester ist ein Dienstprogramm zum Ausführen von Unit-Tests, die mit dem GLib-Test-Framework geschrieben wurden.
Gauge	Gauge is ist ein leichtgewichtiges, plattformübergreifendes Testautomatisierungswerkzeug mit der Möglichkeit, Testfälle in einer Geschäftssprache zu verfassen.
GoogleTest	GoogleTest ist eine von Google erstellte Unit-Testing-Biblio- thek für C++.
JBehave	JBehave ist ein Framework für verhaltensgetriebene Entwick- lung (Behaviour-Driven Development, BDD). BDD ist eine Wei- terentwicklung der testgetriebenen Entwicklung (TDD) und des akzeptanztestgetriebenen Designs, und will diese Prakti- ken für Neueinsteiger und Experten gleichermaßen zugängli- cher und intuitiver machen. Es verlagert das Vokabular von testbasiert auf verhaltensbasiert und positioniert sich als Designphilosophie.
JMeter	JMeter ist eine 100 % reine Java-Anwendung, die für Last- tests des Funktionsverhaltens und zur Leistungsmessung ent- wickelt wurde. Sie wurde ursprünglich für das Testen von Web- anwendungen entwickelt, ist aber inzwischen auf andere Test- funktionen erweitert worden.
JsUnit	JsUnit ist ein Unit-Testing-Framework für client-seitiges (browser-internes) JavaScript. Es ist im Wesentlichen eine Portierung von JUnit auf JavaScript. Ebenfalls enthalten ist eine Plattform zur Automatisierung der Ausführung von Tests auf mehreren Browsern und mehreren Rechnern mit unter- schiedlichen Betriebssystemen.
Jubula	Jubula bietet automatisierte funktionale GUI-Tests für verschiedene Arten von Anwendungen.
JUnit	JUnit ist ein einfaches Framework, um wiederholbare Tests zu schreiben. Es ist eine Instanz der xUnit-Architektur für Unit-Test-Frameworks.
MbUnit	MbUnit ist ein erweiterbares Unit-Testing-Framework für das .NET Framework, das das xUnit-Pattern-Testing aufnimmt und darüber hinausgeht. MbUnit ist Teil des Gallio-Bundles.
MSTest	MSTest ist der Befehlszeilenbefehl, der zum Ausführen von Tests verwendet wird. Dieser Befehl verfügt über mehrere Optionen, mit denen Sie Ihren Testlauf anpassen können.
NUnit	NUnit ist ein Unit-Testing-Framework für alle .Net-Sprachen, das vollständig in C# geschrieben ist und komplett neu entwi- ckelt wurde, um die Vorteile vieler .NET-Sprachfeatures zu nut- zen, z. B. benutzerdefinierte Attribute und andere reflexionsbe- zogene Fähigkeiten.

PHPUnit	PHPUnit ist ein auf Programmierer ausgerichtetes Test- Framework für PHP. Es ist eine Variante der xUnit-Architektur für Unit-Test-Frameworks.
QF-Test	QF-Test ist ein professionelles Werkzeug für das automatisier- te Testen von Java- und Web-Anwendungen mit einer grafi- schen Benutzeroberfläche.
QTestLib	QTestLib Framework ist ein Werkzeug für Unit-Tests von Qt- basierten Anwendungen und Bibliotheken.
Ranorex	Ranorex ist ein GUI-Testautomatisierungs-Framework für das Testen von Desktop-, webbasierten und mobilen Anwendun- gen.
Selenium	Selenium ist ein Webbrowser-Automatisierungstool, das hauptsächlich für das automatisierte Testen von Webanwen- dungen verwendet wird. Selenium ist in der Lage, JUnit -kom- patible Testergebnisse, die in Klaros-Testmanagement impor- tiert werden können.
Squish	Squish ist ein GUI-Testautomatisierungswerkzeug für alle Arten von plattformübergreifenden Desktop-, mobilen, einge- betteten und Web-Anwendungen. Squish ist in der Lage JUnit - kompatible Testergebnisse, die in Klaros-Testmanagement importiert werden können.
TestComplete	TestComplete ist ein plattformübergreifendes automatisiertes GUI-Testwerkzeug von SmartBear Software.
TestComplete TESSY	 TestComplete ist ein plattformübergreifendes automatisier- tes GUI-Testwerkzeug von SmartBear Software. TESSY ist ein Software-Tool zur Automatisierung von Modul- Einheitstests von Embedded-Software, die in verschiedenen Embedded-Dialekten der Programmiersprachen C und C++ geschrieben wurde.
TestNG	 TestComplete ist ein plattformübergreifendes automatisiertes GUI-Testwerkzeug von SmartBear Software. TESSY ist ein Software-Tool zur Automatisierung von Modul-Einheitstests von Embedded-Software, die in verschiedenen Embedded-Dialekten der Programmiersprachen C und C++ geschrieben wurde. TestNG ist ein Test-Framework, das von JUnit und NUnit inspiriert ist, aber einige neue Funktionalitäten einführt, die es leistungsfähiger und einfacher in der Anwendung machen. TestNG ist in der Lage JUnit-kompatible Testergebnisse zu erzeugen, die in Klaros-Testmanagement importiert werden können.
TestNG TUSAR	 TestComplete ist ein plattformübergreifendes automatisiertes GUI-Testwerkzeug von SmartBear Software. TESSY ist ein Software-Tool zur Automatisierung von Modul-Einheitstests von Embedded-Software, die in verschiedenen Embedded-Dialekten der Programmiersprachen C und C++ geschrieben wurde. TestNG ist ein Test-Framework, das von JUnit und NUnit inspiriert ist, aber einige neue Funktionalitäten einführt, die es leistungsfähiger und einfacher in der Anwendung machen. TestNG ist in der Lage JUnit-kompatible Testergebnisse zu erzeugen, die in Klaros-Testmanagement importiert werden können. TUSAR steht für Thales Unified Software Analysis Report. Es ist ein generisches Metrikformat und besteht aus 4 Kategorien: Coverage, Measure, Test und Violations.
TestComplete TESSY TestNG TUSAR Unified Functional Testing (UFT) / QuickTest Professional (QTP)	 TestComplete ist ein plattformübergreifendes automatisiertes GUI-Testwerkzeug von SmartBear Software. TESSY ist ein Software-Tool zur Automatisierung von Modul-Einheitstests von Embedded-Software, die in verschiedenen Embedded-Dialekten der Programmiersprachen C und C++ geschrieben wurde. TestNG ist ein Test-Framework, das von JUnit und NUnit inspiriert ist, aber einige neue Funktionalitäten einführt, die es leistungsfähiger und einfacher in der Anwendung machen. TestNG ist in der Lage JUnit-kompatible Testergebnisse zu erzeugen, die in Klaros-Testmanagement importiert werden können. TUSAR steht für Thales Unified Software Analysis Report. Es ist ein generisches Metrikformat und besteht aus 4 Kategorien: Coverage, Measure, Test und Violations. Unified Functional Testing ist eine Software für funktionale automatisierte Tests von Micro Focus, früher bekannt als QuickTest Professional (QTP) von HPE Software.

Geschwindigkeit und ein geringer Platzbedarf sind sehr wichtige Aspekte.

ValgrindValgrind ist ein Instrumentierungs-Framework zur Erstellung
von dynamischen Analysewerkzeugen.

xUnit.net

xUnit.net ist ein freies, quelloffenes, Community-fokussiertes Unit-Testing-Tool für das .NET Framework. xUnit.net arbeitet mit ReSharper, CodeRush, TestDriven.NET und Xamarin.



Unterstützung für Frameworks, die hier nicht aufgelistet sind

Auch wenn Sie Ihr bevorzugtes Testframework hier nicht gefunden haben, bedeutet das nicht, dass es nicht bereits unterstützt wird.

Viele Tools bieten an, ihre Ergebnisdateien in das JUnit XML Format zu exportieren (z.B. Selenium, Robot und andere), welches ein unterstütztes Format ist.

Wenn Sie ein zusätzliches Format unterstützt sehen möchten, wenden Sie sich bitte an support@verit.de.

Kapitel 9. Auswerten

Dieses Kapitel zeigt, wie die beim Testen angefallenen Daten und Ergebnisse ausgewertet werden können. Es wird beschrieben, wie das Dashboard eingesetzt und angepasst wird, Berichte konfiguriert und angezeigt werden können und wie die gefundenen Issues an das angebundene Issue-Management-System übergeben, Testfällen oder Testsystemen zugeordnet und bei Bedarf geändert werden können.

9.1. Das Dashboard

Ein Dashboard ist eine Übersichtsseite für als Diagramm dargestellte, konfigurierbare Berichte. Dashboards können individuell zusammengestellt und für andere Benutzer freigegeben oder gesperrt werden. Jeder Benutzer kann beliebig viele Dashboards zusammenstellen. Ein Beispiel für ein Dashboard ist in <u>Abbildung 9.1</u> zu sehen.

Neu fügt einen neuen Bericht zum Dashboard hinzu.

Speichern speichert alle Änderungen am Dashboard und den Berichten.

Abbrechen verwirft alle Änderungen am Dashboard und den Berichten.



Abbildung 9.1. Die Seite "Dashboard"

Benutzer können jederzeit zwischen verschiedenen Dashboards wechseln. In der Dropdown-Liste auf der rechten oberen Seite werden alle verfügbaren Dashboards (private und von anderen Benutzern freigegebene) angezeigt. Daneben befinden sich mehrere Icons:

- 密 Wenn ausgewählt, können alle Benutzer dieses Dashboard verwenden.
- A Wenn ausgewählt, kann nur der Eigentümer das Dashboard bearbeiten.

☆ Wenn ausgewählt, ist dies das Standard-Dashboard.

- Erzeugt eine private Kopie des Dashboards.
- Di Löscht das Dashboard und alle Berichte.

Die Anzahl der Spalten, in denen die Berichte angezeigt werden sollen, lässt sich oben rechts in den mit 1 bis 4 nummerierten Buttons festlegen. Die aktuell ausgewählte Anzahl von Berichten pro Zeile ist orange markiert. Berichte können durch Ziehen und Ablegen an eine andere Position verschoben werden.

Alle Dashboard-Berichte sind gleich aufgebaut: Sie bestehen aus einer Titelleiste und dem Bereich, in dem das Diagramm angezeigt wird. Die Titelleiste enthält den Namen des Berichts und mehrere Icons:

- Öffnet einen Dialog zum Bearbeiten der Berichtskonfiguration.
- A Lädt den Bericht als PNG-Datei herunter.
- Maximiert/minimiert den Bericht.
- × Löscht den Bericht.

9.1.1. Standard-Dashboard

Beim ersten Start von Klaros-Testmanagement wird ein Standard-Dashboard mit drei vordefinierten Berichten erstellt. Dieses Dashboard ist für alle Benutzer freigegeben und wird automatisch für alle neuen Benutzer angezeigt. Administratoren können das Standard-Dashboard ändern, indem sie ein beliebiges anderes Dashboard als Standard-Dashboard festlegen.



Anmerkung

Es gibt stets nur ein einziges Standard-Dashboard. Wird ein neues Standard-Dashboard festgelegt, verliert das vorherige Standard-Dashboard diesen Status.

9.1.2. Dashboard-Berichte bearbeiten

Beim Klick auf das Icon 😰 öffnet sich ein Dialog, der alle konfigurierbaren Parameter des Berichts anzeigt, sowie den Berichtstyp, der nicht geändert werden kann. Änderungen können durch Klick auf den Button OK auf den Bericht angewendet oder durch Klick auf den Button Abbrechen verworfen werden.



Abbildung 9.2. Der Dialog "Report konfigurieren"

9.1.3. Berichtstypen

Alle Berichte zeigen initial die Auswertung der Daten aus dem aktiven Projekt an.

Zusätzlich zu den Daten des aktiven Projekts, können in der Klaros-Testmanagement Enterprise Edition auch Daten bestimmter Projekte angezeigt werden.

Die Klaros-Testmanagement Community Edition enthält drei Berichtstypen:

- Projektübersicht
- Letzte Erfolgsrate
- Testaktivität

enterprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Die Klaros-Testmanagement Enterprise Edition enthält zudem vier zusätzliche Berichtstypen:

- Testsystem Übersicht
- Testumgebung Übersicht
- Testfortschritt Historie
- Testfortschritt

Alle Berichte sind in den folgenden Abschnitten detailliert beschrieben.

9.1.3.1. Der Bericht "Projektübersicht"

Der Bericht *Projektübersicht* zeigt verschiedene Metriken eines einzelnen Projekts in einem Diagramm.



Abbildung 9.3. Der Bericht "Projektübersicht"

Dieser Bericht zeigt die folgenden Metriken:

- Die Anzahl der ausgeführten/nicht ausgeführten Testfälle.
- Die Anzahl der ausgeführten/nicht ausgeführten Testsuiten.
- Die Anzahl der geplanten/ausgeführten Aufgaben.

Die folgenden Parameter des Berichtsdiagramms "Projektübersicht" sind konfigurierbar:

- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.
- 9.1.3.2. Der Bericht "Letzte Erfolgsrate"

Der Bericht *Letzte Erfolgsrate* zeigt die neuesten Ergebnisse der ausgeführten Testfälle für die ausgewählte Kombination aus Testsystem (eins oder mehrere) und Testumgebung (eine oder mehrere).

Auswerten



Abbildung 9.4. Der Bericht "Letzte Erfolgsrate"

Die folgenden Parameter des Berichtsdiagramms "Letzte Erfolgsrate" sind konfigurierbar:

- Ein oder mehrere Testsysteme.
- Ein oder mehrere Testumgebungen.
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.

9.1.3.3. Der Bericht "Testaktivität"

Der Bericht *Testaktivität* zeigt die Testfallergebnisse für eine ausgewählte Kombination aus einem oder mehreren Testsystemen und Testumgebungen in einem ausgewählten Zeitraum als Histogramm an.





Die folgenden Parameter des Berichtsdiagramms "Testaktivität" sind konfigurierbar:

- Projekt
- Iteration (sofern vorhanden)
- Ein oder mehrere Testsysteme
- Ein oder mehrere Testumgebungen
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.
- Die zu betrachtende Zeitperiode in Tagen.
- 9.1.3.4. Der Bericht "Testsystem Übersicht"



Der Bericht *Testsystem - Übersicht* zeigt die Erfolgs- und Fortschrittsrate von Testumgebungen unter einem einzigen Testsystem in einem Radardiagramm an. Sind weniger als drei Testumgebungen konfiguriert, wird stattdessen ein Balkendiagramm angezeigt.





Die folgenden Parameter des Berichtsdiagramms "Testsystem - Übersicht" sind konfigurierbar:

- Das Projekt.
- Iteration, sofern vorhanden.
- Ein Testsystem.
- Ein oder mehrere Testumgebungen.
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.
- 9.1.3.5. Der Bericht "Testumgebung Übersicht"



Der Bericht *Testumgebung* - Übersicht zeigt die Erfolgs- und Fortschrittsrate von Testsystemen unter einer einzigen Testumgebung in einem Radardiagramm an. Sind weniger als drei Testsysteme konfiguriert, wird stattdessen ein Balkendiagramm angezeigt.





Die folgenden Parameter des Berichtsdiagramms "Testumgebung - Übersicht" sind konfigurierbar:

- Projekt
- Iteration, sofern vorhanden.
- Ein oder mehrere Testsysteme.
- Eine Testumgebung.
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.

9.1.3.6. Der Bericht "Testfortschritt"



Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Der Bericht *Testfortschritt* zeigt die Rate der ausgeführten gegenüber den definierten Testfällen eines Projekts für eine bestimmte Menge von Testumgebungen und Testsystemen.





Die folgenden Parameter des Berichtsdiagramms "Testfortschritt" sind konfigurierbar:

- Projekt
- Iteration, sofern vorhanden.

- Ein oder mehrere Testsysteme.
- Ein oder mehrere Testumgebungen.
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.
- Die Auswahl, ob nur Testfälle berücksichtigt werden sollen, die Aufgaben zugeordnet sind.

9.1.3.7. Der Bericht "Testhistorie"



Der Bericht *Testhistorie* zeigt die Rate der definierten Testfälle gegenüber den ausgeführten Testfällen eines Projekts für eine oder mehrere Testumgebungen und ein oder mehrere Testsysteme in einer bestimmten Zeitperiode an.



Abbildung 9.9. Der Bericht "Testhistorie"

Die folgenden Parameter des Berichtsdiagramms "Testhistorie" sind konfigurierbar:

- Projekt
- Iteration, sofern vorhanden.

- Ein oder mehrere Testsysteme.
- Ein oder mehrere Testumgebungen.
- Der Name des Berichtsdiagramms.
- Die Auswahl, ob in diesem Bericht das aktive Projekt oder ein anderes verwendet werden soll (nur in der Klaros-Testmanagement Enterprise Edition enthalten).
- Falls nicht das das aktive Projekt verwendet wird, können folgende zu betrachtende Objekte ausgewählt werden:
- Die Auswahl, ob in diesem Bericht die aktive Iteration oder eine andere verwendet werden soll.
- Falls nicht die aktive Iteration verwendet wird, wird hier eine Dropdown-Liste mit den verfügbaren Iterationen angezeigt.
- Die zu betrachtende Zeitperiode in Tagen

9.2. Berichte

Auf der Seite *Berichtsvorlagen* können vordefinierte Berichte als PDF- oder HTML-Datei abgerufen werden. Mit der Klaros-Testmanagement Enterprise Edition können außerdem konfigurierbare Berichte erstellt, angepasst und als PDF oder Excel-Datei abgerufen werden.

9.2.1. Vordefinierte Berichte



Abbildung 9.10. Die Seite "Berichtsvorlagen"

Folgende Berichte sind vordefiniert:

- Der "Testumgebungsbericht"
- Der "Testsystembericht"
- Der "Testsuitebericht"

• Der "Testlauf-Ergebnishistorie"

Die Berichte können in zwei verschiedenen Dateiformaten abgerufen werden:

- 🔎 PDF
- 🗟 HTML

Beim Klick auf eines der Dateityp-Icons wird der Bericht im gewählten Format abgerufen.

Der obere Teil der Seite Berichte zeigt eine Tabelle mit den vorhandenen vordefinierten Berichten:

9.2.1.1. Testumgebungsbericht

Der Testumgebungsbericht zeigt alle Testläufe einer einzelnen Testumgebung mit Ausführungsdatum, ausführendem Benutzer und der Anzahl der Ergebnisse Bestanden, Fehlschlag, Fehler, Unklar und Übersprungen an. Ein Beispiel für den Bericht ist in <u>Abbildung 9.11</u> zu sehen.

Vert Kinnanagement Testumgebungsbericht								
Projekt:	DE Finanz-Tra	DE Finanz-Tracker (P00005)				Felix Mustermann		
Testumgebungen:	5			Da	atum:	07.12.2021	15:50	
ld:	ENV00005							
Beschreibung:	Android 8 Sm	artphone						
Testlauf		Ausführungsdatum	Ausgeführt von	Bestanden	<u>Unklar</u>	<u>Fehlschlag</u>	<u>Fehler</u>	Übersprungen
P00005 TRU0000050: TC00	012 - Überweisen	02.07.20, 14:34:23	Thomas Tafel	0	0	0	1	0
Sie Geld, ohne einen Betrag P00005 TRU0000054: TC00	anzugeben. 012 - Überweisen	07.07.20, 09:33:32	Thomas Tafel	0	0	0	1	0
Sie Geld, ohne einen Betrag P00005 TRU0000064: TC00	anzugeben. 010 - Überweisen	07.07.20, 09:33:32	Tim Thiel	1	0	0	0	0
Sie Geld, ohne einen Empfär P00005 TRU0000065: TC00	nger anzugeben. 004 - Überweisen	07.07.20.09:33:32	Tim Thiel	1	0	0	0	0
Sie Geld an eine ausländisch P00005 TRU0000130: TC00	ne Bank. 010 - Überweisen	09.07.20.06:53:55	Timo Tunklik	1	0	0	0	0
Sie Geld, ohne einen Empfär P00005 TRU0000093: TC00	nger anzugeben. 012 - Überweisen	09 07 20 06:53:55	Tania Toppler	1	0	0	0	0
Sie Geld, ohne einen Betrag	anzugeben. 013 - Überweisen	09.07.20.06:53:55	Timo Tunklik	1	0	0	0	0
Sie Geld ohne Angabe einer	Referenz.	09.07.20, 00.55.55						0
Sie Geld ohne Angabe einer	013 - Uberweisen Referenz.	09.07.20, 06:53:55	limo lunklik	1	U	U	U	U
P00005 TRU0000202: TC00	011 - Überweisen	13.07.20, 13:10:31	Markus Meyer	1	0	0	0	0
P00005 TRU0000168: TC00	012 - Überweisen	13.07.20, 13:10:31	Till Tegen	1	0	0	0	0
Sie Geld, ohne einen Betrag P00005 TRU0000163: TC00	anzugeben. 011 - Überweisen	13.07.20. 13:10:31	Markus Mever	1	0	0	0	0
Sie Geld ohne Angabe der IE P00005 TRU0000171 TC00	BAN. 012 - Überweisen	13.07.20.13:10:31	Till Tegen	1	0	0	0	0
Sie Geld, ohne einen Betrag	anzugeben.	14.07.20.08:57:17	Felix Mustermann	1	0	0	0	0
einen Dauerauftrag hinzu.	ors-rugen sie	14.07.20, 08:37:17	reix wustermann					
PUUUU5 TRU0000255: TC00 einen Dauerauftrag hinzu.	015 - Fügen Sie	14.07.20, 08:57:17	Felix Mustermann	1	0	0	U	U

Abbildung 9.11. Der Bericht "Testumgebung"

9.2.1.2. Testsystembericht

Der *Testsystembericht* zeigt alle Testläufe eines einzelnen Testsystems mit Ausführungsdatum, ausführendem Benutzer und der Anzahl der Ergebnisse *Bestanden*, *Fehlschlag*, *Fehler*, *Unklar* und *Übersprungen* an. Ein Beispiel für den Bericht ist in <u>Abbildung 9.12</u> zu sehen.



Testsystembericht



Projekt:	DE Finanz-	racker (P00005)		A	utor:	Felix Muste	rmann	
Testsysteme:	6			Da	atum:	07.12.2021,	15:53	
ld:	SUT00001							
Version:	Finance Tracker 1.0.	0						
T (1) (
lestiaur		Austunrungsdatum	Ausgefunrt von	Bestanden	Unklar	Fenischiag	Fenier	Ubersprungen
P00005 TRU0000010:	TC00012 - Uberweisen Sie	01.07.20, 07:03:00	Tim Thiel	0	1	0	0	0
P00005 TRU0000006:	TC00010 - Überweisen Sie	01.07.20, 07:03:00	Timo Tunklik	0	1	0	0	0
Geld, ohne einen Empfi P00005 TRU0000005:	änger anzugeben. TC00012 - Uberweisen Sie	01.07.20, 07:03:00	Till Tegen	0	1	0	0	0
Geld, ohne einen Betra P00005 TRU0000011:	g anzugeben. TC00013 - Überweisen Sie	01.07.20, 07:03:00	Timo Tunklik	0	1	0	0	0
Geld ohne Angabe eine P00005 TRU0000016:	er Referenz. TC00005 - Kontostand	01.07.20, 07:03:00	Till Tegen	0	0	1	0	0
P00005 TRU0000013:	TS00008 - Überprüfen Sie	01.07.20, 07:03:00	Sabrina Gidley	0	0	1	0	0
P00005 TRU0000003:	TC00010 - Überweisen Sie	01.07.20, 07:03:00	Timo Tunklik	0	0	1	0	0
P00005 TRU0000021:	anger anzugeben. TC00011 - Überweisen Sie	01.07.20, 07:03:00	Till Tegen	0	1	0	0	0
P00005 TRU0000012:	IBAN. TS00008 - Überprüfen Sie	01.07.20, 07:03:00	Timo Tunklik	0	0	1	0	0
P00005 TRU0000014:	erver TS00008 - Überprüfen Sie	01.07.20, 07:03:00	Sabrina Gidley	0	1	0	0	0
P00005 TRU0000009:	TC00012 - Überweisen Sie	01.07.20, 07:03:00	Tim Thiel	0	1	0	0	0
P00005 TRU0000007:	g anzugeben. TC00010 - Überweisen Sie	01.07.20, 07:03:00	Timo Tunklik	0	1	0	0	0
Geld, ohne einen Empf P00005 TRU0000001:	änger anzugeben. TC00013 - Uberweisen Sie	01.07.20, 07:03:00	Till Tegen	0	1	0	0	0
Geld ohne Angabe eine P00005 TRU0000004:	er Referenz. TC00013 - Überweisen Sie	01.07.20, 07:03:00	Till Tegen	0	1	0	0	0
Geld ohne Angabe eine P00005 TRU0000002:	er Referenz. TC00012 - Überweisen Sie	01.07.20, 07:03:00	Till Tegen	0	0	1	0	0
Geld, ohne einen Betra P00005 TRU0000008:	g anzugeben. TC00013 - Überweisen Sie	01.07.20, 07:03:00	Timo Tunklik	0	0	1	0	0
Geld ohne Angabe eine P00005 TRU0000015:	er Referenz. TS00002 - Überweisungen	01.07.20, 07:03:00	Tim Thiel	0	7	1	0	0

Abbildung 9.12. Der Bericht "Testsystem"

9.2.1.3. Testsuitebericht

Der *Testsuitebericht* zeigt für die Testsuiten des aktiven Projekts alle Testfälle mit Team, Ersteller, Priorität, Namen an. Ein Beispielbericht ist in <u>Abbildung 9.13</u> zu sehen.

		Test	suite - Üb	ersicht
Projekt:	DE Finanz-Trac	ker (P00005)		Autor: Sandra Selen
Testsuiten:	7			Datum: 07.04.2021, 17:06
ld:	TS00002	Testfälle:	8	Testläufe: 7
Name:	Überweisunger	n testen		
Testfall	<u>Team</u>	Erstellt von	Priorität	Name
TC00013	Team Alpha	Talal Arif	Low	Überweisen Sie Geld ohne Angabe einer Referenz
TC00012	Alpha Theam	Talal Arif	High	Überweisen Sie Geld, ohne einen Betrag anzugeben
TC00011		Talal Arif	High	Überweisen Sie Geld ohne Angabe der IBAN.
TC00010	Team Alpha	Talal Arif	Low	Überweisen Sie Geld, ohne einen Empfänger anzugeben.
TC00004	Team Alpha	Sabrina Gidley	Low	Überweisen Sie Geld an eine ausländische Bank.
TC00003	Team Alpha	Sabrina Gidley	Low	Überweisen Sie Geld auf ein anderes Bankkonto
TC00002	Team Alpha	Sabrina Gidley	High	Geldtransfer mit unzureichenden Geldmitteln.
TC00001	Team Alpha	Sabrina Gidley	Medium	Überweisen Sie Geld auf ein lokales Bankkonto.
ld:	TS00001	Testfälle:	4	Testläufe: 12
Name:	Daebboard akt	ualisieren		

Abbildung 9.13. Der Bericht "Testsuite"

9.2.1.4. Testlaufbericht

Der Testlaufbericht zeigt alle Testläufe eines einzelnen Testsystems mit Ausführungsdatum, ausführendem Benutzer und der Anzahl der Ergebnisse Bestanden, Fehlschlag, Fehler, Unklar und Übersprungen innerhalb eines bestimmten Zeitraumes an.



Abbildung 9.14. Der Bericht "Testlauf-Ergebnishistorie"

9.2.1.5. Konfigurierbare Berichte



Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Zusätzlich zu den vordefinierten Berichten können mit der Klaros-Testmanagement Enterprise Edition PDFs und Excel-Dateien aus benutzerdefinierten Berichten erzeugt werden. Das Erstellen benutzerdefinierter Berichte wird in <u>Abschnitt 10.2, "Berichtsvorlagen"</u> erklärt. Mit Klick auf die Icons 🕒 und 🔀 werden die Berichte erzeugt.



Anmerkung

Beim Klick auf eines der Icons 🗋 oder 🗟 öffnet sich ein Dialog, in dem Berichtsparameter eingestellt werden können, falls diese für das Erzeugen benötigt werden.

≡ 👱 K L A R	os test manag	EMENT DE Finanz-Tracker 🖀 🗮	۹ 0۰ ≞۰				
📝 Definieren	Berichte						
🔹 Planen	Name Testumgebungsbericht Bericht üb Testsystembericht Bericht üb	Beschreibung er die Testumgebungen und ihre Testergebnisse er die Testsysteme und ihre Testergebnisse	Aktion				
Austuhren	Testlaufbericht Bericht ub Testlaufbericht Testlauf-H	Testsuitebericht Bericht über die Testsuiten und ihre Testergebnisse Testlaufbericht Testlauf-Historie für ausgewähltes Testsystem und Zeitraum					
e Auswerten	Konfigurierbare Berichte	TElevine Seintunt M J N M 10 y	7 ይ =				
€ Konfigurieren	Name 2 153 Bitte geben Sie die benöligten Joi Joi Joi Joi Sort by (primary) Syt Sort by (secondary) Ter Test Environment Sort Order Show Passed Results Start Date End Date	Beschreibung 2 Revision Geändert von 2 Parameter ein SUT00001 Result (ascending) Duration (ascending) Test Environment (descending) Image OK	Geöndert ◆ Format ◆ Aktion 8.04.21, 04:34 PDF ▲ 7.07.21, 04:24 PDF ▲ 4.04.21, 11:17 Excel ▲ 5.07.21, 04:20 PDF ▲ 5.07.21, 04:20 PDF ▲ 5.06.21, 04:22 PDF ▲				

Abbildung 9.15. Einen parametrisierten Bericht erzeugen



Anmerkung

Der Button OK wird aktiv, sobald alle erforderlichen Parameter eingegeben wurden.

9.3. Testläufe

Diese Seite listet alle Testläufe auf, die im aktiven Projekt ausgeführt wurden.



Anmerkung

Auf dieser Seite werden nur vollständig ausgeführte Testfälle angezeigt. Unterbrochene Testläufe können unter <u>Abschnitt 8.4, "Testlauf fortsetzen"</u> fortgesetzt werden.
								Finane Tracker 🖀	-		α × 0 × ₽
Definieren Test	tläufe										
Directory										Speich	ern Verwarfer
Planen	88		340 Einträge-Seite 1 von 12 🕅 🖣	1 2 3 4	к ► н	30 ¥			Ate anzeigen		Q × &
Ausführen	10 ¢ 0	Start Zugewiesen	iteration @	Aufgabe 🔍 T		Testungebung *	Testsystem 🕈	Ergebrisse 🕈	🕲 o 🔻 o 🕲 o	* (Aktion
1	FRU0005663	vor 2 Jahren Felix Mustermann		т	000005	Android 8 Smartphone	Finance Tracker 2.1.0	1		1	888Q
Auswerten	FRU0005661	vor 2 Jahren Felix Mustermann		J0800355 T	C00015	Android 8 Smartphone	Finance Tracker	1		1	0000
	FRU0005660	vor 2 Jahren Felix Mustermann		т	\$00007	Android 8 Smartphone	Finance Tracker	3		3	0 8 8 Q
Konfigurieren	FRU0005659	vor 2 Jahren Felix Mustermann		т	000006	Android 8 Smartphone	Finance Tracker	1		1	0000
- D1	TRUCCOS658	vor 2 Jahren Felix Mustermann		т	000020	Android 8 Smartphone	Finance Tracker	1		1	0000
	FRU0005365	vor 3 Jahren Felix Mustermann		т	500038	Android Tablet	Finance Tracker	2		1	0000
	FRU0005364	vor 3 Jahren Felix Mustermann		т	\$00034	Android 8 Smartphone	2.1.0 Finance Tracker	2	2		0000
01	FRU0005363	vor 3 Jahren Felix Mustermann		T	500034	Android 8 Smartphone	2.1.0 Finance Tracker	2			
- D1	FRU0005362	vor 3 Jahren Felix Mustermann		Ţ	000005	Android Tablet	2.1.0 Finance Tracker	1			0000
01	FRU0005361	vor 3 Jahren Felix Mustermann		т	000001	Android 8 Smartphone	2.1.0 Finance Tracker	1	1		0000
01	FRU0001757	vor 4 Jahren Felix Mustermann		т	000022	Android 8 Smartphone	Einance Tracker	1		1	0000
	FRU0001732	vor 4 Jahren Felix Mustermann	2.1.x-Sprint 01-Optimierung für die Verwendung auf	т	C00014	Android 10	Enance Tracker	1			0000
	FRU0001731	vor 4 Jahren Felix Mustermann	2.1.x-Sprint 01-Optimierung für die Verwendung auf	т	000005	Android 10	Finance Tracker	1			0000
	FRU0001750	vor 4 Jahren Felix Mustermann	2.1.x-Sprint 01-Optimierung für die Verwendung auf	т	000022	Android Tablet	Finance Tracker	1			0000
	FRU0001729	vor 4 Jahren Felix Mustermann	2.1.x-Sprint 01-Optimierung für die Verwendung auf	т	\$00002	Android Smartwatch	Finance Tracker	8	8 2 3	2	0000
	FRU0001728	vor 4 Jahren Felix Mustermann	2.1.x-Sprint 01-Optimierung für die Verwendung auf	т	\$00036	Android 9 Smartphone	Finance Tracker	7	7		0000
	FRUD001727	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 02-Dashboard erweitern	т	000005	Android Tablet	Finance Tracker	1			0000
	FRU0001726	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 02-Dashboard erweitern	T	000023	Android Tablet	Finance Tracker	1	1		0000
	FRU0001725	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 02-Dashboard erweitern	т	\$00001	Android 9 Smartphone	Finance Tracker	4	2 1 1		0000
	FRUD001724	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 02-Dashboard erweitern	т	\$00036	Android 10	Finance Tracker	7	4		0000
	1910001723	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 02-Dashboard erweitern	т	500022	Android Tablet	Finance Tracker	8	5		0000
01	FRU0001721	vor 4 Jahren Felix Mustermann	2.0.x Sprint 01-Dashboard hinzufügen	т	800007	Android 10	Finance Tracker	3			0000
	FRU0001720	vor 4 Jahren Felix Mustermann	2.0.x-Sprint 01-Dashboard hinzufligen	т	\$00007	Android Tablet	Finance Tracker	3	3		0000

Abbildung 9.16. Die Seite "Testläufe"

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Start	Der Startzeitpunkt des Testlaufs.
Zugewiesen	Der Name des Benutzers, dem der Testlauf zugewiesen ist.
Iteration	Die Iteration, in der der Testlauf ausgeführt wurde.
Aufgabe	Die Aufgabe, mit der der der Testlauf ausgeführt wurde.
Testfall/-suite	Die ID des Testfalls oder der Testsuite des Testlaufs.
Testumgebung	Die Testumgebung, in der der Test ausgeführt wurde.
Testsystem	Das Testsystem, das getestet wurde.
Ergebnisse	Die Anzahl der Testfall-Ergebnisse.
Bestanden	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Bestanden".
Fehlschlag	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Fehlschlag".
Fehler	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Fehler".
Unklar	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Unklar".
Übersprungen	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Übersprungen".
Aktion	Die ausführbaren Aktionen.

9.3.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- Destlauf-Bericht im PDF-Format erzeugen.
- Testlauf-Bericht im HTML-Format erzeugen.
- 🔒 Druckansicht öffnen
- Q Die Testfallergebnisse dieses Testlaufs anzeigen.
- Diesen Testlauf löschen.

Wurde ein Testlauf gelöscht, ist er zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Testläufe steht dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Den Testlauf wiederherstellen.
- ⊘ Den Testlauf unwiederbringlich aus der Datenbank entfernen (nur Administrator).



Einen Testlauf löschen

Beim Löschen eines Testlaufs werden die zugehörigen Testfall- und Testsuiteergebnisse ebenfalls gelöscht.

Sind alle Testläufe für ein bestimmtes Testsystem oder eine bestimmte Testumgebung gelöscht, kann auch das jeweilige Testsystem oder die Testumgebung gelöscht werden. Andernfalls bleiben diese für das Löschen gesperrt.

9.3.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testläufe gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testläufe aus, auf die die Aktion angewendet werden soll.

Für Testläufe werden folgende Massenaktionen unterstützt:

- 🛯 Bearbeiten
- Druckansicht öffnen
- fi Löschen

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

9.3.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- Q Suchen
- 凸 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

9.3.4. Der Testlauf - Bericht

Der Bericht *Testlauf* zeigt die Details eines Testlaufs mit Ausführungsdatum, ausführendem Benutzer, Testsystem, Testumgebung und der Anzahl der Ergebnisse *Bestanden, Fehlschlag, Fehler, Unklar* und *Übersprungen* an. Er kann als PDF oder HTML-Datei ausgegeben werden. Ein Beispiel-PDF ist in <u>Abbildung 9.17</u> zu sehen.



Testergebnis - Details

	Testfall		Ergebnis				
	Testanforderung(en)	Dauer [sec]	Testlauf				
TC00015 - Fü	gen Sie einen Dauerauftrag hinzu.		Bestanden				
	Es ist erforderlich, dass sich die Datenbank immer in einem konsistenten Zustand	8,66	P00005-TRU0005422				
	Testschritt		Ergebnis				
	Starten Sie die Anwendung durch Drücken der blauen Symbolschaltfläche .						
	Wählen Sie die Taste , die sich im Hauptmenü befindet.		Bestanden				
	Wählen die die Taste .		Bestanden				
	Geben sie folgene Informationen in die Felder ein: Begünstigter: Max Mustermann IBAN: DE19 1234 1234						
	Geben Sie die TAN in das TAN Feld ein und drücken Sie die Taste.		Bestanden				

Abbildung 9.18. Der Bericht "Testlauf" (Fortsetzung)

9.3.5. Detailseite

Jeder Testlauf verfügt über eine eigene Detailseite. Mit Klick auf die ID des betreffenden Testlaufs oder auf das Icon Q in der Aktionsspalte gelangen Sie zu der Detailseite. Hier werden alle Testfallergebnisse und das Testsuiteergebnis des Testlaufs - sofern vorhanden - in Tabellen angezeigt.

Es stehen die folgenden Ansichten zur Verfügung: *Eigenschaften, Benutzerdefiniert* und *Ergebnisse* se. Bei erstmaligem Aufruf ist dies die Ansicht *Ergebnisse* mit den Unteransichten *Testfallergebnisse* und *Testsuiteergebnisse*.

9.3.5.1. Aktionen

Die Aktionszeile befindet sich rechts oben in der Titelzeile. Auf den Detailseiten lassen sich jeweils die folgenden Aktionen vornehmen:

Ausführungsansicht öffnen	Diesen Testlauf erneut ausführen.				
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht der Testlauf-Eigen- schaften erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.				
	Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.				
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.				
	Das Anlegen von Lesezeichen ist ausführlich in <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> beschrieben.				
G O Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Eigenschaften gewech- selt werden.				

9.3.5.2. Eigenschaften

Auf der Ansicht Eigenschaften werden die folgenden Angaben dargestellt:

ID	Die automatisch zugewiesene ID.
Ausführender	Der Benutzer, der diesen Testlauf ausgeführt hat.
Testsystem	Das Testsystem, mit dem der Testlauf ausgeführt wurde.*
Testumgebung	Die Testumgebung, mit dem der Testlauf ausgeführt wurde.*
Iteration	Die Iteration, in der der Testlauf ausgeführt wurde.*
Zugewiesen	Der Benutzer, dem dieser Testlauf zugewiesen ist.

* Ein Testmanager oder Administrator kann diese Einträge hier nachträglich korrigieren. Dies kann z.B. bei einer falschen Eingabe beim Start des Testlaufs notwendig sein.

9.3.5.3. Benutzerdefiniert

Für individuelle Vorgaben lassen sich eigene Felder anlegen. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.4</u>, <u>"Benutzerdefinierte Eigenschaften"</u>.

9.3.5.4. Ergebnisse

Die Ansicht *Ergebnisse* teilt sich in die Unteransichten *Testfallergebnisse* und *Testsuiteergebnisse* auf. In der Tabelle werden die folgenden Werte dargestellt:

ID	Die automatisch zugewiesene ID.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Start	Der Startzeitpunkt des Testlaufs.
Iteration	Die Iteration, in der der Testlauf ausgeführt wurde.
Test Case	Der Testfall.
Testumgebung	Die Testumgebung, in der der Test ausgeführt wurde.
Testsystem	Das Testsystem, das getestet wurde.
Ausgeführt von	Der Name des Benutzers, der den Testlauf zuletzt ausgeführt hat.
Dauer	Die Ausführungszeit.
Ergebnis	Das Testfall-Ergebnis (Bestanden, Fehlschlag, Fehler, Unklar, Übersprungen).

Aktion

Die ausführbaren Aktionen.

= 👱 К L A R (o s test management	DE Finanz-Tracker 🗧 🗮	Q × Ø~ ≛×
📝 Definieren	Testlauf: TRU0006660 (SUT00006 - Finance Track	xer 2.1.0 / ENV00005 - Android 8 Smartphone) - 07.12.21, 14:14:4	n 🕹 🖨 🗋 😋 🕤
📇 Planen	Eigenschaften Benutzerdefiniert Ergebnisse (3)	Änderungen	Speichern Verwerfen Zurück
🏟 Ausführen	ID TRU0006660 Ausführender Felix Mustermann		
쓵 Auswerten	Testsystem Finance Tracker 2.1.0 Testumgebung Android 8 Smartphone	▼▼	
🔎 Konfigurieren	Iteration	\checkmark	
	Zugewiesen Felix Mustermann	~	
	Erstellt vor 2 Jahren von Felix Mustermann		Zuletzt geändert vor 2 Jahren von Felix Mustermann
			Speichern Verwerfen Zurück

Abbildung 9.19. Die Detailseite eines Testlaufs

9.3.6. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Testlaufs an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

9.4. Testfallergebnisse

Die Seite Testfallergebnisse zeigt eine Übersicht aller Testfälle mit der Anzahl der Ergebnisse Bestanden, Fehlschlag, Fehler, Unklar und Übersprungen an.

≡ 👱 K L A R	o s test mai	DE Fin	nanz-Tracker 🖻 🛢	Q × 0∕ × ⊥·	
🕑 Definieren	Testfallergebnisse				
🏩 Planen	+	24 Einträge - Seite 1 von 3 📕 ┥		品又	¢ ≛0
📩 Ausführen	□ TC00024 1.0	Name ₹ A Überprüfen Sie die Verbindung zum Server	Requirement 1.2.4 in testplan_16.txt	2/1 3	
Austunien	TC00023 1.0	🔀 Einen Dauerauftrag ändern.	Requirement 1.2.4 in testplan_16.txt	0/2 31	16 9 3 1 2 🔩 Q
e Auswerten	TC00022 1.0	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	5/0 25	11 11 2 1 🎝 Q
	TC00021 1.0	 Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags. 	Requirement 1.2.4 in testplan_16.txt	0/2 24	17 4 3 2 ¢Q
🔑 Konfigurieren	TC00020 1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Requirement 1.2.4 in testplan_16.txt	0/0 19	8 4 5 1 1 2
	TC00019 1.0	S Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/testi ng/web/Connection.java	0/0 3	1 1 1 J
	TC00018 1.0	3 Testen der Verbindung zum Server der Bank.	/FinanceTracker/src/test/java/de/sabrina/testi ng/web/Connection.java	0/0 3	3 - S Q
	TC00017 1.0	🛕 Lastschriftmandat löschen.	Requirement 1.2.4 in testplan_16.txt	0/0 31	12 7 4 8 🚑 Q
	TC00016 1.0	Ø Daueraufträge löschen.	Requirement 1.2.4 in testplan_16.txt	0/0 41	29 4 3 1 4 🔩 📿
	TC00015 1.0	Fügen Sie einen Dauerauftrag hinzu.	Requirement 1.2.4 in testplan_16.txt	0/0 49	34 9 3 2 1 🏖 🔍
	ID Revision				e Aktion

Abbildung 9.20. Die Seite "Testfallergebnisse"

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Revision	Die Revision des Testfalls.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Name	Der Name des Testfalls.
Verfolgbarkeit	Die Anforderung, die diesem Testfall zugeordnet ist.
Issues	Die Anzahl der offenen und geklärten Issues.
Ergebnisse	Die Anzahl der Testfall-Ergebnisse.
Bestanden	Die Anzahl der Testfall-Ergebnisse mit dem Status "Bestan- den".
Fehlschlag	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Fehlschlag".
Fehler	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Fehler".
Unklar	Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Unklar".

Übersprungen		Die Anzahl der Testfall-Ergebnisse mit dem Status/Testurteil "Übersprungen".
Aktion		Die ausführbaren Aktionen.
	Anmerkung	
	Das <i>Ergebnis</i> zeigt das	s jeweils letzte Ergebnis der Testfälle.

9.4.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Le Den Testfall erneut ausführen.

- Ergebnisse importieren.

Q Die Testfallergebnisse dieses Testfalls anzeigen (siehe <u>Abschnitt 9.4.4, "Ergebnisse eines</u> <u>einzelnen Testfalls"</u>).

9.4.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testfälle gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testfälle aus, auf die die Aktion angewendet werden soll.

Für Testfälle werden folgende Massenaktionen unterstützt:

+ Testfälle zu einer Testsuite hinzufügen (siehe <u>Abbildung 9.21</u>).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

≡ 👱 K L A R (OS TEST MA	A N A G E M E N T	DE F	Finanz-Tracker 🖬 🛢	Q × 0∕ × ×
📝 Definieren	Testfallergebnisse				
💐 Planen	+ 24 von 24 ausgewählt	24 Einträge - Seite 1 von 3		器 🏾 😐 🗢 💼	© Q × & Ξ
	🗹 ID 🗢 🛛 Revision			Issues 🗢 Ergebnisse 🗢 😔 🕯	♠ ♠ ♦ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
🏚 Ausführen	✓ TC00024 1.0	🛕 Überprüfen Sie die Verbindung zum Server	Requirement 1.2.4 in testplan_16.txt	2/1 3 1	2
	✓ TC00023 1.0	Einen Dauerauftrag ändern.	Requirement 1.2.4 in testplan_16.txt	0/2 31 16	9 3 1 2 🔩 🔍
e Auswerten	✓ TC00022 1.0	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	5/0 25 11	11 2 1 🏖 Q
-	✓ TC00021 1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Requirement 1.2.4 in testplan_16.txt	0/2 24 17	4 3 2 ¢Q
🔑 Konfigurieren	✓ TC00020 1.0	Aktualisieren Sie das Dashboard durch Stornierung eines Lastschriftmandats.	Requirement 1.2.4 in testplan_16.txt	0/0 19 8	4 5 1 1 2
	✓ TC00019 1.0	Testen der Verbindung zur Datenbank.	/FinanceTracker/src/test/java/de/sabrina/tes	sti 0/0 3	🚺 🚺 👖 占 🔍
	✓ TC00018 1.0	2 Testen d Die Testfälle einer (neuen) Testsuite	zuordnen	/0 3	📑 🛃 🛃 Q
	7000017 1 0	Die folgenden Testfälle wer	rden einer Testsuite hinzugefügt	())) () () () () () () () ()	
	✓ TC00017 1.0			70 31 12	
	✓ TC00016 1.0	Eine neue Testsuite anlegen		/0 41 29	
	D Revision	• Fugen Si		70 49 34	9 3 2 1 🚅 🗸
	ID HEVISION	Die Testfälle einer bestehend	en Testsuite zuweisen	Ligebhisse	Aktor
		Name			
			OK Abbreche	en	

Abbildung 9.21. Der Dialog "Testfälle zu Testsuite hinzufügen"

9.4.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige ausgeführte/alle Testfälle

Nur neueste Ergebnisse/Zeige alle

- Q Suchen
- 公 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

9.4.4. Ergebnisse eines einzelnen Testfalls

Die Seite "Testfallergebnisse" zeigt alle Ausführungen des ausgewählten Testfalls mit dem Ergebnis, dem verwendeten Testsystem sowie der verwendeten Testumgebung an (<u>Abbildung 9.22</u>).

≡	🞐 KLARO) S	TEST	МАМ	I A G E	MENT					DE Finanz-Tracker 🛛 🖻	88		Q X	0∼ ≗~
ľ	Definieren	Tes	stfallergebr	nisse: T	C00022	- Aktualisie	eren Sie das Das	shboard duro	h Hinzufü	gen eines Dauera	uftrags.				
*	Planen	ł	- Testfalldetai	ils											
۵	Ausführen	ð	ID 🖨	0 s	tart 🕏	25 E	Einträge - Seite 1 von 3	K ◀ 1 teration ♦	2 3	► N 10 ▼ Testumaebuna ♦	Testsystem 🖨	Ausgeführt von 🗢	Dauer 🗢	Q Ergebnis 🗘	× & ≡
¢	Auswerten		TCR0002497	₿ vor	4 Jahren	TRU0001757				Android 8 Smartphone	Finance Tracker 2.1.0	Selen Deutsch	00:01:28	*	₽ ₽ Q
-	_		TCR0002460	vor	4 Jahren	TRU0001730	2.1.x-Sprint 01-Opti auf Smartwatches	imierung für die	Verwendung	Android Tablet	Finance Tracker 2.1.0	Felix Mustermann	00:00:08	A	₽ ₽ Q
عر	Konfigurieren		TCR0002439	vor	4 Jahren	TRU0001725	2.0.x-Sprint 02-Das	hboard erweiter	n	Android 9 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:13	A	₽ \$⊖ Q
			TCR0002407	vor	4 Jahren	TRU0001718	1.1.x-Sprint 04-Opti	imierung für Tab	lets	Android 10 Smartphone	Finance Tracker 1.1.1	Felix Mustermann	00:00:06	A	₽ \$⊖ Q
			TCR0002380	vor	4 Jahren	TRU0001705	1.1.x-Sprint 03-Ann Einzugsermächtigu	ullierung von Ingen		Android 9 Smartphone	Finance Tracker 1.1.1	Felix Mustermann	00:00:08	0	₽ ⊖ Q
			TCR0002342	vor	4 Jahren	TRU0001697	1.1.x-Sprint 02-Dau	eraufträge bear	beiten	Android 10 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:12	0	₽ 🛱 Q
			TCR0002338	vor	4 Jahren	TRU0001696	1.1.x-Sprint 02-Dau	eraufträge bear	beiten	Android 10 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:09	0	₽ ⊖ Q
			TCR0002334	vor	4 Jahren	TRU0001695	1.1.x-Sprint 02-Dau	eraufträge bear	beiten	Android 9 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:09	0	₽ 🛱 Q
			TCR0002330	vor	4 Jahren	TRU0001694	1.1.x-Sprint 02-Dau	eraufträge bear	beiten	Android 8 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:09	0	₽ ₽ Q
			TCR0002297	vor	4 Jahren	TRU0001686	1.1.x-Sprint 01 - Üb	ersichten		Android 8 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:09	A	₽ 🛱 Q
			ID		Start	Testlauf		Iteration		Testumgebung	Testsystem	Ausgeführt von	Dauer	Ergebnis	Aktion
															Zurück
_															

Abbildung 9.22. Die Seite "Testfallergebnisse"

Das ausklappbare Feld oben auf der Seite zeigt die Details des ausgewählten Testfalls an. Darunter befindet sich eine Tabelle mit allen Ergebnissen.

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Start	Der Startzeitpunkt des Testlaufs.
Testlauf	Der Testlauf, zu dem das Ergebnis gehört.
Iteration	Die Iteration, in der der Testlauf ausgeführt wurde.
Testumgebung	Die Testumgebung, in der der Test ausgeführt wurde.
Testsystem	Das Testsystem, das getestet wurde.
Ausführender	Der Name des Benutzers, der den Testlauf zuletzt ausgeführt hat.
Dauer	Die Ausführungszeit.
Ergebnis	Das Testfall-Ergebnis (Bestanden, Fehlschlag, Fehler, Unklar oder Übersprungen).
Aktion	Die ausführbaren Aktionen.

Das Icon **9** in der Spalte *Zusätzliche Hinweise* zeigt an, ob das entsprechende Testfallergebnis von einem Benutzer manuell geändert wurde.

Ein Klick auf den Button Zurück führt zurück zur Seite Testfallergebnisse (Abbildung 9.20).

9.4.4.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Le Diesen Testfall erneut ausführen.

Ergebnisse importieren.

Q Die Detailseite des Testfallergebnisses anzeigen (siehe <u>Abschnitt 9.4.4.4, "Die Detailseite</u> <u>"Testfallergebnis""</u>)

9.4.4.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testfall-Ergebnisse gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testfall-Ergebnisse aus, auf die die Aktion angewendet werden soll.

Für Testfall-Ergebnisse werden folgende Massenaktionen unterstützt:

🔒 Druckansicht öffnen

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

9.4.4.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

∀ Filter / Sortieren

Nur neueste Ergebnisse/Zeige alle

- Q Suchen
- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

9.4.4.4. Die Detailseite "Testfallergebnis"

Diese Seite zeigt ein einzelnes Testfallergebnis und dessen Testschrittergebnisse an.

≡	y KLAR	os test m	ANAGEMENT	DE Finanz-Tracker 👕 🗮	વ× છ~ ≛~
ľ	Definieren	🔀 🎧 Testfaller	gebnis: TCR0002497 (TC00022 - Aktualisieren Sie das Dashboard	l durch Hinzufügen eines Dauerauftrags.)	♣; 🖨 🗋 😋 😜
***	Planen	Eigenschaften	Benutzerdefiniert Anhänge Issues (5) Änderungen	Speichern Verwei	rfen Zurück
\$	Ausführen	Zusammenfassung	Die Taste konnte nicht geklickt werden.		
¢	Auswerten	Beschreibung	En kapata kain Daugauftan himugafiikt wardan Dashalk kapata dia Taatawifiika	ung nicht mahr ausgeführt worden	
ų	Konfigurieren	Desencebung	es konne ken baueraurrag inizugerant werden. Desnab konne die restausram	ung mont meni ausgerum t werden.	
		Ergebnis Start Ausgeführt von Ausführungsdauer Testlauf	V Unklar 18.08.20, 14:43:30 Selen Deutsch 00:01:28 TRU0001757 - 18.08.20, 14:43:30		
		Testfall Testumgebung Testsystem	1.0 - TC00022, Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauft ENV00005 - Android 8 Smartphone SUT00006 - Finance Tracker 2.1.0	rags.	
		Schritte	e Schritt	Zucommonfoccung Bocobroibung	Davor Aktion
		1 🖉 🗸 Besta	Schnitt S	zusammentassung eeschreibung	00:00:06
		2 🖉 🗸 Besta	anden Aktion Erwartetes Ergebnia Vorbedingung Nachbedingung Wählen Sie die Taste Dauerauftrag , die sich im Hauptmenü befindet.		00:00:01 🥔
		3 🔀 🗸 V	klar Aktion Erwartetes Ergebnis Vorbedingung Nachbedingung	Die Taste konnte nicht Es konnte kein Dauerauftrag hinz geklickt werden. werden. Deshalb konnte die	zugefüht 00:01:13 🖉

Abbildung 9.23. Die Seite "Testfallergebnis"

9.4.4.4.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

2. Diesen Testfall / diese Test- Führt den Testfall oder die Testsuite erneut aus. suite ausführen

- Diesen Testfall / diese Test- suite ausführen	Führt den Testfall oder die Testsuite erneut aus.				
🔒 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Testfall-Ergebnisses erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.				
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.1, "Druckansichten"</u> .				
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.				
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> .				
G 🕤 Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testfallergebnissen gewechselt werden.				

Die Zusammenfassung, Beschreibung und das Ergebnis des Testfallergebnisses und jedes einzelnen Schrittergebnisses kann hier bearbeitet werden. Alle Änderungen werden automatisch protokolliert und jede Änderung eines Ergebnisses führt zu einer Neuberechnung des entsprechenden Testfallergebnisses, des Testsuiteergebnisses und des Testlaufs. Wird ein Testfallergebnis geändert, wird das Icon ? auf dieser Seite und in allen Tabellen, in denen das Ergebnis angezeigt wird, angezeigt. Das Ergebnis Feld eines Testfallergebnisses oder Testschrittergebnisses kann nur in Klaros-Testmanagement Enterprise Edition geändert werden.

≡ 📌 к L А	ROS TE	EST MANAGE	MENT		DE Finanz-Tracker 🧧 🗮	० छ~ 				
📝 Definieren	🔀 🖓 T	Festfallergebnis: TCR0	0002497 (TC00022 - Aktualisieren Sie da	s Dashboard durch Hinzu	fügen eines Dauerauftrags.)	🛃 🖓 🚭 🕰				
🏝 Planen	Eigensch	naften Benutzerdefinie	ert Anhänge Issues(5) Änderungen		Speichern	Verwerfen Zurück				
🌣 Ausführen	Zusamm	Zusammenfassung Die Taste konnte nicht geklickt werden.								
븑 Auswerten	Be	Beschreibung Es konnte kein Dauerauftrag hinzugefüht werden. Deshalb konnte die Testausführung nicht mehr ausgeführt werden.								
✗ Konfigurieren	Infigurieren Ergebnis 24 ♥ Unklar Start 18.08 20, 14/43:30 Ausgeführt von Selen Deutsch Ausführungsdauer 00:01:28 Testlagt TRU0001757 - 18.08.20, 14/43:30 Testfall 1.0. * TC00022, Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags. Testumgebung ENV00005 - Android 8 Smartphone Testsystem SUT00006 - Finance Tracker 2.1.0 Scheitte									
	#	Ergebnis	Schritt	Zusammenfassung	Beschreibung	Dauer Aktion				
	1 0	 Bestanden Aktion Bestanden Fehlschlag Fehler 	In Erwartetes Ergebnis Vorbedingung Benutzerdefiniert Sie die Anwendung durch das <i>drücken</i> der bla symbolschaltfläche.			00:00:06 🥳				
	2 🥥	🗙 Unklar	Erwartetes Ergebnis Vorbedingung			00:00:01 🥝				

Abbildung 9.24. Ein Testschrittergebnis bearbeiten

Druckoptionen Leere Einträge ausblenden 🗸							
Details	Vollständig Kurz 🖓						
Testfallergebnis	TC00005: Kontostand prüfen.						
Eigenschaften							
Ergebnis	A Fehlschlag: Schaltfläche "Kontostand" inaktiv.						
Zusammenfassung	Schaltfläche "Kontostand" inaktiv.						
Start	04.08.21, 15:43:41						
Ausgeführt von	Felix Mustermann						
Ausführungsdauer	00:03:11						
Testlauf	TRU0005362 - 04.08.21, 15:43:40						
Testfall	TC00005 - Kontostand prüfen.						
Testumgebung	ENVUUUU2 - Android Tablet						
restsystem	SUTUDUUO - Filialite Hackel 2.1.0						
Schritte							
Schritt	1/2 🥑 Bestanden						
Aktion	Starten Sie die Anwendung durch Drücken der blauen Symbolschaltfläche						
Erwartetes Ergebnis	Sie gelangen auf das Startbild.						
Vorbedingung	Die Anwendung ist auf dem Gerät installiert. Wenn nicht, folgen Sie den Anweisungen: https://finanztracker/install						
Nachbedingung	Die Anwendung ist erfolgreich gestartet.						
Schritt	2/2 🔥 Fehischlag						
Beschreibung	Schaltfläche "Kontostand" ist ausgegraut (deaktiviert).						
Zusammenfassung	Schaltfläche "Kontostand" nicht klickbar						
Aktion	Wählen Sie die Schaltfläche Kontostand, die sich im Hauptmenü befindet.						
Erwartetes Ergebnis	Der Kontostand wird angezeigt.						

Abbildung 9.25. Die Druckansicht "Testfallergebnis"

9.4.4.5. Gründe für das Überspringen von Testfällen

≡ 🖞 K L A R	o s test management	DE Finanz-Tracker 🖬 📑	Q 0 × ≞ ×
📝 Definieren	Testlauf: TRU0004560 (SUT00006 - Finance Tracker 2.1.0 / ENV00005 -	Android 8 Smartphone) - 26.03.21, 11:24:50	🎝 🖓 🖓 🖓
北 Planen	1 Einträge - Seite 1 von 1 📕 🚽 📘	₩ 10 🗸	Q × & ≡
🌣 Ausführen	ID ÷ V Start ÷ Testfall ¢ Iteration ÷ Testumgebun TCR0006336 9 26.03.21, 11:24:50 TC00021 Android 8 Smartphor	g ← Testsystem ← Ausgefuhrt von ← Dauer ← e Finance Tracker 2.1.0 Selen Deutsch 00:00:18	Ergebnis ÷ Details Image: Construction of the second secon
쓵 Auswerten	Aus folgenden Gründen übersprungen: - Server ist nicht erreichbar.		Zurück
🖌 Konfigurieren			

Abbildung 9.26. Begründung für das Überspringen eines Testfalls anzeigen

Wird ein Testfall übersprungen, kann der Tester einen oder mehrere Begründungen aus vordefinierten Vorlagen eingeben (siehe <u>Abschnitt 10.4.3, "Testausführung"</u> und <u>Abschnitt 8.3.3.1.1,</u> <u>"Testfälle endgültig überspringen"</u>). Das Icon ③ erscheint auf den entsprechenden Seiten im Bereich *Auswerten*. Befindet sich der Mauszeiger über dem Icon, werden die vordefinierten Begründungen als Tooltip angezeigt.

9.5. Testsuiteergebnisse

Die Seite *Testsuiteergebnisse* zeigt eine Übersicht aller Testsuiten inklusive der Anzahl der Ergebnisse an.

≡	Y KLAR (DS TEST M.	ANAGEMENT	DE Finanz-Tracker 📑	≣Q× 2	~							
ľ	Definieren	Testsuiteergebnisse											
:2:	Planen	ID 🖨 Bevision 🕯	7 Einträge - Seite 1 von 1 🕅 🚽 1 🕨 🕅 10 💌	Testfälle 🗢 Fr	RT⊕‡≥0 Q × &								
\$	Ausführen	TS00008 1.0 C TS00007 1.0 >	Überprüfen Sie die Verbindung zum Server Daueraufträge bearbeiten	1		えん							
¢	Auswerten	TS00006 1.0 C TS00004 1.0 C TS00003 1.0 4	Übersicht Verbingungen überprüfen Activitäten verschiedener Zeiträume anzeigen	7 2 4	19 11 4 4 2 2 5 C	えんれてい							
ŗ	Konfigurieren	TS00002 1.0 TS00001 1.0	Öberweisungen testen Dashboard aktualisieren	8 4		ええ							
		ID REVISION	beschreibung	Testialle	-igebnisse Aktio								

Abbildung 9.27. Die Seite "Testsuiteergebnisse"

Die Tabelle zeigt die folgenden Werte:

ID	Die automatisch zugewiesene ID.
Revision	Die Revision des Testsuite.
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Beschreibung	Die Beschreibung der Testsuite.
Testfälle	Die Anzahl der Testfälle.
Ergebnisse	Die Anzahl der Testsuite-Ergebnisse.
Aktion	Die ausführbaren Aktionen.

9.5.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Le Testsuite erneut ausführen.

Ergebnisse importieren.

Q Die Testsuiteergebnisse dieser Testsuite anzeigen (siehe <u>Abschnitt 9.5.3, "Ergebnisse einer</u> <u>einzelnen Testsuite"</u>).

fi Löschen

9.5.2. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

Filter / Sortieren

Nur ausgeführte Testsuiten/Zeige alle Testsuiten

Q Suchen

- 凸 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

9.5.3. Ergebnisse einer einzelnen Testsuite

Die Seite "Testsuiteergebnisse" zeigt alle Ausführungen der ausgewählten Testsuite mit dem Ergebnis, dem verwendeten Testsystem sowie der verwendeten Testumgebung an (<u>Abbildung</u> <u>9.28</u>).

≡ 👱 K L A R (OS TEST				DE Finanz-Tracker 📑	■		Q X	Ø~ ≗ ~
Definieren	Testsuiteerge	bnisse: TSC	00007 - Daueraufträge bearbeiten						
北 Planen		Start A	21 Einträge - Seite 1 von 3 🖌 ◀		Tasteustan		Davas 🌢	Q	× & ≡
🅸 Ausführen	TSR0001208	vor 2 Jahren	iteration 👻	ENV00005 - Android 8 Smartphone	Finance Tracker 2.1.0	Felix Mustermann	00:00:26	× ×	2 0 € Q
	TSR0000304	vor 4 Jahren	2.0.x-Sprint 01-Dashboard hinzufügen	ENV00001 - Android 10 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:15	0	₽ ₽ 0
e Auswerten	TSR0000303	vor 4 Jahren	2.0.x-Sprint 01-Dashboard hinzufügen	ENV00002 - Android Tablet	Finance Tracker 2.0.0	Felix Mustermann	00:00:17	0	₽ ⊕ Q
🔑 Konfigurieren	TSR0000302	vor 4 Jahren	2.0.x-Sprint 01-Dashboard hinzufügen	ENV00004 - Android 9 Smartphone	Finance Tracker 2.0.0	Felix Mustermann	00:00:17	8	₽ ₽ Q
	TSR0000298	vor 4 Jahren	1.1.x-Sprint 04-Optimierung für Tablets	ENV00002 - Android Tablet	Finance Tracker 1.1.1	Felix Mustermann	00:00:15	0	₽ 🛱 Q
	TSR0000295	i vor 4 Jahren	1.1.x-Sprint 03-Annullierung von Einzugsermächtigungen	ENV00005 - Android 8 Smartphone	Finance Tracker 1.1.1	Felix Mustermann	00:00:20	A	₽ ₽ Q
	TSR0000287	vor 4 Jahren	1.1.x-Sprint 01 - Übersichten	ENV00001 - Android 10 Smartphone	Finance Tracker 1.1.0	Felix Mustermann	00:00:16	A	₽ ₽ Q
	TSR0000283	vor 4 Jahren	1.0.x-Sprint 02-Erweiterung der Grundfunktionen	ENV00004 - Android 9 Smartphone	Finance Tracker 1.0.0	Felix Mustermann	00:00:28	0	₽ ₽ 0
	TSR0000223	vor 4 Jahren		ENV00005 - Android 8 Smartphone	Finance Tracker 2.1.0	Felix Mustermann	00:00:13	0	₽ ₽ Q
	TSR0000222	vor 4 Jahren		ENV00005 - Android 8 Smartphone	Finance Tracker 2.1.0	Felix Mustermann	00:00:20	A	₽ ₽ Q
	ID	Start	Iteration	Testumgebung	Testsystem	Ausgeführt von	Dauer	Ergebnis	Aktion
									Zurück

Abbildung 9.28. Die Seite "Testsuiteergebnisse" für eine Testsuite

Das ausklappbare Feld oben auf der Seite zeigt die Details der ausgewählten Testsuite an. Darunter befindet sich eine Tabelle mit allen Ergebnissen.

Die Tabelle zeigt die folgenden Werte:

Die automatisch zugewiesene ID.

Start	Der Startzeitpunkt des Testlaufs.
Testumgebung	Die Testumgebung, in der der Test ausgeführt wurde.
Testsystem	Das Testsystem, das getestet wurde.
Ausführender	Der Name des Benutzers, der den Testlauf zuletzt ausgeführt hat.
Dauer	Die Ausführungszeit.
Ergebnis	Das Testfall-Ergebnis (Bestanden, Fehlschlag, Fehler, Unklar oder Übersprungen).
Aktion	Die ausführbaren Aktionen.

9.5.3.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Les Diese Testsuite erneut ausführen.

Ergebnisse importieren.

Q Die Detailseite des Testsuite-Ergebnisses anzeigen (siehe <u>Abschnitt 9.5.3.4, "Die Detailseite</u> <u>"Testsuiteergebnis""</u>)

ff Löschen

9.5.3.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Testfall-Ergebnisse gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Testfall-Ergebnisse aus, auf die die Aktion angewendet werden soll.

Für Testsuite-Ergebnisse werden folgende Massenaktionen unterstützt:

- 🔒 Druckansicht öffnen
- 前 Löschen

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

9.5.3.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- √ Filter / Sortieren
- Q Suchen
- 公 Exportieren

\equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

9.5.3.4. Die Detailseite "Testsuiteergebnis"

Diese Seite zeigt ein einzelnes Testsuiteergebnis und dessen Testfallergebnisse an.

≡ 👱 K L A R	OS TEST						DE Finanz-Tracker 🗧	8		Q X (∂ ~ ≗ ~
🕑 Definieren	🔀 Testsuite	ergebnis: TS	R0001208 (TS00007 - Dauer	aufträge bea	arbeiten)					.	
📇 Planen									_ /		
-			3 Einträge - Seite 1	von 1 🕅 🖣		10 🗸		V	7	Q	×ಔ≡
🔅 Ausführen	ID ≑ 💱	Start 🗢	Testfall 🗢	Testlauf 🗢	Iteration 🗢	Testumgebung 🗢	Testsystem 🗢	Ausgeführt von 🗢	Dauer 🗧 E	rgebnis 🗢	Aktion
	TCR0009177	vor 2 Janren	Dauerauttrage loschen.	1800006660		Smartphone	2.1.0	Felix Mustermann	00:00:11	24	
Auswerten	TCR0009176	vor 2 Jahren	Einen Dauerauftrag ändern.	TRU0006660		Android 8 Smartphone	Finance Tracker 2.1.0	Felix Mustermann	00:00:07	24	₽ ₽ Q
🖌 Konfigurieren	TCR0009175	vor 2 Jahren	Fügen Sie einen Dauerauftrag hinzu.	TRU0006660		Android 8 Smartphone	Finance Tracker 2.1.0	Felix Mustermann	00:00:08	*	₽ ₽ Q
	ID	Start	Testfall	Testlauf	Iteration	Testumgebung	Testsystem	Ausgeführt von	Dauer	Ergebnis	Aktion
	Erstellt vor 2 Jah	ren von Felix Mu	stermann					Zuletzt geän	dert vor 2 Jah	ren von Felix	Zurück Mustermann

Abbildung 9.29. Die Detailseite "Testsuiteergebnis"

9.5.3.4.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

Diese Testsuite ausführen	Führt die Testsuite erneut aus.
- Diese Testsuite ausführen	Führt die Testsuite erneut aus.
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Testsuite-Ergeb- nisses erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.1, "Druckansichten"</u> .
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> .
G 🕤 Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Testsuite-Ergebnissen gewechselt werden.

Druckoptionen Leere Einträge ausblenden Details Vollständig Kurz Orucken					
Testsuiteerg	gebnis TSR0000986: Server nicht erreichbar				
Zusammenfas Beschreibung Ergebnis:	ssung: Server nicht erreichbar ; Sehler (Restanden: 0. Eehlechlag: 0. Eehler: 2. Übersprun	nen: 0)			
Teetfellerrehni	ince	gen. oj			
Testrallergebni	ISSE				
Ergebnis:	8 Fehler: Datenbank nicht verfügbar				
ID:	: TCR0007444				
Start:	: 04.08.21, 16:32:06				
Testfall:	: 1.0 - TC00018, Testen der Verbindung zum Server der Bank.				
Testlauf:	: TRU0005364	TRU0005364			
Testumgebung:	ENV00005 - Android 8 Smartphone				
Testsystem:	SUT00006 - Finance Tracker 2.1.0				
Ausgeführt von:	Felix Mustermann				
Ergebnis: ID: Start: Testfall: Testlauf: Testumgebung: Testsystem: Ausgeführt von:	 Fehler: Verbindungsaufbau zum Server fehlgeschlagen TCR0007443 04.08.21, 16:32:06 1.0 - TC00019, Testen der Verbindung zur Datenbank. TRU0005364 ENV00005 - Android 8 Smartphone SUT00006 - Finance Tracker 2.1.0 Felix Mustermann 				

Abbildung 9.30. Die Druckansicht "Testsuiteergebnis"

9.6. Issues

Issues aus externen Issue-Management-Systemen (wie Jira oder GitLab) lassen sich mit Testfällen und Testsystemen in Klaros-Testmanagement verknüpfen. Ebenso lassen sich Issues von Klaros-Testmanagement aus in angebundenen Issue-Management-Systemen erstellen (auch während der Testausführung).

Dazu muss mindestens ein Issue-Management-System konfiguriert und dem gewünschten Projekt zugeordnet werden. Die Konfiguration eines Issue-Management-Systems ist in <u>Abschnitt 10.5.1, "Issue Management"</u> beschrieben, <u>Abschnitt 6.1.2.6.1, "Issue Management"</u> zeigt wie das Issue-Management-System einem Projekt zugeordnet wird.

9.6.1. Übersichtsseite

Die Übersichtsseite zeigt alle im Projekt vorhandenen Issues in einer Tabelle an. In diesem Kapitel wird gezeigt, wie Issues erstellt, verlinkt und bearbeitet werden.

Ein Klick auf den Button Neu öffnet die Seite Abschnitt 9.6.6, "Issue - Details".

Ein Klick auf den Button Verknüpfen öffnet das Issues verknüpfen Panel, in dem vorhandene Issues aus Issue-Management-Systemen mit Testfällen und Testsystemen verknüpft werden können (siehe <u>Abschnitt 9.6.7, "Einen Issue verlinken"</u>).

≡ 👱 к L A R (o s test manageme	E N T		DE Finanz-Tracker 📑		۹	× 0× ±×
📝 Definieren	Issues						C
💐 Planen	Neu Verknüpfen					Speichern	Verwerfen
_	₽ û	12 Einträge - Seite 1 von 2	M 🚽 🚺 2 🕨 M 10 🗸		Alle anzeigen		Q × & ≡
📩 Δusführen	🔲 ID 🗢 🛛 🖓 System 🗢						Aktion
	PLAYGROUND-25702 🙀 IM00008	test Issuevtvttvtv	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Finance Tracker 2.0.1	admin Blocker	Open	☞ ८ 🖨 û
e Auswerten	PLAYGROUND-25702 👻 IM00008	test Issuevtvttvtv	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Finance Tracker 1.1.1	admin Blocker	Open	┏ ┎ ᇢ @
🔑 Konfigurieren	PLAYGROUND-25702 🙀 IM00008	test Issuevtvttvtv	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.	Finance Tracker 2.1.0	admin Blocker	Open	☞ ☞ 흩 莭
	PLAYGROUND-25702 🙀 IM00008	test Issuevtvttvtv	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.		admin Blocker	Open	┏┎┎╘╓
	PLAYGROUND-25701 🙀 IM00008	test Issue	Aktualisieren Sie das Dashboard durch Hinzufügen eines Dauerauftrags.		gidley Blocker	Open	┏┎╘р
	PLAYGROUND-25700 🙀 IM00008	test Issue	🛕 Überprüfen Sie die Verbindung zum Server		admin Critical	Open	☞ 匠 🖨 🛍
	PLAYGROUND-25699 🙀 IM00008	test Issue	🛕 Überprüfen Sie die Verbindung zum Server		admin Blocker	Open	┏ ┖ 🖨 🛍
	PLAYGROUND-22965 ¥ IM00008	Closed Issue for Test Data	🛕 Überprüfen Sie die Verbindung zum Server		hudson Major	Closed	┏ ┖ 🖨 🛍
	PLAYGROUND-22965 V IM00008	Closed Issue for Test Data	Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Finance Tracker 2.1.0	hudson Major	Closed	☞ ८ 🖨 û
	PLAYGROUND-22963 ¥ IM00008	Closed Issue for Test Data	🔀 Einen Dauerauftrag ändern.		hudson Major	Closed	┏ ┖ 🖨 🛍
	ID System	Zusammenfassung	Testfälle	Testsysteme	Zuständig Prioriti	at Status	Aktion
	Neu Verknüpfen					Speichern	Verwerfen

Abbildung 9.31. Die Übersichtsseite "Issues"

Die Tabelle zeigt die folgenden Werte:

ID	Die externe Issue Referenz.
Zusätzliche Hinweise	Hier wird zum Einen das Symbol des Issue-Managament-Systems angezeigt. Ein Klick auf das Symbol öffnet das Issue im Issue-Management-System. Zum Anderen kann hier auch ein \triangle erscheinen. In diesem Fall sollte sofort der Administrator informiert werden. Ursachen für das Anzeigen des \triangle können unter Anderem sein:
	Deaktivierung des Issue-Management-Systems
	 Fehlerhafte Konfiguration des Issue-Management-Systems in Klaros-Testmanagement oder im Issue-Management-System. Die Konfiguration eines Issue-Management-Systems ist in <u>Abschnitt 10.5.1, "Issue Management"</u> beschrieben. Ein Tooltip erscheint, wenn der Cursor über dem hier angezeigten Icon platziert wird.
System	Die ID des Issue-Management-Systems.
Zusammenfassung	Die Zusammenfassung des Issues.
Testfälle	Eine Liste der verknüpften Testfälle mit dem Ergebnis ihrer letzten Ausführung.
Testsysteme	Eine Liste der verknüpften Testsysteme.
Zuständig	Der Benutzername des zuständigen Benutzers.
Priorität	Die Priorität des Issues.
Status	Der Status des Issues.

Aktion

Die ausführbaren Aktionen.

9.6.2. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- 🗹 Issue im externen Issue-Managment-System öffnen
- 🔒 Druckansicht öffnen
- 🗊 Löschen

Wurde ein Issue gelöscht, ist es zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Issues stehen dann statt *Löschen* die folgenden Aktionen zur Verfügung:

- Wiederherstellen
- ♦ Das Issue aus der Datenbank entfernen



Wichtig

Ein Klick auf Δ löscht nicht das Issue im Issue-Management-System, sondern es wird nur aus der Datenbank entfernt!

9.6.3. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Issues gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Issues aus, auf die die Aktion angewendet werden soll.

Für Issues werden folgende Massenaktionen unterstützt:

- 🔒 Druckansicht öffnen
- 🗊 Löschen
- Wiederherstellen des Issues (nur Administrator).
- ⊘ Das Issue aus der Datenbank entfernen (nur Administrator).

Weitere Informationen hierzu finden Sie unter Abschnitt 5.2.3.1.5, "Massenaktionen".

9.6.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

Q Suchen

- 요 Exportieren
- Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

9.6.5. Detailseite

Jedes Issue verfügt über eine eigene Detailseite mit weitergehenden Unteransichten. Mit Klick auf die ID des betreffenden Issues oder auf das Icon
rechts in der Aktionsspalte gelangen Sie zu der als letztes ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht *Eigenschaften*.

Es stehen die folgenden Ansichten zur Verfügung: Eigenschaften, Testfälle, Testsysteme und Änderungen.

9.6.5.1. Aktionen

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

🗹 Externen Link öffnen	Öffnet das Issue im externen Issue-Management-System.
Aktualisieren	Synchronisiert das lokale Issue mit der externen Quelle.
읍 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Issues erstellt werden. Mit Klick auf das Icon 🖨 öffnet sich diese in einem neuen Browsertab.
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.1, "Druckansichten"</u> .
☐ Lesezeichen anlegen	Jede einzelne Detailseite kann auch direkt über einen Hyper- link erreicht werden. Mit Klick auf das Icon Д wird dieser Link in die Zwischenablage kopiert.
	Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.2, "Lesezeichen"</u> .
G D Blättern	Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Issues gewechselt wer- den.

9.6.5.2. Eigenschaften

In dieser Ansicht (<u>Abbildung 9.32</u>) werden verschiedene Attribute des Issues angezeigt und können bearbeitet werden, wenn der Benutzer die entsprechende Berechtigung hat.

≡ 🞐 K L A R C) s test manag	EMENT	DE Finanz-Tracker 📓 🗮	५ × 0 × ≛×
📝 Definieren	PLAYGROUND-22965 - Clo	osed Issue for Test Data		◩◸◓▯◷◓
📇 Planen	Eigenschaften Testfälle (1) Testsysteme (1) Änderungen	Speichern	Verwerfen Zurück
🏟 Ausführen	Issue Management System	IM00008 - Jira (legacy), PLAYGROUND, Klaros 🔀 📧 (hudson)		
쓵 Auswerten	ID Status	PLAYGROUND-22965 Closed		X
G VanGauniana	Zusammenfassung *	Closed Issue for Test Data		
Konligurieren	Vorgangstyp	Bug 🗸		
	Priorität	Major 🗸		
	Komponente(n)	~		
	Betroffene Version(en)	~		
	Lösungsversion(en)	~		
	Zugewiesen			
	rang			
	Beschreibung	Open and Closed Issue Test Cases		
		(3a)		
		TC00078; TC00081; TC00084; TC00089; TC00094; TC00097;		
	Testfall Links hinzufügen			
	Testumgebung			

Abbildung 9.32. Die Ansicht "Eigenschaften"

Die Issue- ID ist ein Hyperlink zum entsprechenden Issue im Issue-Management-System.

9.6.5.3. Testfälle

In dieser Ansicht werden alle Testfälle aufgelistet, die mit diesem Issue verknüpft sind.

\equiv 👱 KLAR	o s test management	DE Finanz-Tracker 🖬 🗮 🔍 🔍 🕹 🗸
📝 Definieren	PLAYGROUND-22965 - Closed Issue for Test Data	년 2 음 지 3 9
🏩 Planen	Eigenschaften Testfälle (1) Testsysteme (1) Änderungen	Speichern Verwerfen Zurück
🏟 Ausführen	- 1 Einträge - Seite 1 von 1 🕅 ┥ 🧻 🕨 🕅 🔟 🗸	Λ σ × φ ≡
욙 Auswerten	ID € Revision € Name € TC00021 1.0 € Aktualisieren Sie das Dashboard durch Stornierung eines Dauerauftrags.	Aktion
🖌 Konfigurieren	Erstellt vor 8 Jahren von Felix Mustermann	Zuletzt geändert vor 26 Tagen von System Account
	Zuweisen	Speichern Verwerfen Zurück

Abbildung 9.33. Die Ansicht "Testfälle"

Mit Klick auf den Button Zuweisen werden Testfälle einem Issue zugewiesen. Es öffnet sich ein Dialog, in dem mehrere Testfälle auf einmal ausgewählt werden können.

= 📌 KLAR) s test manageme	NT	DE Finanz	-Tracker 🖬 🗮		Q × 0 × ≗×
📝 Definieren	PLAYGROUND-22965 - Closed Issue for Test Data					ưୁୁ⇔ୁୁୁୁୁ
				St	eichern \	/erwerfen Zurück
🕾 Planen	Eigenschaften Testfälle (1) Test	systeme (1) Änderungen				Lardon
Auoführen						
Ausiunien		1 Einträge - Seite 1 von 1 📕 🖣 🔽 🕨	Name a		Y	Q × 战 ≡
, Auswerten	TC00021 1.0 O Aktualisiere	n Sie das Dashboard durch Stornierung eines Dauerauftrag	js.			-
	ID Revision		Name			Aktion
差 Konfigurieren	Ers Sind Sie sicher?				tv	wa einem Jahr von System Account
	Die ausgewählten Testfälle we	rden diesem Issue hinzugefügt			1	/enverfen Zurück
						Zuluck
	23 Einträge -	Seite 1 von 3 🕅 ┥ 1 2 3 🕨 🕅 10 🗸	7	Q	× & ≡	
	■ ID Revision	Name 🕈	;			
	TC00024 1.0 Überprüfen S	Sie die Verbindung zum Server				
	TC00023 1.0 Einen Dauer	auftrag ändern.				
	TC00022 1.0 Aktualisierer	n Sie das Dashboard durch Hinzufügen eines Dauerauftrag	S.			
	TC00020 1.0 Aktualisierer	n Sie das Dashboard durch Stornierung eines Lastschriftma	andats.			
	TC00019 1.0 Testen der V	/erbindung zur Datenbank.				
	TC00018 1.0 Testen der V	/erbindung zum Server der Bank.				
	TC00017 1.0 Lastschriftm	nandat löschen.				
	TC00016 1.0 Daueraufträg	ge löschen.				
	TC00015 1.0 Fügen Sie ei	nen Dauerauftrag hinzu.				
	TC00014 1.0 Aktualisierur	ng des Dashboards nach einer Banküberweisung.				
				Zuweisen	Abbrechen	

Abbildung 9.34. Der Dialog "Testfälle zuordnen"

Für Testfälle werden folgende Massenaktionen unterstützt:

Entfernen

Die Detailseite eines Testfalls zeigt eine Liste aller verknüpften Issues. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 6.7.2.8, "Issues"</u>.

9.6.5.4. Testsysteme

Die Seite listet alle dem Issue zugeordneten Testsysteme auf.

Möchten Sie die Testsysteme mit dem Issue verknüpfen, klicken Sie auf den Button Zuweisen und wählen Sie ein oder mehrere Testsysteme aus.

Für Testsysteme werden folgende Massenaktionen unterstützt:

– Entfernen

Die Detailseite eines Testsystems zeigt eine Liste aller verknüpften Issues. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 6.5.2.6, "Issues"</u>.

≡ 👱 K L A R ()S TEST MANAGEMENT	DE Finanz-Tracker 🖬 🗮 🔍 🔍 🔍 🕹 🗸
📝 Definieren	PLAYGROUND-22965 - Closed Issue for Test Data	C C ⊖ Q
😤 Planen	Eigenschaften Testfälle (1) Testsysteme (1) Änderungen	Speichern Verwerfen Zurück
🏟 Ausführen	- 1 Einträge - Seite 1 von 1 🕅 🖣 1 🕨 🕅 10 🗸	Υ <u></u>
쓵 Auswerten	Version Ver	Aktion
🗲 Konfigurieren	Erstellt vor 8 Jahren von Felix Mustermann	Aktion Zuletzt geändert vor 26 Tagen von System Account
	Zuweisen	Speichern Verwerfen Zurück



≡ 🞐 K L A R C)S TEST MANAGEMENT	DE Finanz-Tracker 🖬 🛢	५ × 0 × ≜×
🕑 Definieren	PLAYGROUND-22965 - Closed Issue for Test Data		◩◸◓▯◷◓
🏖 Planen	Eigenschaften Testfälle (1) Testsysteme (1) Änderungen	Speichern	Verwerfen Zurück
🏟 Ausführen	- 1 Einträge - Seite 1 von 1 🕅 ┥ 🚺 🕨 🕅 🔟 💌	∇	Q × & Ξ
욙 Auswerten	UD Version Version Version SUT00006 Finance Tracker 2:1.0		Aktion
🗲 Konfigurieren	Sind Sie sicher? Ole ausgewählten Testsysteme werden diesem Issue hinzugefügt		t vor 26 Tagen von System Account Verwerfen Zurück
	5 Einträge - Seite 1 von 1 🕅 ┥ 🚺 🕨 🕅 10 🗸	∇ Q × & Ξ	
	ID Version Version Version Version T		
	SUT00004 Finance Tracker 2.0.0		
	SUT00002 Finance Tracker 1.1.0		
	SUT00001 Finance Tracker 1.0.0		
		Zuweisen Abbrechen	

Abbildung 9.36. Der Dialog "Testsysteme zuordnen"

9.6.5.5. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Issues an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

9.6.6. Issue - Details

≡ 👱 K L A R	os test manag	S E M E N T	DE Finanz-Tracker 🖀 🗮	વ× છ~ ≛*
📝 Definieren				\$7 🖨 🛛 😋 🕏
🙁 Planen	Eigenschaften		Speichern	Verwerfen Zurück
🏟 Ausführen	Issue Management System	IM00008 - PLAYGROUND 🗸 🔀 (hudson)		*
쓵 Auswerten	ID Status	Open		*
🗲 Konfigurieren	Zusammenfassung * Vorgangstyp Priorität Komponente(n) Betroffene Version(en)	Confirm button is not clickable. Bug Blocker Confirm button is not clickable.		
	Lösungsversion(en) Zugewiesen Fällig			
	Beschreibung	After editing a standing order, the confirm button is not clickable.		
	Testumgebung	Android Tablet, ID:A76t		

Abbildung 9.37. Die Seite "Issue erstellen"

Das erste Issue-Management-System des aktiven Projekts wird automatisch aus der Dropdown-Liste Issue-Management-System ausgewählt.



Anmerkung

Falls die Liste der Issue-Management-Systeme leer ist, muss mindestens ein Issue-Management-System konfiguriert und an das aktive Projekt angebunden werden. Die Konfiguration von Issue-Management-Systemen ist in <u>Abschnitt 10.5.1, "Issue</u> <u>Management"</u> beschrieben. <u>Abschnitt 6.1.2.6.1, "Issue Management"</u> enthält Anweisungen zum Hinzufügen eines Issue-Management-Systems zu einem Projekt.

Standardmäßig verwendet Klaros-Testmanagement die auf dem Anmeldebildschirm eingegebenen Anmeldedaten zur Authentifizierung der Benutzer mit dem Issue-Management-System. Bei Verwendung von Bugzilla wird auch die E-Mail/Passwort-Kombination des Benutzers ausprobiert, da die E-Mail üblicherweise der Anmeldename in Bugzilla ist. Wenn die bei der Anmeldung eingegebenen Anmeldedaten nicht funktionieren und das Issue-Management-System während der Sitzung noch nicht verwendet wurde, wird Klaros-Testmanagement neue Authentifizierungsinformationen in einem Dialogfenster anfordern.



Token-basierte Authentifizierung (Jira Cloud instances, GitLab)

Klaros-Testmanagement unterstützt auch ein Token-basierte Authentifizierung. Verwenden Sie dazu einen leeren Benutzernamen und das Token als Passwort.

Einige Anbieter wie Jira und GitLab beschränken den Zugriff auf Token-basierte Authentifizierung. Die Authentifizierung über Benutzername und Passwort wird für diese Systeme nicht mehr unterstützt. Beim Klick auf das Icon 🗹 öffnet sich das externe Issue-Management-System in einem neuen Browser-Tab.

Eim Klick auf das Icon 🖬 öffnet einen Dialog zum Ändern der Authentifizierungsinformationen.

Sobald ein Issue-Management-System ausgewählt wurde, wird eine Liste der Issue-Felder angezeigt. Die Anzahl und Bezeichnungen dieser Felder variieren je nach Art und Konfiguration des Issue-Management-Systems. Pflichtfelder sind farbig unterlegt.

0000000	Anmerkung			
	Um das Issue speiche	rn zu können, müssen alle Pflichtfelder ausgefüllt sein.		
	Issue-Fe	elder für jedes Issue-Management-System		
Testfälle		Die Testfälle, mit denen dieser Issue verknüpft ist.		
Testsystem	ie	Die Testsysteme, mit denen dieser Issue verknüpft ist.		
Testfallstatus aktualisieren		Durch Aktivieren dieser Checkbox wird neue Dropdown-Liste geöffnet, in der der neue Status der verknüpften Testfälle aus- gewählt werden kann.		

9.6.6.1. Einen Issue in Bugzilla anlegen

Issue Management System	IM00012 - TestProduct 💙 🗹 🖬 (root@verit.de)
ID	
Statue	
Zucommonfocoung *	UNCONFIRMED
Zusammenrassung "	
Plattorm	
Betriebssystem	
Priorität	Highest 💙
Wichtigkeit	blocker 💙
Komponente(n)	Left Component 💙
Version	unspecified 🗸
Meilenstein	V
Beschreibung	
Beschreibung (generiert)	
	TC00002 - Kontostand abrufen
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9-
	90000104921
Large Text	
Multiple Select	×
Drop down	*
Bug ID	
Free Text	
Integer	
Date	
Date/Time	
Date, Hille	
Testfall	TC00002 - Kontostand abrufen 🗙 🗸
Testsystem	FinanceTracker-V16.0.4 X
Testfall-Status ändern?	×
* Pflichtfelder müssen ausge	füllt werden
	Opsishers Varusafan 7
	Speichern Verwerten Zuruck

Abbildung 9.38. Die Seite "Issue Details Bugzilla"

9.6.6.2.	Einen	Issue	in	Github	anlegen

Issue Management System ID	
Status	open 💙
Zusammenfassung *	
Zugewiesen	
Meilenstein	V
Beschreibung	
Beschreibung (generiert)	TC00002 - Kontostand abrufen
Testfall	TC00002 - Kontostand abrufen 🗙 🗸
Testsystem	FinanceTracker-V16.0.4 🗙 🗸
Testfall-Status ändern?	V
* Pflichtfelder müssen ausge	füllt werden
	Speichern Verwerfen Zurück

Abbildung 9.39. Die Seite "Issue Details - GitHub"

9.6.6.3. Einen Issue in GitLab anlegen

Issue Management System	
ID	· · · · · · · · · · · · · · · · · · ·
Status	opened ¥
Zusammenfassung *	
Labels:	
Zugewiesen	V
Meilenstein	V
Beschreibung	
Beschreibung (generiert)	
	TC00002 - Kontostand abrufen
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9-
	900001104921
Testfall	TC00002 - Kontostand abrufen 🗙 🗸
Testsystem	FinanceTracker-V16.0.4 🗙
Testfall-Status ändern?	×
* Pflichtfelder müssen ausge	fjillt werden
i montrelaci madden aadge	
	Speichern Verwerfen Zurück

Abbildung 9.40. Die Seite "Issue Details - GitLab"

9.6.6.4. Einen Issue in JIRA anlegen

Issue Management System	IM00006 - PLAYGROUND 💙 🗹 🖬 (builder)
ID	· · · · · · · · · · · · · · · · · · ·
Zusammenfassung *	
Vorgangstyp	Taek V
Priorität	
Komponente(n)	
Retroffene Version(en)	
Lösungsversion(en)	
Zugewiesen	
Fällig	
rang	
Beschreibung	
Beschreibung (generiert)	
5(5)	TC00002 - Kontostand abrufen
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9-
	9be06f10492f
Testumaebuna	
, , , , , , , , , , , , , , , , , , , , , , , , , , , , , , , ,	
Testumgebung (generiert)	
restungebung (generiert)	ENV00001 - ANDROID 13
Sprint	×
Epic Link	
JiraClientCustom	
Testfall	TC00002 - Kontostand abrufen 🗙
Testsystem	FinanceTracker-V16.0.4 X
Testfall-Status ändern?	V
Testumgebung	ANDROID 13 ¥
* Pflichtfelder müssen ausge	füllt werden
	Speichern Verwerfen Zurück

Abbildung 9.41. Die Seite "Issue Details - JIRA"

9.6.6.5. Einen Issue in Mantis anlegen

Issue Management System	IM00005-MANTISTEST V 🗹 🖬 (user)
U	• •
Status	new 🗸
Zusammenfassung *	
Reproduzierbarkeit	always 🗸
Priorität	none 🗸
Wichtigkeit	feature 💙
Produktversion	2.0 🗸
Zielversion	2.0
Fix-Version	2.0 ¥
Zugewiesen	v
Kategorie	General 🗸
Beschreibung	
Beschreibung (generiert)	
	TC00002 - Kontostand abrufen
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9-
	9DeUot10492t
Schritte zur Reproduktion	
Zusätzliche Informationen	
Zusatziiche informationen	
DATE	
ENUM	1 🗸
FLOAT	
NUMERIC	
MULTILIST	~
TEXTBOX	
RADIO	0 1 0 2
STRING	
Checkboxes are fun	~
EMAIL	
LIST	1 🗸
Testfall	TC00002 - Kontostand abrufen X
Testsystem	FinanceTracker-V16.0.4 X
Testfall-Status ändern?	v
* Pflichtfelder müssen ausge	füllt werden

Abbildung 9.42. Die Seite "Issue Details - Mantis"

9.6.6.6. Einen Issue in Redmine anlegen

Issue Management System	IM00007 - PLAYGROUND 💙 🗹 🖬 (builder)
ID	
Zusammenfassung *	
Vorgangstyp	Bug 🗸
Priorität	Low
Zugewiesen	▼
Version	v
Kategorie	Left Category 💙
Status	New 🗸
Fortschritt	0% 🗸
Start	
Fällig	
Desekssikura	
Beschreibung	
Beschreibung (generiert)	T000002 Kentestand shrufan
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9- 9be06f10492f
Geschätzte Dauer	
String	
Boolean	
Date	曲
Float	
Text	
List	×
Int	
Testfall	TC00002 - Kontostand abrufen X
Testsystem	FinanceTracker-V16.0.4 X
Testfall-Status ändern?	v
* Pflichtfelder müssen ausge	füllt werden
	Speichern Verwerfen Zurück

Abbildung 9.43. Die Seite "Issue Details - Redmine"

9.6.6.7. Einen Issue in Trac anlegen

Issue Management System	IM00013
ID	
10	
Zusammenfassung *	
Vorgangstyp	defect 🗸
Priorität	blocker 🗸
Komponente(n)	V
Besitzer	
Version	×
Meilenstein	v
Beschreibung	
Beschreibung (generiert)	
5 (5)	TC00002 - Kontostand abrufen
	http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9-
	9be06f10492f
011 1	
Stichworte	
test_three	
test one	
test_one	
tes_two	
test_rive	Uno dos tres cuatro cinco
test_six	Default text
Testfall	TC00002 - Kontostand abrufen X
Testsystem	
Teetfall-Status ändern?	
restrain status alluern?	Y
* Pflichtfelder müssen ausge	füllt werden
	Speichern Verwerfen Zurück

Abbildung 9.44. Die Seite "Issue Details - Trac"

9.6.6.8.	Einen	Issue in	YouT	rack	anlegen
5.0.0.0.		100000 111	1001	10010	annegen

Issue Management System ID	IM00011 - DEMO Y Z I (TOKEN)
Zusammenfassung *	
Beschreibung	
Beschreibung (generiert)	TC00002 - Kontostand abrufen http://localhost:8080/klaros-web/browse.jsp?type=testCase&uuid=39ced99b-e06f-4049-8ed9- 9be06f10492f
Priority	Show-stopper 💙
Туре	Bug 🗸
State	Submitted ¥
Subsystem	No Subsystem 💙
Fix versions	✓
Affected versions	✓
Fixed in build	Next Build 💙
Assignee	Unassigned 💙
Estimation	
Spent time	
Testfall Testsystem Testfall-Status ändern? * Pflichtfelder müssen ausge	TC00002 - Kontostand abrufen X FinanceTracker-V16.0.4 X V füllt werden
	Speichern verwenen Zuruck

Abbildung 9.45. Die Seite "Issue Details - YouTrack"

9.6.7. Einen Issue verlinken

Issues können aus Klaros-Testmanagement heraus erstellt werden oder aus einem Issue-Management-System mit Testfällen und Testsystemen in Klaros-Testmanagement verknüpft werden.

≡ 📌 K L A R) S test management	DE Finanz-Tracker	≣ Q× _ ×
📝 Definieren	Issues		G
🙁 Planen	Neu Verknüpfen		Speichern Verwerfen
_	Issue verknüpfen		
🏟 Ausführen	System IM00008 - PLAYGROUND V C II (hudson)		
욙 Auswerten		·	
🖋 Konfigurieren	Verknüpfen		Schließen
	12 Einträge - Seite 1 von 1	2 🕅 🚽 1 2 🕨 🕅 10 🗸	√ Alle anzeigen Q × Alle Ξ
	🔲 ID 🗢 🛛 🖓 System 🗢 Zusammenfassung 🖨	Testfälle Testsysteme	Zuständig 🗢 Priorität 🗢 Status 🗢 Aktion
	PLAYGROUND-25702 🙀 IM00008 test Issue	> Aktualisieren Sie das Dashboard durch Hinzufügen Finance Tracker eines Dauerauftrags. 2.0.1	admin Blocker Open 🗹 🖒 🖨 🛍
	PLAYGROUND-25702 🙀 IM00008 test Issue	>\$ Aktualisieren Sie das Dashboard durch Hinzufügen Finance Tracker eines Dauerauftrags. 1.1.1	admin Blocker Open 🗹 🖒 🛱 🛍

Abbildung 9.46. Das "Issue verknüpfen" Panel

Um ein Issue mit einem Testfall oder Testsystem zu verknüpfen, muss zuerst ein Issue-Management-System ausgewählt werden. Ein Klick auf den Button Verknüpfen verknüpft den Issue mit der eingegebenen *ID* mit den ausgewählten Testfällen und Testsystemen.

SystemDas Issue-Management-System, von dem aus Issues ver-
knüpft werden sollen.

ID Die ID des zu verknüpfenden Issues.

Beim Klick auf das Icon Q wird das Issue-Management-System nach dem Issue mit der eingegebene ID durchsucht. War die Suche erfolgreich, verknüpft ein Klick auf den Button Verknüpfen den Issue mit den ausgewählten Testfällen und Testsystemen.

Kapitel 10. Konfigurieren

Im Bereich *Konfigurieren* werden alle administrativen Aufgaben vorgenommen. Dazu gehören die **Systemkonfiguration**, das Verwalten der **Berichtsvorlagen**, das Erstellen und Verwalten der **Benutzerkonten**, das Ändern von Einstellungen innerhalb des **Systems**, die **Integration** der externen Systeme sowie das **Sichern und Wiederherstellen** von Projekten.

Um auf die hier beschriebenen Funktionen zugreifen zu können, ist in den meisten Fällen die globale Rolle *Administrator* erforderlich.

10.1. Überblick

Auf der Seite Übersicht finden Sie die folgenden Informationen:

- Server
- Datenbank
- Betriebssystem
- Gebietsschema
- Java
- Speicher
- Lizenz
- Version



Abbildung 10.1. Die Seite "Übersicht"

10.2. Berichtsvorlagen

enterprise edition

Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Auf der Seite *Berichtsvorlagen* können benutzerdefinierte Berichtsvorlagen erstellt, bearbeitet, dupliziert und Berichte als PDF oder Excel-Datei ausgegeben werden.



Anmerkung

Die vordefinierten Berichtsvorlagen sind nicht editierbar und sollten zur Bearbeitung dupliziert werden.



Abbildung 10.2. Die Seite "Berichtsvorlagen"

Die Tabelle zeigt die folgenden Werte:

Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Name	Der Name der Berichtsvorlage.
Beschreibung	Die Beschreibung der Berichtsvorlage.
Revision	Die Revision der Berichtsvorlage.
Status	Der Status der Berichtsvorlage (Entwurf, Genehmigt).
Format	Das Ausgabeformat des Berichts (PDF oder Excel).
Aktion	Die ausführbaren Aktionen.



Anmerkung

Die Spalte Zusätzliche Hinweise zeigt an, ob eine Berichtsvorlage zur Bearbeitung gesperrt ist.

10.2.1. Aktion

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- Den Bericht als PDF-Datei erzeugen.
- Den Bericht als Excel-Datei erzeugen.
- 🕜 Bearbeiten
- Duplizieren
- 🗊 Löschen

Wenn ein Bericht erzeugt werden soll, der die Eingabe von Parametern erfordert, öffnet sich ein Dialog. Nach Bestätigung mit OK wird der Bericht erzeugt. Der Button bleibt gesperrt, wenn Pflichtparametern eine Benutzereingabe fehlen. Mit dem Button Abbrechen wird die Aktion abgebrochen.



Abbildung 10.3. Einen parametrisierten Bericht erzeugen

1	00	000	00	0
1	Ξ	-		\rightarrow
	=			
ľ		-		

Anmerkung

Beim Klick auf den Button <u>Neu</u> öffnet sich die Seite *Berichtsvorlage - Detailansicht* (<u>Abbildung 10.4</u>).

10.2.2. Berichtsvorlage - Detailansicht

Auf der Seite *Berichtsvorlage - Detailansicht* können der Name und die Beschreibung der Berichtsvorlage festgelegt werden. Außerdem müssen ein gültiges Skript und eine Layout-Vorlage angegeben werden. Das Skript und die Layout-Vorlage können direkt im Textbereich bearbeitet oder
als externe Datei hochgeladen werden. Im Reiter Revisionen können ältere Versionen des Berichts wiederhergestellt werden sowie Unterschiede zu einer vorangegangenen Version aufgelistet werden.

≡	Y KLAR	OS TEST	MANAGEMENT		DE Finanz-Tracker	≡	५ × 0 × ≛ ×
ľ	Definieren	RT00007 - Issu	ue Overview Report				6 9
2	Planen	Eigenschaften	Parameter (3) Skript	Layout-Vorlage Revisionen Änderungen	Vorschau	Speichern V	erwerfen Zurück
\$	Ausführen	Name	Label	Beschreibung	Тур	Voreinstellun	g Verpflichtend Aktion
		sortBy	Sort By	The sort parameters for the report	Aufzählung	Creation Date	1
6	Auswerten	onlyOpen	Only Unresolved Issues	List only the issues that are not yet resolved	Boolescher Wert	\checkmark	1 🗊
		testCaseDetail	Show Test Case Details	Show additional information for test cases	Boolescher Wert	\checkmark	1 🗊
2	Konfigurieren	Neuer F	Parameter		Vorschau	Spelchem V	erwerfen Zurück

Abbildung 10.4. Die Seite "Berichtsdetails"



Integrierte Syntax-Überprüfung

Um die Berichtsvorlage speichern zu können, dürfen das Skript und die Layout-Vorlage keine syntaktischen Fehler enthalten. Vor dem Speichern findet eine Überprüfung statt.

Eine ausführliche Beschreibung zum Erstellen von benutzerdefinierten Berichten finden Sie in <u>Abschnitt 11.2, "Eine neue Berichtsvorlage erstellen"</u>.

10.3. Benutzer

Im Bereich Benutzer werden die Benutzerkonten erstellt und bearbeitet.

10.3.1. Übersichtsseite

Die Übersichtsseite gibt einen Überblick über die Benutzerkonten in Klaros-Testmanagement. Die Benutzerkonten werden hier erstellt, bearbeitet, zusammengefasst, aktiviert und deaktiviert.

= 📌 K L A R G	DS TEST MA	NAGEMENT		DE Finanz-	Tracker 🖬 🗮	Q X	0× ≛×
📝 Definieren	Benutzer						
半 Planen	Neu					Speichern	Verwerfen
	C ⊖ ⊠ 📚 🔟 □ Ω Benutzername <	Rolle	21 Einträge - Seite 1 von 3 Voller Name 🗢	M 🚽 1 2 3 🕨 M 10	E-Mail 🗢	Alle anzeig	en & ≡ Aktion
📮 Ausführen	🗌 🔺 admin	Administrator Felix Mustermann					໕ﯘﺷ
e Auswerten	gross koennecke	Gast Günther Gross Administrator Claudia Könnneck	e	gross@mail.de			C 🖨 🛈 C 🖨 🛍
6 H F F	A manager	Testmanager Max Mustermann					
Konfigurieren	olli	Tester Oliver Krams		te@mail.de			C - U C - Ú
	reilly	Administrator Patrick Reilly					C D Ó
	Sabrina	Administrator Sabrina Gidley Tester Sahra Berger		gidley@verit.de			K ⇔ W K ⇔ M
	selen	Administrator Sandra Selen		arif@verit.de			C 🖨 🛈
	Benutzername	Rolle	Voller Name		E-Mail	System-Account	Aktion
	Neu					Speichern	Verwerfen

Abbildung 10.5. Die Übersichtsseite "Benutzer"

Die Tabelle zeigt die folgenden Werte:

Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.
Benutzername	Der eindeutige Benutzername.
Rolle	Die Rolle des Benutzers (Administrator, Testmanager, Tester, Gast).
Voller Name	Der vollständige Name des Benutzers.
E-Mail	Die E-Mail-Adresse des Benutzers.
Aktionen	Die ausführbaren Aktionen.

10.3.1.1. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

- 🛯 Bearbeiten
- 🔒 Druckansicht öffnen
- 🗊 Löschen

Wurde ein Benutzer gelöscht, ist er zunächst mit einer Löschmarkierung versehen und nur noch für Administratoren sichtbar. Für gelöschte Benutzer steht dann statt *Löschen* die folgenden Aktionen zur Verfügung:

Wiederherstellen (nur Administrator)

10.3.1.2. Massenaktionen

Bestimmte Aktionen lassen sich auch auf mehrere Benutzerkonten gleichzeitig anwenden. Dazu wählen Sie in der Spalte ganz links die Benutzerkonten aus, auf die die Aktion angewendet werden soll.

Für Benutzerkonten werden folgende Massenaktionen unterstützt:

- 🛯 Bearbeiten
- 🔒 Druckansicht öffnen
- Email verschicken
- Senutzerkonten zusammenführen

Es besteht nicht die Möglichkeit Benutzer direkt endgültig zu löschen aber es können Benutzer zusammengeführt werden. Hierbei wird im Dialog der zu behaltene Benutzer ausgewählt (<u>Abbildung 10.6</u>). Der ausgewählte Benutzer übernimmt dabei die Aufgabenzuweisungen, Testergebnisse und Testläufe der anderen Benutzer. Anschließend werden die anderen Benutzer endgültig gelöscht und sind somit nicht mehr in der Benutzerliste aufgelistet.

Wiederherstellen (nur Administrator)

Massenaktionen sind ausführlich in <u>Abschnitt 5.2.3.1.5, "Massenaktionen"</u> beschrieben.

≡ 📌 K L A R	d s test management	DE Finanz-Tracker	a 2	α × 0γ ≗γ
📝 Definieren	Benutzer			
	Neu			Speichern Verwerfen
🅸 Ausführen	Image: Constraint of the state of the s	3 Eintrage - Seite 1 von 1 N Voller Name	E-Mail 🗢	V ^a Alle anzeigen 23 ≡ System-Account ♦ Aktion
🚯 Auswerten	Image: Testmanager Testmanager Image: Max Mustermann Image: Tester Image: Tester Image: Tester			2 0 0 0 2 0 2 0 2 0
🔑 Konfigurieren	Sind Sie sicher?			System-Account Aktion Spelchern Verwerfen
	ID Felix Mustermann Benutzername: admin - ID: Felix Mustermann Max Mustermann Benutzername: manager - ID: Max Mustermann Erika Mustermann Benutzername: tester - ID: Erika Mustermann Zusammengeführter Benutzer Zusammengeführter Benutzer Max Alle anderen Benutzer werden nach der Zusammenführu	Beschreibung \$	Erstellt Geändert 14.06.2020 09.03.2022 14.06.2020 14.06.2020 14.06.2020 14.06.2020 Erstellt Geändert acht werden. × X Abbrechen	

Abbildung 10.6. Der Dialog "Benutzer zusammenführen"

10.3.1.3. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Zeige alle / Nur aktive (Nur Administrator)

- 公 Exportieren
- \equiv Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.



Benutzerrollen ändern

Benutzer - einschließlich Administratoren - können nicht ihr eigenes *System Account* Flag oder die eigene *Rolle* ändern. Dies kann nur von einem anderen Benutzerkonto aus erfolgen, das die Rolle *Administrator* hat.

10.3.1.4. Einen Benutzer anlegen

Mit Klick auf den Button Neu wird ein neues Benutzerkonto erstellt (<u>Abbildung 10.5</u>). Die Attribute *Benutzername*, *Rolle*, *Vollständiger Name* und *E-Mail* werden hier eingegeben.

Mit Klick auf den Button <u>Speichern</u> erscheint für jedes neue Benutzerkonto ein Dialog in dem das *Passwort* angegeben und festgelegt wird, ob es sich um einen *System-Account* handelt, siehe (<u>Abbildung 10.7</u>).

≡	👱 KLAR	0 S	TEST MA	NAGEM	ENT			DE Finanz-Tracker			৭ @শ ≛শ
	Definieren	Benutzer									
			Neu							Speichern	Verwerfen
:2:	Planen		_								
		Ľ			22 Einträge - Seite 1 vo	in 2 🕅 ┥ 🚺	2	20 🗸		Y	Nur aktive
ń	Ausführen		Benutzername \$	Rolle 🗢	Voller	Name 🗢			E-Mail 🗢		Aktion
· *	/ dorum on			Tester							2 8 🛍
			🛕 admin	Administrator	Felix Mustermann						ぱ 🖨 🛍
Ģ	Auswerten		gross	Gast	Es sind weitere Angaben erforderlie	ch					☞ 🖨 🛍
			koennecke	Administrator	Bitte vervollständigen Sie d	ie Angaben					☞ 🖨 🛍
يو ا	Konfigurieren		🛕 manager	Testmanager		No Felgaberi					☞ 🖨 🛍
ſ ″			Meyer	Testmanager							r 🖨 🛍
			olli	Tester	Benutzername *	alex					☞ 🖨 🛍
			reilly	Administrator	Voller Name *	Alexander Müller					☞ 🖨 🛍
			Sabrina	Administrator	Neues Passwort *	•••••					☞ 🖨 🛍
			sahra	Tester	Wiederholung Neues Passwort *	•••••					☞ 🖨 🛍
			selen	Administrator	Rolle	Testmanager 🗸					C 🖨 🗓
			selen_DE	Administrator	E-Mail	mueller alexander@kl	aros tastman	acomont do			☞ 🖨 🟛
			selen2	Administrator	System-Account		alositestillali	sgemencue			r 🖨 🛍
			🛕 system	Tester	ojstem Account	V					ぱ 🖨 🛍
			Tafel	Tester			ок	Abbrechen			☞ 🖨 🛍
			Talal	Administrator							r 🖨 🖻
			Tegen	Tester	Till Tegen			tegen@mail.de			r 🖨 🛍
			🛕 tester	Tester	Erika Mustermann						☞ 🖨 🛍
			Thiel	Tester	Tim Thiel			thiel@mail.de			r 🖨 🛍
			Toppler	Tester	Tanja Toppler			toppler@mail.de			ピ 🖨 🛍
			Benutzername	Rolle	Volle	r Name			E-Mail		Aktion
			Neu							Speichern	Verwerfen

Abbildung 10.7. Der Dialog "Neuen Benutzer speichern"

Standardmäßig können Benutzer der Rolle Administrator oder Manager neue Benutzerkonten anlegen. Benutzer der Rolle Manager können nur Benutzerkonten der Rolle Tester oder Gast anlegen.



Einschränken der Kontoerstellung auf Administratoren

Wenn eine restriktivere Richtlinie gewünscht ist, kann die Einstellung *Nur Administratoren können Benutzer anlegen* verwendet werden. Siehe hierzu <u>Abschnitt 10.4.1,</u> <u>"Sonstiges"</u> für weitere Informationen.

10.3.1.5. Einen Benutzer löschen

Bei Klick auf das Icon fil wird ein Benutzerkonto deaktiviert. Nur Benutzer der Rolle Administrator können deaktivierte Benutzer sehen und wiederherstellen.



Benutzerkonten können nur deaktiviert, nicht gelöscht werden!

Durch Umschalten der Buttons Nur Aktive und Zeige Alle oberhalb der Tabelle können Administratoren die Anzeige der deaktivierten Benutzerkonten ein- oder ausblenden.

Anstatt Benutzerkonten zu löschen können diese mit anderen existierenden Benutzerkonten zusammengefasst werden. Bei diesem Vorgang werden alle zusammenzufassenden Benutzerkonten ausser dem verbleibenden Konto gelöscht.

10.3.2. Detailseite

Jedes Benutzerkonto verfügt über eine eigene Detailseite mit mehreren weiteren Unteransichten. Mit Klick auf den Benutzernamen des Benutzerkontos oder auf das Icon 😰 rechts in der Aktionsspalte gelangen Sie zu der zuletzt ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht Übersicht.

≡ 👱 K L A R	os test mana	AGEMENT		DE Finanz-Tracker 🛛 🚆	Q × 6	} ∼ ≗ ∼
Definieren	Benutzer					
::::::::::::::::::::::::::::::::::::::	Neu				Speichern	/erwerfen
	໕ 🖨 ⊠ 📚 û	2	21 Einträge - Seite 1 von 3 🛛 🖌 🔌	2 3 🕨 🗎 10 🗸	Alle anzeige	n & ≡
🏚 Ausführen	🔲 🖗 Benutzername 🗢	Rolle 🗢	Voller Name 🗢	E-Mail 🗢	System-Account 🗢	Aktion
	🗌 🛕 admin 🛛 A	Administrator Felix Mustermann		0.11		C C D D
Auswerten	gross G	Gast Günther Gross		gross@mail.de		
	Koennecke A	Administrator Claudia Konnnecke				
6 11 1 1	Mover T	Testmanager Max Mustermann		mover@mail.do		
Kontigurieren		Tester Oliver Krams		te@mail.de		C A A
	reilly A	Administrator Patrick Reilly		telemande		戊日前
	Sabrina A	Administrator Sabrina Gidlev		gidlev@verit.de		C 🛆 🏛
	sahra T	Tester Sahra Berger		3		RAM
	selen A	Administrator Sandra Selen		arif@verit.de		C D D
	Benutzername	Rolle	Voller Name	E-Mail	System-Account	Aktion
	Neu				Speichern \	/erwerfen

Abbildung 10.8. Die Ansicht "Eigenschaften"

Es stehen die folgenden Ansichten zur Verfügung: Eigenschaften, Projektrollen, Aufgaben, Ergebnisse und Änderungen.

10.3.2.1. Aktionen

Auf den Detailseiten befinden sich rechts oben in der Kopfzeile weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

🔒 Druckansicht erstellen	Hier kann eine druckfreundliche Ansicht des Benutzerac-
	counts erstellt werden. Mit Klick auf das Icon 🔒 öffnet sich
	diese in einem neuen Browsertab.

Druckansichten sind ausführlich in <u>Abschnitt 5.2.3.2.1, "Druck-ansichten"</u> beschrieben.

G 🕤 Blättern

Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Benutzerkonten gewechselt werden.

10.3.2.2. Eigenschaften

In dieser Ansicht (<u>Abbildung 10.8</u>) werden die folgenden Attribute des Benutzerkontos angezeigt und können bearbeiten werden:

Benutzername	Der Login-Name des Benutzers
Voller Name	Der volle Name des Benutzers
E-Mail	Die E-Mail-Adresse des Benutzers, die für E-Mail-Benachrich- tigungen verwendet wird. Falls ein entsprechendes Ereignis konfiguriert wurde (siehe <u>Abschnitt 10.4.2, "Benachrichtigun-</u> <u>gen"</u>) und beim Anlegen eines neuen Benutzers eine E-Mail- Adresse angegeben wird (siehe <u>Abschnitt 10.3.1.4, "Einen</u> <u>Benutzer anlegen"</u>), erhält dieser Benutzer eine Registrie- rungs-E-Mail mit Benutzernamen und Passwort.
System Account	Ist dieses Kästchen aktiviert, kann sich dieser Benutzer nicht mehr auf der Anmeldeseite anmelden. Systemkonten sind für automatisierte Aufgaben wie das Importieren von Daten vor- gesehen.
Aktiv	Ist dieses Kästchen deaktiviert, kann der Benutzer sich nicht mehr auf der Anmeldeseite anmelden.
Passwort	Das Passwort, um sich in Klaros-Testmanagement anzumel- den.
Passwort wiederholen	Die Bestätigung des Passwortes.
Rolle	Die Benutzerrolle. Zur Verfügung stehen die Rollen Administra- tor, Testmanager, Tester oder Gast.



Beschränkungen beim Ändern von Benutzerrollen

Benutzer können ihr eigenes *System Account* Flag und ihre *Rolle* nicht ändern. Dies muss durch einen anderen Benutzer mit der Rolle *Administrator* geschehen.

10.3.2.3. Projektrollen



Die Ansicht *Projektrollen* zeigt alle Projektrollen, die der Benutzer in Projekten mit projektspezifisch vergebenen Rollen besitzt (siehe <u>Abschnitt 6.1.2.5, "Zugang"</u>). Administratoren können hier die Projektrollen für diese Projekte ändern. Diese Ansicht ist erst verfügbar, sobald das erste Projekt mit projektspezifischen Rollen versehen wurde.

= 📌 K L A R	o s test management	DE Finanz-Tracker 🗃 🗮 🔍 🔍 🗙 🚱 V 💄 V
📝 Definieren	Oliver Krams	a 🗘 🖓
🏩 Planen	Eigenschaften Projektrollen (1) Aufgaben Ergebnisse Änderungen	Speichern Verwerfen Zurück
💠 Ausführen	2 ① 1 Einträge - Seite 1 von 1 K ◀ 1	₩ 10 ▼ 公 =
🚯 Auswerten	ID Beschreibung Beschreibung	Rolle ¢ Tester
🔑 Konfigurieren	ID Beschreibung Erstellt vor 4 Jahren von Felix Mustermann	
	Zuweisen	Zuletzt geändert vor 4 Jahren von Felix Mustermann Speicherm Verwerfen Zurück

Abbildung 10.9. Die Ansicht "Projektrollen"

Mit Klick auf den Button Zuweisen öffnet sich ein Dialog mit einer Liste aller Projekte mit projektspezifisch vergebenen Rollen, denen der Benutzer noch nicht zugewiesen ist. Mit Ok wird der Benutzer den ausgewählten Projekten zugeordnet.

= 👱 K L A R) s test management	DE Finanz-Tracker 🖀 🗮 🚺 🖉 🖉 🖕 🖕
📝 Definieren	Markus Meyer	⊖
🏝 Planen	Eigenschaften Projektrollen Aufgaben Ergebnisse Änderungen	Speichern Verwerfen Zurück
🏟 Ausführen	2 🛍 0 Einträge - Seite 1 von 1 🖌 ┥ 🕨	₩ 10 ▼ 公 =
Auswerten	Beschreibung	Rolle 🗢
	ID Beschreibung	
🌽 Konfigurieren	Erstellt vor 3 Jahren von Felix Mustermann	Zuletzt geändert vor 3 Jahren von Felix Mustermann
	Projektrolle zuweisen	Verwerfen Zurück
	Der Benutzer wird den folgenden Projekten mit der unten ausgewählten Rolle zugewiesen.	
	1 Einträge - Seite 1 von 1 🔣 🖣 🔰 🕨 10 💌	7 & ≡
	ID Beschreibung	
	P00003 Printer Tester	
	Benutzer hinzufügen als Tester V	
		OK Abbrechen

Abbildung 10.10. Der Dialog "Projektrollen zuweisen"

10.3.2.4. Aufgaben

Die Ansicht *Aufgaben* listet alle Aufgaben auf, die dem Benutzer zugewiesen sind. Unter <u>Abschnitt 7.1, "Aufgaben"</u> finden Sie weitere Informationen zur Verwaltung von Jobs.

10.3.2.5. Ergebnisse

Die Ansicht *Ergebnisse* zeigt alle Testergebnisse der Tests, die der Benutzer durchgeführt hat, siehe <u>Abschnitt 5.2.3.2.7, "Testläufe und Ergebnisse"</u>.

10.3.2.6. Änderungen

In der Ansicht Änderungen ist die Änderungshistorie zu sehen, siehe <u>Abschnitt 5.2.3.2.8, "Ände-</u> rungshistorie".

10.4. System

Auf der Seite *System* werden die Einstellungen für das Verhalten und die Darstellung der Anwendung festgelegt. Es sind fünf Ansichten vorhanden: *Sonstiges, Benachrichtigungen, Testausführung, Oberfläche* und *Sprachen*.

10.4.1. Sonstiges

≡ 🞐 K L A R (o s test management	DE Finanz-Tracker 🖬 🗮 🔍 🔍	× 0~ ≛~
📝 Definieren	System		
📇 Planen	Sonstiges Benachrichtigungen Testausführung Oberfläche Spracher	n	
🏟 Ausführen	URL der Anwendung	http://infosys.verit.de/klaros-prime	
🕒 Auswerten	Maximale Dateigröße beim Hochladen [Byte]	10485760	
🔑 Konfigurieren	Session Timeout des Containers benutzen Session Timeout (Minuten)	30	
	Nur Administratoren können neue Benutzer anlegen Nur Manager und Administratoren sehen andere Benutzer in der Benutzerverwaltung Ändern des Aufgabenstatus erfordert die Rolle Manager Nur genehmigte Berichtsvorlagen unter Evaluieren/Berichte anzeigen Vordefinierte Berichte verbergen Such-Index	Aktualisieren	Verwerfen

Abbildung 10.11. Die Ansicht "Sonstiges"

Hier können folgende Einstellungen vorgenommen werden, um das Verhalten der Anwendung festzulegen:



Session Timeout

Ein geänderter Session Timeout beeinflusst nicht die aktuelle Sitzung.

URL der Anwendung	Soll auf die Anwendung über einen Proxy-Server zugegriffen werden, muss die dort konfigurierte URL hier angegeben werden.
	Detaillierte Informationen zur Konfiguration eines Proxies finden Sie unter https://httpd.apache.org/docs/2.4/mod/ mod_proxy.html (Apache), https://docs.nginx.com/nginx/ admin-guide/web-server/reverse-proxy/ (NGinx) https://docs.

microsoft.com/en-us/iis/extensions/ configuring-application-request-routing-arr/creating-aforward-proxy-using-application-request-routing (IIS).

	Mit Klick auf das Icon ଜ wird die angegebene Verbindung überprüft.
Maximale Dateigröße beim Hochladen [Byte]	Die maximale Größe für hochzuladende Anhänge.
Session Timeout des Contai- ners benutzen	Ist diese Option aktiviert, wird das Standard-Session-Timeout für Tomcat-Sitzungen verwendet (30 Minuten).
Session Timeout (Minuten)	Ist die obige Checkbox nicht aktiviert, wird der hier angegebe- ne Wert als Session Timeout verwendet.
Nur Administratoren können neue Benutzer anlegen	Wenn aktiviert, können nur Administratoren neue Benutzer anlegen. Wenn nicht, können auch Manager neue Benutzer anlegen.
Nur Manager und Administra- toren sehen andere Benutzer in der Benutzerverwaltung	Wenn aktiviert, sehen Benutzer in den Rollen <i>Tester</i> und <i>Gast</i> nur ihren eigenen Benutzereintrag, Einträge anderer Benutzer werden nicht angezeigt.
Ändern des Aufgabenstatus erfordert die Rolle Manager	Wenn aktiviert, können nur Benutzer in der Rolle <i>Manager</i> oder höher den Status von Aufgaben ändern.
Nur genehmigte Berichtsvorla- gen unter Evaluieren/Berichte anzeigen	Falls aktiviert, werden auf der Seite <i>Evaluieren / Berichte</i> in der Tabelle <i>Konfigurierbare Berichte</i> nur noch Berichte mit dem Status <i>Genehmigt</i> angezeigt.
Vordefinierte Berichte verber- gen	Falls aktiviert, wird auf der Seite <i>Evaluieren / Berichte</i> die Anzei- ge der Tabelle <i>Berichte</i> unterdrückt.
Such-Index	Sollten Probleme mit der allgemeinen Suche auftreten, kann hier mit dem Button Aktualisieren eine Neuindizierung der Datenbank gestartet werden.

10.4.2. Benachrichtigungen

edition

erprise Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Bei verschiedenen Ereignissen, wie zum Beispiel der Einrichtung eines Benutzerkontos oder der Zuweisung einer Aufgabe, werden vom System Benachrichtigungen per E-Mail verschickt. In der Ansicht *Benachrichtigungen* werden die entsprechenden Bedingungen dafür festgelegt.

In einem Benachrichtigungsschema können mehrere Benachrichtigungseinstellungen definiert und für mehrere Projekte wiederverwendet werden.



E-Mail-Konfiguration erforderlich!

Damit Benachrichtigungen ordnungsgemäß funktionieren, müssen wie in <u>Abschnitt 10.5.3, "E-Mail"</u> beschrieben die E-Mail-Server-Einstellungen konfiguriert werden und es muss sichergestellt sein, dass alle Benutzerkonten eine gültige E-Mail-Adresse haben.

10.4.2.1. Übersichtsseite

Die Übersichtsseite zeigt alle vorhandenen Benachrichtigungsschemata in einer Tabelle an. Neue Benachrichtigungsschemata werden hier erstellt.

= 坐 K L A R C) s test managem	IENT	DE Fina	inz-Tracker 🖀 🗮	२ ४ 0 × ≜ ×
📝 Definieren	System				
📇 Planen	Sonstiges Benachrichtigungen	Testausführung Oberfläche Sp	orachen		
	Benachrichtigungsschemata				
Austunren		24 Einträge - Seite 1 vo	n 3 🕅 ┥ 1 2 3 🕨 🕅 10	~	7 & ≡
	ID ♥ Name ∜		Beschreibung 🗢	Voreinstellung Ereig	nisse 🗢 Projekte 🗢 🛛 Aktion
C Auswerten	NS00294 test	dis			
	NS00290 test	dis			
Konfigurieren	NS00255 test	dis			
	NS00244 test	dis			
	NS00229 test	dis			0 0 12 日前
	NS00225 test	dis			5 0 201
	NS00218 test	dis			5 0 戊口前
	NS00205 test	dis			5 0 🖉 🗋
	NS00195 test	dis			5 0 ピロ⑪
	ID Name				gnisse Projekte Aktion
	Neu				Speichern Verwerfen

Abbildung 10.12. Die Übersichtsseite "Benachrichtigungsschemata"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.
Name	Der Name des Benachrichtigungsschemas.
Beschreibung	Die Beschreibung des Benachrichtigungsschemas.
Voreinstellung	Wenn aktiviert, ist das Benachrichtigungsschema das Stan- dardschema. Wenn ein Benachrichtigungsschema als Stan- dardschema deklariert ist, wird es automatisch jedem bestehenden Projekt zugewiesen, es sei denn, für dieses Pro- jekt wurde bereits ein anderes Schema definiert.
Ereignisse	Die Anzahl der Benachrichtigungsereignisse.
Projekte	Die Anzahl der Projekte, denen dieses Benachrichtigungssche- ma zugeordnet ist.
Aktionen	Die ausführbaren Aktionen.

10.4.2.1.1. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

🛯 Bearbeiten

Duplizieren

fi Löschen

10.4.2.1.2. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

- Filter / Sortieren
- 公 Exportieren
- ∃ Spaltenauswahl

Alle Operationen sind ausführlich in <u>Abschnitt 5.2.3.1, "Übersichtsseite"</u> beschrieben.

10.4.2.1.3. Detailseite

<u>Abbildung 10.13</u> zeigt die Bearbeitungsoptionen für ein Benachrichtigungsschema.

= 🞐 KLAR	o s test management		DE Finanz-Tracker 🖀 🛢	५ ४ 0∕ × ≜ ×
📝 Definieren	System			
砦 Planen	Sonstiges Benachrichtigungen Testausführung	Oberfläche Sprachen		
🄹 Ausführen	Schema: NS01889 - SupportAutoReply Ereignis	E	Empfänger	Aktion
🚯 Auswerten	Aufgabe bereit zur Ausführung Aufgabe zugewiesen			+
🔑 Konfigurieren	Benutzerkonto Passwort geändert			+
	Benachrichtigung hinzufügen Testausf Ereignis	Empfänger	_	+
	Erstellt v Aufgabe bereit zur Ausführung	O Verantwortlicher	art vor v	venigen Sekunden von Selen Deutsch
	Autgabe zugewiesen Benutzerkonto Passwort geändert Benutzerkonto angelegt	Aktiver Benutzer Spezifischer Benutzer	audia Könnnecke 🗸	Verwerfen Zurück
	Testausführung fehlgeschlagen			
			Hinzufügen Abbrechen	

Abbildung 10.13. Der "Benachrichtigung hinzufügen" Dialog

Auf dieser Seite sind alle Benachrichtigungen für das Benachrichtigungsschema aufgelistet. Zum Erstellen einer Benachrichtigung können Ereignisse aus einem vordefinierten Satz verfügbarer Benachrichtigungsereignisse individuell einem oder mehreren Empfängern (Benutzerkonten) zugeordnet werden.

Die verfügbaren Benachrichtigungsereignisse sind:

Aufgabe zugewiesen	Dieses Ereignis wird ausgelöst, sobald einem Benutzer eine Aufgabe zugewiesen wird.
Aufgabe bereit zur Ausführung	Dieses Ereignis wird ausgelöst, wenn eine nicht ausführbare Aufgabe ausführbar wird, oder alle Abhängigkeiten der Aufga- be erfüllt sind (siehe <u>Abschnitt 7.1.2.5</u> , <u>"Abhängigkeiten"</u>).

Benutzerkonto angelegt	Dieses Ereignis wird ausgelöst, sobald ein Benutzerkonto erstellt wurde.	
Benutzerkonto Passwort geän- dert	Dieses Ereignis wird ausgelöst, sobald ein Passwort in der lokalen Benutzerdatenbank geändert wurde.	
Testausführung fehlgeschlagen	Dieses Ereignis wird nach Beendigung einer Testfallausfüh- rung ausgelöst, falls ein Testfallschritt während der Ausfüh- rung als fehlgeschlagen oder fehlerhaft markiert wurde.	
Die verfügbaren Benachrichtigungsziele sind:		

Verantwortlicher	Der Verantwortliche Benutzer der Aufgabe/des Testfalls bzw. das angelegte/bearbeitete Benutzerkonto.
Ersteller	Der Benutzer, der für das Auslösen des Benachrichtigungser- eignisses verantwortlich ist, z.B. durch die Zuweisung einer Aufgabe.
Aktiver Benutzer	Der derzeit aktive Benutzer.
Spezifischer Benutzer	Ein bestimmter Benutzer.

Mit Klick auf den Button Projekte zuweisen kann das Benachrichtigungsschema einem oder mehreren Projekten zugewiesen werden (siehe <u>Abbildung 10.14</u>).

= 🞐 K L A R	o s test managem	ENT		DE Finanz-Tr	acker 🖬 🚆	Q X (Ø× ≛×
🕑 Definieren	System						
📇 Planen	Sonstiges Benachrichtigungen	Projekte dem Benachrichtigungsschema	zuweise	n			
🏟 Ausführen	Schema: NS00294 - test	Verfügbar P00011		Zugewiesen			
e Auswerten	Ereignis Aufgabe bereit zur Ausführung	P00010 - Test PRV P00009 - PRV 2					Aktion
🔑 Konfigurieren	Aufgabe zugewiesen Benutzerkonto Passwort geändert	P00008 - PRV P00007 - Issue Management Int	>				++
	Benutzerkonto angelegt Testausführung fehlgeschlagen	P00005 - DE Finanz-Tracker	»				++
	Erstellt vor 22 Stunden von selen2 seler	P00002 - Finance Tracker	«		Zuletz	t geändert vor 22 Stunden vo	n selen2 selen2
	Projekte zuweisen	P00001 - FinanceTrackerLocalT			Speichern	Verwerfen	Zurück
				Schließen			

Abbildung 10.14. Projekte einem Benachrichtigungsschema zuweisen

10.4.3. Testausführung



Abbildung 10.15. Die Ansicht "Testausführung"

In dieser Ansicht können folgenden Einstellungen bezüglich der Testausführung vorgenommen werden:

Beim Import von Testergebnis- sen für unbekannte Tests auto- matisch neue Testfälle anlegen	Wenn aktiviert, wird automatisch ein Testfall für die jeweiligen Testfallinformationen in der Ergebnisdatei erstellt. Diese Infor- mationen können je nach Importformat variieren.
Beim Import von Testergeb- nissen ein Testsuiteergebnis erzeugen und falls notwendig eine neue Testsuite anlegen	Wenn aktiviert, wird automatisch ein Testsuite-Ergebnis für die Testsuite-Informationen in der Ergebnisdatei erstellt. Diese Informationen können je nach Importformat variieren. Zusätz- lich wird für das Ergebnis der Testsuite eine entsprechende Testsuite erstellt, falls diese noch nicht vorhanden ist.
Tester dürfen Tests nur mittels Aufgaben ausführen	Wenn aktiviert, können Benutzer in der Rolle <i>Tester</i> Tests nur durch Ausführen einer Aufgabe starten.
Tester dürfen nur eigene Test- läufe wiederaufnehmen	Wenn aktiviert, können Benutzer in der Rolle <i>Tester</i> nur Test- läufe wiederaufnehmen, die sie selbst gestartet haben. Wurde ein Testlauf über eine Aufgabe gestartet, können sie diesen nur dann wiederaufnehmen, wenn ihnen diese Aufgabe zuge- ordnet ist.

10.4.3.1. Begründungsvorlagen für übersprungene Testergebnisse

Wenn Tester während der manuellen Testausführung einen Testfall überspringen wollen, so wird vom System eine Begründung hierfür erfragt. Diese Tabelle enthält die Vorlagen für diese Begründungen (siehe <u>Abschnitt 8.3.3.1.1, "Testfälle endgültig überspringen"</u>). Diese Vorlagen können beim Überspringen übernommen werden.

10.4.3.1.1. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

🛯 Bearbeiten

fi Löschen

≡ 🞐 K L A R	OS TEST MANAGEMENT DE Franz-Tracker	α × Θ∗ ≛×
📝 Definieren	System	
🎥 Planen	Sonstiges Benachrichtigungen Testausführung Oberfläche Sprachen	
🌣 Ausführen	Beim Import von Testergebnissen automatisch neue Testfälle für unbekannte Tests anlegen v Beim Import von Testergebnissen ein Testsuiteergebnis erzeugen und falls notwendig eine neue Testsuite anlegen v Tester dürfen Tests nur mittels Aufgaben ausführen Tester dürfen nur eigene Testläufe wieder aufnehmen	
🔑 Konfigurieren	Tester müssen eine Zusammenfassung des Testschritt-Ergebnisses hinzufügen	
	begrundungsvoriagen iur ubersprungene restergebnisse 699 Einträge-Seite 1 von 70 K ◀ 1 2 3 4 5 6 7 8 9 10 K K 10 ✓	Å.
	Begründung bearbeiten	Aktion
	Begründung Server ist nicht erreichbar.	2 0 2 0 2 0 2 0 4 0
	Zurück	2 0 C Û C Û
		ピ ⑪ ピ ⑪ ピ ⑪

Abbildung 10.16. Der Dialog "Begründung bearbeiten"

10.4.4. Oberfläche

In der Ansicht Oberfläche wird das Verhalten der Benutzeroberfläche festgelegt.

≡ 🞐 K L A R (o s test management		DE Finanz-Tracker 🧧 🚆	Q × 0 × ≗×
📝 Definieren	System			
🏝 Planen	Sonstiges Benachrichtigungen Testausführung O	berfläche Sprachen		
🏟 Ausführen	Der Testrunner startet in der Tabellenansicht Testfallinformationen bei der Ausführung einblenden			
🕒 Auswerten	Zeilen pro Tabellenseite	10 🗸		
🔑 Konfigurieren	Maximale Höhe des Vorschaubildes Maximale Breite des Vorschaubildes	72 320		
	Versionsnummer in Fußzeile anzeigen	v		
	Nachricht des Tages			
	Zufälliges Zitat des Tages verwenden			
		Name 🗢		Größe Aktion
	quotes.txt			24 KB 🔟
	Zitate hochladen			
				Speichern Verwerfen

Abbildung 10.17. Die Ansicht "Oberfläche"

Die folgenden Einstellungen können hier vorgenommen werden:

Der Testrunner startet in der	Wenn aktiviert, zeigt der Testrunner standardmäßig die tabel-
Tabellenansicht	larische Ansicht an.

Wenn aktiviert, werden Testfall- informationen bei der Ausfüh- rung eingeblendet	Wenn diese Option aktiviert ist, wird das Informationsfens- ter für Testfalldetails beim Starten oder Ausführen von Tests standardmäßig immer aufgeklappt.		
Der Testschritt-Editor startet in der registerbasierten Ansicht	Wenn aktiviert, ist die tabellarische Ansicht in der Ansicht <i>Schritte</i> auf der Seite <i>Testfalldetails</i> die Standardansicht.		
Zeilen pro Tabellenseite	Mit dieser Option wird die Anzahl der Zeilen pro Tabellenseite voreingestellt.		
Maximale Höhe des Vorschau- bildes	Diese Option legt die maximale Höhe der Vorschaubilder von eingebetteten Anhängen während der Testfall-Ausführung fest (siehe <u>Abbildung 6.45, "Ein Verweis auf einen Anhang wird</u> <u>durch eine Vorschau des Anhangs ersetzt"</u>).		
Maximale Breite des Vorschau- bildes	Diese Option legt die maximale Breite der Vorschaubilder von eingebetteten Anhängen während der Testfall-Ausführung fest (siehe <u>Abbildung 6.45, "Ein Verweis auf einen Anhang wird</u> <u>durch eine Vorschau des Anhangs ersetzt"</u>).		
Versionsnummer in Fußzeile anzeigen	Wenn aktiviert, wird die Versionsnummer der Anwendung in der Fußzeile angezeigt.		
Zitat des Tages	Eine Nachricht, die auf dem Anmeldebildschirm angezeigt werden soll.		
Zufälliges Zitat des Tages ver- wenden	Wenn ausgewählt, wird eine zufällige Zeile aus der unten hoch- geladenen Datei als Zitat des Tages angezeigt.		
Zitate hochladen	Mit Klick auf den Button Zitate hochladen kann eine neue Text- datei mit Zitaten ausgewählt und hochgeladen werden. Die Datei muss ein Zitat pro Zeile enthalten. Mit dem Icon in kann die hochgeladene Datei entfernt und durch die mitgelieferte Datei der Anwendung ersetzt werden.		
	<u>Abschnitt 4.2, "Zitat des Tages"</u> zu finden.		

10.4.5. Sprachen

Die Benutzeroberfläche kann mittels Sprachdateien um weitere Sprachen ergänzt werden.

= 👱 K L A R	o s test management	DE Finanz-Tracker 🖬 🗮 💽 🔍 🔍 🔍 🔍 🕹 V
📝 Definieren	System	
📇 Planen	Sonstiges Benachrichtigungen Testausführung Oberfläche Sprachen	
🌣 Ausführen	Verfügbare Sprachen	Aktivierte Sprachen
욙 Auswerten		Deutsch (de_DE)
🔑 Konfigurieren		
		>
		»
		<
		«
	Parashdotajan aktualiajaran	

Abbildung 10.18. Die Ansicht "Sprachen"

10.4.5.1. Sprachen aktivieren und deaktivieren

Es können bis zu fünf verschiedene Sprachen gleichzeitig aktiviert werden. Mit Klick auf das Icon > wird die ausgewählte Sprache hinzugefügt, mit dem Icon < entfernt. Um alle verfügbaren Sprachen hinzuzufügen, klicken Sie auf das Icon » bzw. auf « um alle Sprachen zu entfernen oder verwenden Sie die Drag and Drop-Funktion.



Dateiformat

Weitere Informationen über das Hinzufügen von Sprachen und dem verwendeten Dateiformat sind unter <u>Abschnitt 4.1, "Sprachdateien aktivieren und deaktivieren"</u> beschrieben.

Klaros-Testmanagement wird mit Sprachdateien in deutscher und englischer Sprache ausgeliefert. Die Unterstützung für weitere Sprachen kann bei Bedarf mittels zusätzlicher Sprachdateien hinzugefügt werden. Die entsprechende Sprachdatei muss hierzu im Verzeichnis .klaros/ resources/messages gespeichert werden.

Mit Klick auf den Button Sprachdateien aktualisieren werden die Sprachdateien im Verzeichnis . klaros/resources/messages bei laufender Anwendung neu geladen und in dieser Ansicht angezeigt.

10.5. Integration

Auf der Seite Integration werden externe Systeme angebunden und deren Einstellungen verwaltet. Hier gibt es sechs Ansichten, Issue Management, Anforderungsmanagement, E-Mail, Netzwerk, LDAP und CAS.

10.5.1. Issue Management

Ein Issue-Management-System (auch Issue Tracker, Bug Tracker genannt) ist eine Software-Anwendung zum Verwalten von Issues, die während des Testprozesses auftreten. Issues können Berichte über Fehler in einem Softwaresystem oder andere beobachtete Informationen enthalten.

Es können Issues erstellt, abgerufen und fehlgeschlagenen Testfällen zugeordnet werden. Es ist möglich, mehrere Issue-Management-Systeme zu konfigurieren und gleichzeitig zu verwenden.

Klaros-Testmanagement unterstützt die folgenden Issue-Management-Systeme:

Bugzilla, ein kostenloses Open-Source Issue-Management-System (siehe http://www.bugzilla.org/).

GitHub, ein webbasierter Git-Repository-Hosting-Service für private Repositories oder kostenlose Konten, die üblicherweise als Host für Open-Source-Software-Projekte verwendet werden (siehe http://github.com/).

GitLab, ein webbasiertes DevOps-Lifecycle-Tool, das einen Git-Repository-Manager mit Wiki, Issue-Tracker und CI/CD-Pipeline-Funktionen unter einer Open-Source-Lizenz bereitstellt (siehe http://gitlab.com/).

Jira, ein von Atlassian Pty Ltd. hergestelltes kommerzielles Issue-Management-System (siehe http://www.atlassian.com/).

Mantis, ein kostenloses Open-Source Issue-Management-System (siehe http://www.man-tisbt.org/).

Redmine, ein kostenloses Open-Source Issue-Management-System (siehe http://www.redmine.org/).

Trac, ein kostenloses Open-Source Issue-Management-System (siehe http://trac.edgewall. org/).

YouTrack, ein von JetBrains hergestelltes kommerzielles Issue-Management-System (siehe https://www.jetbrains.com/youtrack/).



Externe Konfiguration notwendig

Einige Issue-Management-Systeme müssen konfiguriert werden, bevor sie erfolgreich an Klaros-Testmanagement angebunden werden können. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 3.14</u>, <u>"Externe Issue-Management-Systeme konfigurieren"</u>.



Token-basierte Authentifizierung

Manche Issue-Management-Systeme wie Jira, GitHub, GitLab, Redmine und YouTrack benötigen eine Token-basierte Authentifizierung. Die Authentifizierung über Benutzername und Passwort werden in diesem Fall nicht unterstützt.

Klaros-Testmanagement unterstützt ebenfalls das Token-basiertes Authentifizierungsschema. Verwenden Sie einfach den erforderlichen Benutzernamen und das Token als Passwort.

= 📌 K L A R	d s test management		DE Finanz-Tracker 🛛 📑	२ × छ ∽ ≛ ×
🕑 Definieren	Integration			
北 Planen	Issue Management (8) Anforderungsma	nagement (1) E-Mail Netzwerk LDAP	CAS	
🏟 Ausführen	 ID ♦ ID ♦ ID ♦ ID ♦ 	8 Einträge - Seite 1 von 1 🖌 🚽 1 ojekt-ID 🗘 URL 🗢	▶ N 10 ▼ Beschreibung ≑ Projekte €	✓ Alle anzeigen 22 Ξ Issues Aktion
🚯 Auswerten	IM00017 Aredmine Playground	http://dev-redmine.verit.de/ http://infosys.verit.de/jira6		2 மீற்மீறி 1 மீற்மீற்
🔑 Konfigurieren	☐ IM00015 A ♀ Jira (legacy) ☐ IM00014 → Mantis MANTISTE ☐ IM00013 ◯ GitHub	http://intosys/jira/ ST http://dev-mantis2.verit.de https://github.com/susp248/ktmtest	Issue Management Integ Issue Management Integ	0 亿已区Ш ration 1 亿包岔面 ration 2 乙包闭面
	IM00012 🖊 GitLab	https://dev- gitlab.verit.de/root/awesomeproject	Gitlab - test project Issue Management Integ PRV PRV 2 Test PRV	ration 4 🗹 🖻 🗭 🛍
	IM00010 🔥 Redmine financetrac	ker http://dev-redmine40.verit.de/		• ビゼピ⑪
	🗌 IM00008 🦉 Jira (legacy) PLAYGROU	ND http://infosys/jira/	Klaros DE Finanz-Tracker Finance Tracker FinanceTrackerLocalTest	36 🗹 🖻 🗹 🗓 ter
	ID System Externe F	rojekt-ID URL	Beschreibung Projekte	Issues Aktion
	Neu			Speichern Verwerfen

Abbildung 10.19. Die Ansicht "Issue Management"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.	
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.	
	$\underline{\Lambda}$ es wurden noch keine Anmeldeinformationen zur Hintergrundsynchronisation hinterlegt.	
System	Das verwendete Issue-Management-System (Bugzilla, Jira (Version 4.3 oder höher), Jira Legacy (Version 4.2 oder niedri- ger), GitHub, GitLab, Mantis, Redmine und Trac).	
URL	Der Link zum Issue-Management-System.	
Externe Projekt-ID	Die Projekt-ID des Issue-Management-Systems. Siehe Abschnitt 10.5.1.1, "Externe Projekt-ID".	
Beschreibung	Zusätzliche Informationen zum Issue-Management-System.	
Projekte	Die Projekte in denen das Issue-Management-System verwen- det wird.	
Aktionen	Die ausführbaren Aktionen.	

10.5.1.1. Externe Projekt-ID

Falls das Issue-Management-System Issues in Projekten organisiert, wird die Projekt-ID benötigt, um das jeweilige Projekt anzusprechen. Jira, Redmine, Mantis, Bugzilla und YouTrack verwenden eine solche Projekt-ID. Bei GitHub, GitLab und Trac wird die URL verwendet, um unterschiedliche Projekte anzusprechen.

Die gültigen Projekt-ID-Werte finden Sie in den Issue-Management-Systemen an den folgenden Stellen:

10.5.1.1.1. Bugzilla

Die Bugzilla-Projekt-ID entspricht dem Feld Product

Edit De product	escription
TestProduct Th in:	his is a test product. This ought to be blown away and replaced with real stuff in a finished istallation of bugzilla.

Abbildung 10.20. Die Bugzilla-Projekt-ID

10.5.1.1.2. Jira

Die Jira-Projekt-ID entspricht dem Feld Key



Abbildung 10.21. Die Jira-Projekt-ID

10.5.1.1.3. Mantis

Die Mantis-Projekt-ID entspricht dem Feld Project Name

	TB		
Logged in as: admin (Mantis	Administrator - administrator)		2011-05-27 10:20 CEST
	Main My View View Issu	ies <u>Repo</u>	rt Issue Change Log Roadmap Summary Docs
	[<u>Mana</u>	<u>ge Users</u>]	[Manage Projects] [Manage Custom Fields] [Man.
	Edit Project		
	Project Name	<	Playground
	Status		development 💌
			-

Abbildung 10.22. Die Mantis-Projekt-ID

10.5.1.1.4. Redmine

Die Redmine-Projekt-ID entspricht dem Feld Identifier

Home My page Projects Administration Help
MyAwesomeProject
+ Overview Activity Issues Spent time Gantt Calendar News Documents Wiki Files Settings
Settings
Project Members Issue tracking Versions Issue categories Repositories Forums Time tracking
Name * MyAwesomeProject
Description Edit Preview B I U S C HI H2 H3 🗄 🗄 🗐 Pre 💠 📾 🖷 🚱
Talanahiting W
Identifier * awesome
Homepage
Public ∉ Public projects and their contents are available to all logged-in users.
Inherit members

Abbildung 10.23. Die Redmine-Projekt-ID

10.5.1.1.5. YouTrack

Die YouTrack-Projekt-ID entspricht dem Feld ID

YouTrack	Issues v Dashboards Agile Boards Reports Projects More v ⑦ O _
,	Clone project Archive project Delete project
Settings Access	Team Fields VCS Notifications Build Server Integration Time Tracking Workflow
Name	Demo project
ID	DEMO
	Will be used as issue ID prefix
Project owner	Administrator (root) v 3
Logo	DEM Reset to default
	Upload a JPG, GIF, or PNG file. The image is resized to 48 × 48 pixels automatically
Custom project template	
Default visibility setting	Select an option v
	Sets the default visibility for new issues and articles in the project
Recommended visibility options	Project Default •
Description	
Description	Normal text v B I 5 A v III

Abbildung 10.24. Die YouTrack-Projekt-ID

10.5.1.2. Ein neues Issue-Management-System hinzufügen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Hier können Attribute des Issue-Management-Systems festgelegt werden.

Mit Klick auf Speichern wird das Issue-Management-System gespeichert. Die ID des Issuemanagement-Systems (IM0001) wird dabei automatisch von Klaros-Testmanagement vergeben.

Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs enthalten Änderungen und sind noch nicht gespeichert!

10.5.1.3. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Öffnet den externen Link zum Issue-Management-System

Die URL auf Gültigkeit prüfen (siehe Abschnitt 10.5.1.5, "Die URL auf Gültigkeit prüfen")

Bearbeiten (siehe <u>Abschnitt 10.5.1.6</u>, <u>"Ein vorhandenes Issue-Management-System bear-</u><u>beiten"</u>)

ff Löschen

10.5.1.4. Tabellenoperationen

In der Zeile rechts oberhalb der Tabelle lassen sich folgende Operationen vornehmen:

√ Filter / Sortieren

Nur aktive / Zeige alle

公 Exportieren

 \equiv Spaltenauswahl

Alle Operationen sind ausführlich in Abschnitt 5.2.3.1, "Übersichtsseite" beschrieben.

10.5.1.5. Die URL auf Gültigkeit prüfen

In der Spalte *Aktion* kann mit Klick auf 🖨 überprüft werden, ob eine Verbindung zum Issue-Management-System hergestellt werden kann.



Schrägstriche am Ende von URLs

Prüfen Sie im Falle eines Authentifizierungsfehlers zunächst, ob das Hinzufügen oder Löschen eines nachgestellten Schrägstrichs an die URL das Problem behebt.

10.5.1.6. Ein vorhandenes Issue-Management-System bearbeiten

Jedes angelegte Issue-Management-System verfügt über eine eigene Detailseite mit weitergehenden Unteransichten. Mit Mit Klick auf die ID des betreffenden Issue-Management-Systems oder auf das Icon
Bearbeiten rechts in der Aktionsspalte gelangen Sie zu der als letztes ausgewählten Detailansicht. Bei erstmaligem Aufruf ist dies die Ansicht *Eigenschaften*.

Es stehen die folgenden Ansichten zur Verfügung: Eigenschaften, Anhänge, Projekte und Änderungen.

```
10.5.1.7. Aktionen
```

Auf den Detailseiten befinden sich rechts oben weitere Icons. Folgende Aktionen lassen sich hier vornehmen:

🗹 Externen Link öffnen	Öffnet	einen	externen	Link	zum	entsprechenden	Issue-
	Manag	ement-	System.				



Mit den grünen Pfeilen ganz oben rechts kann zwischen den auf der vorherigen Seite vorhandenen Issue Management Systemen gewechselt werden.

10.5.1.8. Eigenschaften

. ≡ 🞐 K L A R) s test management	DE Finanz-Tracker 👕 🔜	<i>م</i> ×	0~ 1 ~
📝 Definieren	IM00017 - Redmine, Playground			200
🏝 Planen	Elgenschaften Anhänge Projekte Issues (2) Änderungen		Speichern Verwerfen	Zurück
🏟 Ausführen	System Redmine V			
😫 Auswerten	VRL http://dev-redmine.verit.de/			
🔑 Konfigurieren	Authentifizierung			
	Bier geben Sie die Anmeldeinformationen für die Hintergrundsynchronisierung ein. Der hier eingetragene Benutzer muss die Administrator Rolle besitzen			×
	Benutzername admin Passwort Authentifizierung testen Statusnamen			
	Die in der folgenden Tabelle aufgeführten Statusnamen entsprechen einem erledigten issue. Der angegebene Wert muss exakt mit dem Statusnamen im angebundenen System übereinstimmen.			×
	Name			Aktion
	Closed			Ū.
	Rejected Resolved			ش
	Neu			Voreinstellung
	Erstellt vor 3 Jahren von Felix Mustermann		Zuletzt geändert vor 3 Jahren von	Felix Mustermann
			Speichern Verwerfen	Zurück

Abbildung 10.25. Die Ansicht "Eigenschaften"

Die Ansicht *Eigenschaften* ist in drei Bereiche aufgeteilt: Issue Management System, Authentifizierung, Statusnamen.

Issue-Management-System

System	Mit Hilfe eines Dropdown-Menüs kann das gewünschte Sys- tem ausgewählt werden.
URL	Die URL, unter der das System erreichbar ist.
Projekt	Der Name des Projektes.
Beschreibung	Eine optionale Beschreibung des Systems.

Darunter befindet sich der Abschnitt "Authentifizierung". Hier geben Sie die Anmeldeinformationen für die Hintergrundsynchronisierung ein. Der hier eingetragene Benutzer muss vollen Zugriff auf das eingetragene Projekt haben.

Benutzername	Der gültige Benutzername.
Passwort	Das gültige Passwort.

Mit dem Button Anmeldevorgang testen kann geprüft werden, ob die Eingaben korrekt sind.



Wichtig

Die hier eingegebenen Anmeldeinformationen werden ausschließlich für die Hintergrundsynchronisation verwendet. Wenn Benutzer ein Issue erstellt oder aktualisiert, werden dessen persönliche Anmeldedaten zur Authentifizierung gegenüber dem Issue-Management-System verwendet. Wenn Anmeldeinformationen fehlen oder nicht gültig sind, werden Benutzer aufgefordert, diese einzugeben.

Der letzte Bereich ist "Statusnamen". Die in der folgenden Tabelle aufgeführten Statusnamen entsprechen einem erledigten Issue. Der angegebene Wert muss exakt mit dem Statusnamen im angebundenen System übereinstimmen.

≡ 🞐 к L А F	ROS TEST MANAGEMENT	DE Finanz-Tracker 📓 🔜	Q 0~ ⊥ ~
📝 Definieren	Integration		
📇 Planen	Issue Management Anforderungsmanagement E-Mail Netzwerk LDAP CAS		
	6 Einträge - Seite 1 von 1 🕅 ┥ 🚺 🕨 🗎 🔟 🗸	·	γ Alle anzeigen 🕹 \equiv
Austuhren	ID	Projekt 🖨 🛛 Be	schreibung 🗢 🛛 Aktion
🕒 Auswerten	M00014 💓 Mantis http://dev-mantis.verit.de/mantis	MANTISTEST	5 C 11 5 C 11
🔑 Konfigurieren	Hier geben Sie die Anmeldeinformationen für die Hintergrundsynchronisierung ein. Der hier eingetragene Benutzer muss vollen Zugriff auf das eingetragene Projekt haben.	×	9 C 11 9 C 11
* Konigunaton	Popularmana usa		9 C 11 9 C 11
	Passwort		
	Anmeldevorgang testen		Speichern Verwerfen
	Die in der folgenden Tabelle aufgeführten Statusnamen entsprechen einem erledigten Issue. Der angegebene Wert muss exakt mit dem Statusnamen im angebundenen System übereinstimmen.	×	
	Name	Aktion	
	resolved		
	Neu	Voreinstellung Verwerfen	
		OK Abbrechen	

Abbildung 10.26. Bearbeiten der Status-Zuordnungen

Mit dem Button Neu kann ein neuer Statusname hinzugefügt werden.

Mit dem Button Voreinstellung werden die Felder auf die Standardwerte zurückgesetzt.

Alle Eingaben werden mit dem Button Speichern in die Datenbank übernommen. Der Button Verwerfen löscht die Eingaben. Der Button Zurück führt auf die Übersichtseite mit den angelegten Issue-Management-Systemen.

10.5.1.9. Anhänge

Einem Issue-Management-System können beliebige Dateien als Anhänge hinzugefügt werden. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 5.2.3.2.6, "Anhänge"</u>.

10.5.1.10. Projekte

Die Ansicht *Projekte* zeigt alle Projekte an, denen das Issue-Management-System zugeordnet wurde.

10.5.1.11. Änderungen

Die Ansicht Änderungen zeigt die Änderungshistorie dieses Issue Management Systems an.

Eine ausführliche Beschreibung der Ansicht Änderungen finden Sie unter <u>Abschnitt 5.2.3.2.8,</u> "Änderungshistorie".

10.5.1.12. Zuordnung der Statuswerte für erledigte Issues

Jedes Issue hat ein Statusfeld, das seinen aktuellen Zustand dokumentiert (z.B. "In Bearbeitung" oder "Behoben"). Die einzelnen Status und deren Namensgebung variieren je nach verwendetem Issue-Management-System. Klaros-Testmanagement unterscheidet nur zwischen offenen und erledigten Issues. Ein Issue gilt dann als erledigt, wenn keine weitere Arbeit dafür aufgewendet werden muss (z.B. "Geschlossen", "Abgelehnt"), andernfalls ist es noch offen.

Erledigte Issues werden in Grün dargestellt, offene in Rot. Abschnitt 9.6.5, "Detailseite"

Die Zustände der erledigten Issues müssen genau so benannt werden wie im verwendeten Issue-Management-System, sonst werden sie nicht richtig identifiziert und als offene Issues gezählt.

10.5.2. Anforderungsmanagement

Anforderungen können entweder lokal in Klaros-Testmanagement oder in einem externen System verwaltet werden.

Lokales Anforderungsmanage- ment	Standardmäßig können Test-Manager Anforderungen für ein Projekt aus dem Projekt heraus erstellen, bearbeiten und löschen.
Externes Anforderungsmanage- ment	Ein Projekt kann seine Anforderungen auch von einem exter- nen Anforderungsmanagementsystem (RMS) erhalten. Ist dies konfiguriert, kann das Erstellen, Bearbeiten und Löschen von Anforderungen nur über das angeschlossene RMS erfol- gen.

Derzeit unterstützt Klaros-Testmanagement die folgenden externen Anforderungsmanagementsysteme:

Jira 4.3 oder neuer	Eine von	Atlassian	Pty	Ltd.	veröffentlichte	kommerzielle
	Projektma	anagement-	Soft	ware,	siehe http://ww	ww.atlassian.
	com/.					

= 📌 K L A R) s test management	DE Finanz-Tracker 🧧 📑	२ × 0 × ≛ ×
🗭 Definieren	Integration		
北 Planen	Issue Management (8) Anforderungsmanagement (1) E-Mail Netzwerk LDAP	CAS	
🌣 Ausführen	I Einträge- Seite 1 von 1 M 1 ID + Q System + URL + Externe Projekt-ID +	► N 10 ▼ Beschreibung ♦ Projekte ♦	✓ Alle anzeigen Alle anzeigen Anforderungen Aktion
Auswerten	RM00001 🛆 🔷 Jira http://infosys.verit.de/jira6 SYNC	Printer Tester	4 C C C D
	ID System UHL Externe Projekt-ID	Beschreibung Projekte	Anforderungen Aktion
🔑 Konfigurieren	Neu		Speichern Verwerfen

Abbildung 10.27. Die Ansicht "Anforderungsmanagement"

Die Tabelle zeigt die folgenden Werte:

ID	Wird automatisch zugewiesen.				
Zusätzliche Hinweise	Ein Tooltip erscheint, wenn der Cursor über dem hier angezeig- ten Icon platziert wird.				
	合 das Anforderungsmanagement-System kann nicht gelöscht werden, da es mit mindestens einem Projekt ver- knüpft ist.				
	$\underline{\Lambda}$ es wurden noch keine Anmeldeinformationen zur Hintergrundsynchronisation hinterlegt.				
System	Das verwendete Anforderungsmanagement-System (Jira Version 4.3 oder höher).				
URL	Der Link zum Anforderungsmanagement-System.				
Externe Projekt-ID	Siehe Abschnitt 10.5.1.1, "Externe Projekt-ID".				
Beschreibung	Eine kurze Beschreibung des Anforderungsmanagement-Sys- tems.				
Projekte	Die Projekte in denen das Issue-Management-System verwen- det wird.				
Aktionen	Die ausführbaren Aktionen.				

10.5.2.1. Ein neues Anforderungsmanagement-System hinzufügen

Mit einem Klick auf den Button Neu erscheint eine neue leere Tabellenzeile. Hier können alle Attribute des Anforderungsmanagement-Systems festgelegt werden.

Mit Klick auf <u>Speichern</u> wird das Anforderungsmanagement-System gespeichert. Die ID des Anforderungsmanagement-Systems (RM0001) wird dabei automatisch von Klaros-Testmanagement vergeben. Mit Neu können auch mehrere Tabellenzeilen gleichzeitig erzeugt und bearbeitet werden. Erst der Klick auf Speichern übernimmt die Daten in die Datenbank.

Mit Verwerfen werden alle Änderungen rückgängig gemacht.



Rote IDs

Alle Zeilen mit roten IDs enthalten Änderungen und sind noch nicht gespeichert!

10.5.2.2. Aktionen

Die Aktionsspalte befindet sich ganz rechts in der Tabelle. Folgende Aktionen lassen sich hier vornehmen:

Die URL auf Gültigkeit prüfen (siehe Abschnitt 10.5.1.5, "Die URL auf Gültigkeit prüfen")

 ${\boldsymbol{\mathcal{C}}}$ Anforderungen synchronisieren

Bearbeiten (siehe <u>Abschnitt 10.5.2.3, "Ein vorhandenes Anforderungsmanagement-System</u> <u>bearbeiten"</u>)

🗊 Deaktivieren



Anhänge synchronisieren

Besteht eine Verbindung zu einem Jira Server, werden Anhänge aus dessen Anforderungen bei der Synchronisation automatisch nach Klaros-Testmanagement übertragen.

10.5.2.3. Ein vorhandenes Anforderungsmanagement-System bearbeiten

Mit Klick auf das Icon 😰 rechts in der Aktionsspalte öffnet sich ein Dialog. Hier können die Anmeldeinformationen für die Hintergrundsynchronisation hinterlegt werden.

10.5.3. E-Mail

In der Ansicht *E-Mail* werden die Mail-Server-Settings der Anwendung konfiguriert. Diese Einstellungen werden für den Versand von Benachrichtigungs-E-Mails benötigt (siehe <u>Abschnitt 10.4.2</u>, <u>"Benachrichtigungen"</u>). Die Attribute, die bearbeitet werden können, werden in <u>Abbildung 10.28</u> gezeigt.

≡	👱 KLAR	OS TEST MA	A N A G E M E N T			DE Finanz-Tracker 👕 📑	५ x 0⁄ ≗ ×
ľ	Definieren	Integration					
:	Planen	Issue Management	(8) Anforderungsmanagement (1)	E-Mail Netzwerk	LDAP CAS		
\$	Ausführen	SMTP-Server SMTP-Server-Port	localhost 25				
¢	Auswerten	Absender-Adresse Authentifizierung	keine Authentifizierung 💙				
×	Konfigurieren	Sicherheit	Keine				
			SSL				
			Test-E-Mail versenden				
							Speichern Verwerfen

Abbildung 10.28. Die Ansicht "E-Mail"



Anmerkung

Wenden Sie sich im Zweifelsfall an Ihren Systemadministrator, welche Werte Sie hier angeben müssen.

SMTP-Server	Der Hostname Ihres Mail-Servers.
SMTP-Server-Port	Der Port, über den Ihr Mailserver erreichbar ist. Dieser Wert hängt normalerweise von der unten gewählten Einstellung für die Transportsicherheit ab. Der Standardwert bei keiner Trans- portsicherheit ist 25.
Absender-Adresse	Die Absender-Adresse der erzeugten E-Mails.
Benutzername	Der Benutzername, der zur Authentifizierung gegenüber dem SMTP-Server benötigt wird. Dieses Feld ist nur sichtbar, wenn die SMTP- oder POP-Authentifizierung im Feld <i>Authentifizierung</i> ausgewählt wurde.
Passwort	Das Passwort, das zur Authentifizierung gegenüber dem SMTP-Server benötigt wird. Dieses Feld ist nur sichtbar, wenn SMTP- oder POP-Authentifizierung im Feld <i>Authentifizierung</i> ausgewählt wurde.
Authentifizierung	Um Spam zu verhindern oder zu identifizieren, verlangen eini- ge Mailserver von den Benutzern eine Authentifizierung, bevor sie ihnen das Versenden von Mails erlauben. Mit dieser Opti- on wird ausgewählt, ob eine Authentifizierung erforderlich ist oder eine Benutzername/Passwort-Kombination über SMTP oder POP verwendet wird. Wenn eine der beiden letztgenann- ten Optionen gewählt wird, werden die Felder <i>Benutzername</i> und <i>Passwort</i> in der Benutzeroberfläche angezeigt.

Sicherheit	Diese Einstellung definiert die Sicherheit der Transportschicht, die für den Zugriff auf den SMTP-Mail-Server verwendet wird. Jede Transportsicherheitseinstellung kann einen ande- ren Wert für den Port erfordern. Die Optionen sind <i>Keine</i> (Port 25), <i>Secure Socket Layer / SSL</i> (Port 465) oder <i>Transport Layer</i> <i>Security / TLS</i> aka STARTTLS (Port 25). Ihre lokalen Port-Ein- stellungen können von den angegebenen Standardeinstellun- gen abweichen.
Test-E-Mail senden	Dieser Link sendet eine Test-E-Mail, um zu prüfen, ob die E- Mail-Server-Einstellungen gültig sind. Es ist daher erforderlich, dass in Ihrem Benutzerkonto eine gültige E-Mail-Adresse ange- geben ist.

Klicken Sie auf Test-E-Mail versenden, um zu sehen, ob die Felder korrekt ausgefüllt sind.

10.5.4. Netzwerk

Klaros-Testmanagement unterstützt die Verwendung von HTTP- und SOCKS-Proxies für externe Netzwerkverbindungen. Proxy-Einstellungen werden in der Ansicht *Netzwerk*.

≡	y KLAR	os test mana	GEMENT			DE Finanz-Tracker 🛅 📑	Q x Ø~ ≛~
ľ	Definieren	Integration					
***	Planen	Issue Management (8)	Anforderungsmanagement (1)	E-Mail Netzwerk	LDAP CAS		
۵	Ausführen	Proxy-Host Port	proxy.verit.de 3128				
Ģ	Auswerten	Kein Proxy für Proxy-Art	HTTP O SOCKS				
×	Konfigurieren	Benötigt Authentifizierung	Proxy-Einstellung prüfen				
							Speichern Verwerfen

Abbildung 10.29. Die Ansicht "Netzwerk"



Festlegen der Anwendungs-URL

Wenn Klaros-Testmanagement hinter einem Proxy ausgeführt wird, muss die Anwendungs-URL entsprechend gesetzt werden, damit Klaros-Testmanagement ordnungsgemäß funktioniert. Die Anwendungs-URL kann in der Ansicht *Verschiedenes* unter *Allgemeine Einstellungen* eingestellt werden (siehe <u>Abschnitt 10.4.1, "Sonstiges"</u>).

Die Eigenschaften der Proxy-Einstellungen sind:

Proxy Host	Die Netzwerkadresse oder der Hostname des Proxyservers.
Port	Der Port, auf dem der Proxyserver aktiv ist.

Kein Proxy für	Eine durch getrennte Liste von Domänen, die die Proxy-Ein- stellungen umgehen sollen.						
Ргоху-Тур	Mögliche Einstellungen sind HTTP-Proxy oder SOCKS-Proxy.						
Authentifizierung erforderlich	Gibt an, ob der Proxy eine Authentifizierung erfordert oder nicht.						
Benutzername	Der Benutzername, der für die Benutzername/Pass- wort-Authentifizierung verwendet werden soll.						
Passwort	Das Passwort, das für die Benutzername/Passwort-Authenti- fizierung verwendet werden soll.						

Die Proxy-Einstellungen können durch Klick auf den Button Proxy-Einstellungen prüfen überprüft werden.

10.5.5. LDAP





Tipp

Um auf einen LDAP- oder Active Directory-Server zuzugreifen, müssen zuerst verschiedene Konfigurationsparameter eingestellt werden. Wenden Sie sich im Zweifelsfall an Ihren Systemadministrator, welche Werte Sie hier angeben müssen.

= 👱 K L A R	os test man,	AGEMENT			DE Finanz-Tracker 📑		Q X	@ ~	* ~
🕑 Definieren	Integration								
北 Planen	Issue Management (8)	Anforderungsmanagement (1) E-Mail	Netzwerk LD/	AP CAS					
 Ausführen Auswerten 	Server Address Server Port Gesichert (LDAPS) Bind DN Bind Credentials	Idap1.verit.de 389 cn=Manager,dc=verit.dc=de		Einfac	User Context DN User Object Classes her DN Matching-Modus User DN Prefix User DN Suffix	ou=Users,dc=verit,dc=de person,posixAccount uid= ,ou=Users,dc=verit,dc=de			
	Referrals folgen User Search Attribute User Name Attribute	uid			Full Name Attribute	cn			
	User Password Attribute	userPassword			E-Mail Attribute	mail			
	Login Voreinstellung	값 LDAP-Zugriff testen	Autor	natische Benutzern Passwordsynd	egistrierung unterbinden chronisation unterbinden	S	eichern	Verwe	erfen

Abbildung 10.30. Die Ansicht "LDAP"

Parameter, die zur Kontaktaufnahme mit dem LDAP-Server erforderlich sind:

Server-Adresse

Die URI, unter dem sich der LDAP-Server befindet (z.B. 1dap. acme.com).

Server Port	Der Port, auf dem sich der LDAP-Server befindet (typischerweise 389 für unverschlüsselte Verbindungen oder 686 für LDA- PS).			
Secure (LDAPS)	Wenn aktiviert, wird das verschlüsselnde LDAPS-Protokoll ver- wendet.			
Bind DN	Der Distinguished Name des Benutzerkontos, das sich am Ser- ver anmeldet und den Bindungsvorgang ausführt.			
Bind Credentials	Die Kennwortberechtigung des Benutzerkontos, das sich am Server anmeldet und den Bindungsvorgang ausführt.			
Follow Referrals	Wenn aktiviert, folgt die Suche in einem Verzeichnis automa- tisch jedem Verweis auf andere LDAP-Server, den der Ser- ver möglicherweise zurückgibt. Wenn die Verweise deaktiviert sind, werden sie ignoriert und es werden keine weiteren Server während der Suche kontaktiert.			
Parameter, die zum Auffinden eine	es Benutzerkontos benötigt werden:			
User Context DN	Der Distinguished Name des Pfades, unter dem nach Benut- zerkonten gesucht wird (z.B. ou=Benutzer,dc=verit,dc=de).			
User Object Classes	Eine durch Kommata getrennte Liste der LDAP-Objektklassen, denen ein Benutzerkonto entsprechen muss, um in die Suche einbezogen zu werden (z. B. person, posixAccount).			
Enable naive DN Matching Mode	Wenn diese Option aktiviert ist, wird der zurückgegebene DN einer Benutzersuche direkt zur Authentifizierung des Benut- zers verwendet. Wenn diese Option deaktiviert ist, werden die folgenden zwei Parameter in Verbindung mit dem <i>User Name</i> <i>Attribute</i> verwendet, um einen DN zur Authentifizierung zu erstellen.			
User DN Prefix	Das Präfix des Distinguished Name, der zur Lokalisierung von Benutzerkonten verwendet wird (z.B. uid=).			
User DN Suffix	Das Suffix des Distinguished Name, der zum Auffin- den des Benutzerkontos verwendet wird (z. B. ,ou=- Users,dc=acme,dc=com). Beim Auffinden von Benutzerkonten werden das Präfix, die Konto-ID und das Suffix miteinander verkettet, um den Distinguished Name des Benutzerkontos zu bilden.			
Parameter, die die Attribute eines Benutzerkontos beschreiben:				
User Search Attribute	Das LDAP-Benutzernamensattribut, das dem Klaros-Test- management-Kontoname entspricht (z.B. uid).			

User Name AttributeDas LDAP-Attribut, das in der DN-Bind-Aktion verwendet wird,
die den Benutzer authentifiziert (wenn der naive DN-Matching-

Modus deaktiviert ist). In einfachen Szenarien wird dies mit dem *Benutzersuchattribut* übereinstimmen.

Wenn Ihr LDAP-Server-Setup es nicht erlaubt, einen Benutzer mit dem angegebenen Benutzer-Suchattribut zu binden, sollten Sie hier das entsprechende Attribut angeben (z. B. cn) und dieses in Verbindung mit dem entsprechenden Benutzer-DN-Präfix/Suffix verwenden.

- User Password Attribute Das LDAP-Passwortattribut, das dem Kontopasswort in Klaros-Testmanagement entspricht (z. B. userPassword).
- Full Name AttributeDas LDAP-Attribut, das den vollständigen Namen des Benut-
zers enthält (z. B. cn). Wenn angegeben, wird dieser bei der ers-
ten erfolgreichen Anmeldung automatisch in die Klaros-Test-
management Datenbank übertragen.
- Email AttributeDas LDAP-Attribut, das die E-Mail-Adresse des Benutzerkon-
tos enthält (z.B. mail). Falls angegeben, wird dies automatisch
bei der ersten erfolgreichen Anmeldung in die Datenbank von
Klaros-Testmanagement übertragen.
- Enabled AttributeWenn angegeben, definiert dieses boolesche LDAP-Attribut,
ob sich ein Benutzer anmelden darf. Nicht alle Verzeichnisser-
ver bieten ein solches Attribut an.

Wenn *Login Voreinstellung* aktiviert ist, wird der Anmeldebildschirm standardmäßig für alle Benutzer auf LDAP-Authentifizierung eingestellt. Es ist weiterhin möglich, dass sich bestehende Benutzer gegen die klaros; Benutzerdatenbank authentifizieren, wenn dies im Anmeldebildschirm ausgewählt wird.

Verwenden Sie die Option Automatische Benutzerregistrierung unterbinden, um das automatische Anlegen eines Benutzers in Klaros-Testmanagement bei der ersten erfolgreichen Anmeldung zu verhindern.

Bei der ersten erfolgreichen Anmeldung wird ein passender Passwort-Hash in der lokalen Benutzerdatenbank erstellt, sodass sich die Benutzer mit ihrem LDAP-Passwort auch gegenüber der lokalen Benutzerdatenbank authentifizieren können. Wenn die Option *Passwortsynchronisierung deaktivieren* aktiviert ist, wird diese Synchronisierung nicht durchgeführt und Benutzer können sich nur dann lokal anmelden, wenn ihnen ein Administrator interaktiv ein lokales Passwort zuweist.

Mit *LDAP-Zugriff testen* wird getestet, ob die auf dieser Seite eingegebenen Parameter korrekt sind. Der Testprozess ist in zwei Phasen unterteilt:

In der ersten Phase wird versucht, eine Anmeldung am Server und eine Suche nach allen verfügbaren Benutzern durchzuführen. Wenn dies erfolgreich ist, wird ein Dialog angezeigt, der die im LDAP-Verzeichnis gefundenen Benutzer auflistet.

In der zweiten Phase können ein Benutzername und ein Passwort eingegeben werden, um die LDAP-Anmeldung eines Benutzers zu testen. Das Ergebnis des Anmeldeversuchs wird im Log-Panel protokolliert.

≡ 🎐 K L A R 0	DS TEST MANA	GEMENT DE Finanz-Tracker 🖬 📰	વ × છ~ ≜×
📝 Definieren	② 11:51:17 Der LDAP-Zug	riff war erfolgreich, es wurden 34 Benutzer gefunden	+ x
	Integration		
Aug fillerer	Issue Management (8)	Anforderungsmanagement (1) E-Mail Netzwerk LDAP CAS	
Ausrunren	Server Address	ldap1.verit.de User Context DN ou=Users,dc=verit,dc=de	
🕒 Auswerten	Server Port	389 User Object Classes person, posixAccount	
	Gesichert (LDAPS)	Einfacher DN Matching-Modus	
🔑 Konfigurieren	Bind DN Bind Credentials	cn=Manager,dc=verit,dc=de User DN Suffix uid=	
	Referrals folgen	Authentifizierung benöttigt für LDAP	
	User Search Attribute User Name Attribute User Password Attribute Login Voreinstellung	ud 34 Einträge gefunden stolp ud Bitte geben Sie einen gültigen Benutzername und Passwort ein, um die Authentifizierung am LDAP-Server zu testen. Attribute mail Benutzername selen_DE Passwort terbinden einen ein	
		C Lora cognitioner	

Abbildung 10.31. Der Dialog "LDAP-Authentifizierung"

10.5.6. CAS



Klaros-Testmanagement unterstützt Single Sign-on-Authentifizierung (SSO) unter Verwendung eines zentralen Authentifizierungsdienstes (CAS).

≡ 📌 K L A R	os test management		DE Finanz-Tracker 🔤 📑	द x 0 ~ ≛~
📝 Definieren	Integration			
😤 Planen	Issue Management (8) Anforderungsmanag	ement (1) E-Mail Netzwerk LDAP CAS		
🏟 Ausführen	CAS Server URL Prefix CAS Server Login URL	https://192.168.56.2/cas https://192.168.56.2/cas/login		
🕒 Auswerten	CAS Server Logout URL Automatische Benutzerregistrierung unterbinden	https://192.168.56.2/logout		
🄑 Konfigurieren	CAS Authentifizierung aktivieren			
				Speichern Verwerfen



Parameter, die zur Kontaktaufnahme mit dem CAS-Server erforderlich sind:

CAS Server URL Prefix	Die URL, unter der sich der CAS-Server befindet (z.B. cas.acme. com).
CAS Server Login URL	Die URL des CAS-Servers für Single Sign-on-Anmeldungen. Ein Benutzer, der noch nicht von CAS authentifiziert wurde, wird auf diese URL umgeleitet.

CAS Server Logout URL	Die URL des CAS-Servers für Single Sign-on-Abmeldung. Wenn sich ein Benutzer von Klaros-Testmanagement abmeldet, wird er zu dieser URL umgeleitet.
Automatische Benutzerregis- trierung unterbinden	Wenn aktiviert, wird bei erfolgreicher CAS-Authentifizierung nicht automatisch ein Gastbenutzerkonto im System erstellt.
CAS-Authentifizierung aktivie- ren	Wenn aktiviert, wird die gesamte Authentifizierung an den CAS-Server delegiert und das Standard-Anmeldefenster ist nicht mehr erreichbar. Um die Aktivierung abzuschließen, ist ein Neustart des Anwendungsservers erforderlich.



CAS Single Sign-on deaktiviert die lokale Anmeldung

Sobald die CAS-Unterstützung aktiviert ist und Klaros-Testmanagement neu gestartet wurde, ist es nicht mehr möglich sich auf dem Anmeldebildschirm von Klaros-Testmanagement anzumelden.

Bitte überprüfen Sie vor der Aktivierung, ob die Kontoüberprüfung funktioniert. Im Falle einer fehlerhaften Einrichtung setzen Sie bitte die Eigenschaft cas.enabled in der Konfigurationsdatei %KLAROS_HOME%/.klaros/klaros.properties auf false und starten Sie den Anwendungsserver neu. Dadurch wird das Standardverhalten wiederhergestellt.

10.6. Sicherung

Auf der Seite *Sicherung* lassen sich komplette Projekte im XML-Format sichern und wiederherstellen. Nur Administratoren und Manager sind dazu berechtigt, Projekte zu importieren oder zu exportieren.

10.6.1. Export

In dieser Ansicht können ein oder mehrere Projekte ausgewählt und mit Klick auf den Button Exportieren exportiert werden (<u>Abbildung 10.33</u>).

= 📌 KLAR	o s test management			DE Finanz-Tracker			Q X	0×	- ~
📝 Definieren	Sicherung								
📇 Planen	Export Import								
	9 von 9 ausgewählt	9 Einträge - Seite 1 von 1	K 🖪 🚺 🕨	▶ 10 ¥				V 2	አ =
🔯 Ausführen	✓ ID \$	Beschreibung 🗢			Anforderungen 🗢	Testfälle 🗢	Testsuiten	Testläi	ufe 🗘
	✓ P00011				0	1	0	0	
🕒 Auswerten	 P00010 Test PRV 				0	5	1	3	
	P00009 PRV 2				0	1	0	0	
🄑 Konfigurieren	✓ P00008 PRV				0	3	1	1	
	P00007 Issue Management Integration				0	4	0	0	
	P00005 DE Finanz-Tracker				6	24	7	877	7
	P00003 Printer Tester				4	1	0	0	
	P00002 Finance Tracker				6	24	7	33	8
	P00001 FinanceTrackerLocalTester				6	26	9	20	4
								Exportie	ren
								Exportio	

Abbildung 10.33. Die Ansicht "Export"

Mit Exportieren werden die ausgewählten Projekte in eine XML-formatierte Ausgabedatei exportiert.



Anmerkung

Die Sicherungsdateien sind mit der jeweiligen Datenbankversion gekennzeichnet, die exportiert wurde. Sicherungsdateien können nur dann importiert werden, wenn die Datenbankversion der Klaros-Installation mit der Version der Sicherungsdatei übereinstimmt.

10.6.2. Import

In dieser Ansicht können Projekte aus Sicherungsdateien importiert werden. Mit Klick auf den Button Sicherung hochladen kann eine Sicherungsdatei hochgeladen werden. Nach dem Hochladen werden die Projekte angezeigt, die in der Datei enthalten sind. Eines oder mehrere dieser Projekte können ausgewählt werden (<u>Abbildung 10.34</u>).

≡ к l a r	o s test management	Netzwerkadministration 🔳 📑	۹ 🛛 ۲
📝 Definieren	Sicherung		
📇 Planen	Export Import		
🏚 Ausführen	Version: 5.1.0 Erstellt von Felix Mustermann am 09.04.21, 12:54		×
🕒 Auswerten	Beschreibung 🗢	Anforderungen 🖨	Testfälle 🗘 Testsuiten 🗘 Testläufe 🖨
-	Finanz-Tracker App	6	15 3 0
🔑 Konfigurieren	Finanz-Tracker Web	3	48 4 0
Ronngancien	Druckerverwaltung	5	0 2 0
	Abrechnung (intern)	4	10 1 0
	Netzwerkadministration	12	14 3 0
	Sicherung hochladen		Importieren Verwerfen

Abbildung 10.34. Die Ansicht "Import"

Mit Klick auf den Button Importieren werden die ausgewählten Projekte importiert, mit Klick auf den Button Verwerfen wird die Wiederherstellung abgebrochen.



Anmerkung

Ein Import überschreibt niemals bestehende Projekte oder darin vorhandene Daten.



Wichtig

Beim Versuch, eine Sicherungsdatei aus einer früheren Klaros-Datenbankversion zu importieren, wird dieser Import mit einer entsprechenden Fehlermeldung abgewiesen. Sie müssen diese Sicherungsdatei zunächst mit einer zur Sicherungsdatei passenden Version von Klaros-Testmanagement importieren und die Installation dann auf Ihre aktuelle Version aktualisieren.

Ein Export aus dieser Version ist nun mit Ihrer aktuellen Version kompatibel.
Kapitel 11. Benutzerdefinierte Berichte

edition

Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

Obwohl Klaros-Testmanagement bereits eine Vielzahl von vordefinierten Berichten bietet, reichen diese oft nicht aus, wenn spezielle Anforderungen an den Inhalt oder das Layout vorliegen. Deshalb können mit der Klaros-Testmanagement Enterprise Edition auch beliebige eigene Berichte erstellt werden. <u>Abbildung 11.1, "Ablauf der Berichtserstellung"</u> gibt einen Überblick, wie Berichte erstellt werden.



Abbildung 11.1. Ablauf der Berichtserstellung

Am Erstellen eines Berichtes sind zwei Benutzergruppen beteiligt: Berichtdesigner und Benutzer.



Programmierkenntnisse

Für das Erstellen von eigenen Berichte sind grundlegende Kenntnisse der Programmiersprachen Java oder Groovy sowie der XML-basierten Template-Erstellung mit SeamPDF oder SeamExcel notwendig.

- Der **Berichtsdesigner** legt die Parameter eines Berichtes fest und definiert welche Daten hierfür aus der Datenbank herangezogen werden sollen. Außerdem legt er das Layout und Inhalt der Berichtsvorlage fest.
- Der **Benutzer** wählt die Parameter aus, mit denen der Bericht erstellt werden soll und kann den Bericht als PDF- oder Excel-Datei abrufen.
- Die **Layout-Vorlage** definiert das Layout und den Inhalt des Berichts und ist in einer XML-basierten Beschreibungssprache verfasst.
- Das Berichts-Skript extrahiert die Daten aus der Datenbank und bereitet f
 ür die Verwendung in der Berichtsvorlage auf. Das Skript kann in Groovy oder Java implementiert werden und stellt eine Klasse bereit, die die Schnittstelle <u>Abschnitt E.2, "KlarosScript Interface"</u> implementieren muss. Dieses Interface definiert die zentrale Methode execute und nimmt ein Objekt des Typs <u>Abschnitt E.2, "KlarosScript Interface"</u> als Input-Parameter.

Dieses Kapitel richtet sich hauptsächlich an **Berichtsdesigner** und zeigt, wie die benutzerdefinierten Berichtsvorlagen erstellt werden. Benutzer sehen in <u>Abschnitt 9.2, "Berichte"</u> wie daraus Berichte erstellt werden können.

11.1. Das Kontext-Objekt

Im Berichts-Skript werden die Daten für die Darstellung in der Layout-Vorlage bereitgestellt. Es bezieht die Daten entweder direkt aus dem Objektmodell <u>de.verit.klaros.core.model</u> oder aus der Datenbank über die Abfragesprache HQL.

Die resultierenden Daten für die Layout-Vorlage müssen dann im Objekt <u>Abschnitt E.2, "KlarosS-</u> <u>cript Interface"</u> gespeichert werden, das an die Methode execute übergeben wurde.

Sind Parameter definiert, sind sie über das Kontextobjekt sowohl für das Berichts-Skript als auch für die Layout-Vorlage verfügbar.



Vordefinierte Objekte im Kontext

Das Objekt <u>Abschnitt E.2, "KlarosScript Interface"</u> enthält bereits vordefinierte Objekte. Eine vollständige Liste ist unter <u>Abschnitt E.1, "Der Klaros Report-Kontext"</u> zu finden.

11.2. Eine neue Berichtsvorlage erstellen

Wählen Sie unter *B* Konfigurieren den Menüpunkt Berichtsvorlagen aus.

≡ `	🞐 KLAR (OS TEST	MANAGEMENT		DE Finanz-Tracker 📑		વ	× 0~ ±~
C I	Definieren	Berichtsvorla	gen					
📇 Planen		Neu					Speichern	Verwerfen
				7 Einträge - Seite 1 von 1 📕 ┥ 🚺 🕨 🕅 🔟	0 🗸		AII	e anzeigen 🕹 📃
n .	Ausführen	🔲 ID 🗢 🛛 🖓	Name 🗢	Beschreibung 🗢	Status 🗢 🛛 Format 🗢	Revision	Geändert von 🗢 Geändert	Aktion
* '	Adoramen	RT00007	Issue Overview Report	Lists the issues in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖾 🔍 💭 🛍
-		RT00006	Iteration Status Report	Summary of an iteration status in a project	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖾 🔍 💭 🛍
(; <i>i</i>	Auswerten	RT00005 🖰	Job Excel Overview Report	Lists the jobs in a project	Genehmigt Excel	1.1	System Account vor 2 Jahr	en 🖈 🔍 💭 🛍
		RT00004	Job Progress Report	Lists the jobs in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖾 🔍 💭 🛍
ا عر	Konfigurieren	RT00003	System under Test Status Report	Lists the test case results in a SuT sorted by selectable parameters	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖪 🔍 💭 🛍
		RT00002	Test Run Overview Report	Lists the test runs in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖪 🔍 💭 🛍
		RT00001 🛆	Job Status Report	Lists the jobs in a project grouped by selectable parameters	Genehmigt PDF	1.1	System Account vor 2 Jahr	en 🖪 🔍 💭 🛍
		ID	Name	Beschreibung	Status Format	Revision	Geändert von Geänder	t Aktion
		Neu					Speichern	Verwerfen

Abbildung 11.2. Die Seite "Berichtsvorlagen"

Ein Klick auf den Button Neu erzeugt eine neue Berichtsvorlage. Es öffnet sich die Seite Berichtsvorlage - Detailansicht:

Name	Der Name des Berichts.
Beschreibung	Eine Beschreibung des Berichts.
Ausgabeformat	Das Ausgabeformat: PDF oder Excel.
Status	Der Status des Berichts: Entwurf oder genehmigt

Klaros-Testmanagement wird mit mehreren vordefinierten Berichtsvorlagen ausgeliefert. Diese sind schreibgeschützt, können aber mit dem Icon Duplikat werden. Das Duplikat kann dann nach Belieben geändert werden.

≡	👱 K L A R (DS TEST N				DE Finanz-Tracker 📋 📑		ୟ × ଡେ∽ ≛∽
C Definieren RT00005 - Job Excel Overview Report								60
*	Planen	Eigenschaften	Parameter (2) Skr	ript Layout-Vorlage Rev	isionen Änderungen	Vorschau	Speichern Verwer	fen Zurück
\$	Ausführen	Name	Label		Beschreibung	Тур	Voreinstellung	Verpflichtend Aktion
	Augustan	sortJobBy	Sort Job List	Select the order to show the	obs	Mehrfachaufzählung	Job ID	1 🗇
	Auswerten	snowFinishedJob	s Show Finished	It selected, the report include	s closed, resolved and rejected jobs	Boolescher wert	v	÷ Ш
×	Konfigurieren	Neuer Pa	rameter					
						Vorschau	Speichern Verwer	fen Zurück

Abbildung 11.3. Die Seite "Berichtsvorlagen - Details"

• Die Ansicht **Parameter** zeigt die für diese Vorlage definierten Parameter an. Bei neu erzeugten Berichten ist diese Liste leer.

- Die Ansicht *Script* verwaltet das Berichts-Skript. Neu erzeugte Berichte verfügen bereits über eine vordefinierte Klasse mit den notwendigen Methoden.
- Die Layout-Vorlage wird in der Ansicht *Layout-Vorlage* verwaltet. Wie beim Berichts-Skript ist für neue Berichte bereits ein Grundlayout vorhanden.



Anmerkung

Für größere Berichtsvorlagen kann es hilfreich sein, eine IDE (z.B. Eclipse) für das Skript und die Layout-Vorlage zu verwenden. Die erstellten Dateien können dann in Klaros-Testmanagement hochgeladen werden. Um Fehler zu vermeiden, fügen Sie die Klaros-Modellbibliotheken zum Build-Pfad Ihres Eclipse-Projekts hinzu. Das Tutorial-Dokument verfügt über ein Kapitel zur Erstellung von Berichtsvorlagen, in dem dies detailliert erläutert wird.

Anstatt das Berichts-Skript und die Vorlage direkt in der Anwendung zu editieren, können diese auch aus einer Datei importiert werden. Geben Sie die zu verwendende Datei an, indem Sie auf den Button Browse klicken und die zu importierende Datei auswählen. Ein Klick auf *OK* im Datei-Dialog und danach auf Upload importiert die ausgewählte Datei in das Skript- oder Vorlagenfeld.

Diese Seite bietet drei ausführbare Aktionen:

Vorschau erzeugt eine Vorschau des Berichtes.

Speichern speichert die Berichtsvorlage.

Verwerfen verwirft alle Änderungen an der Berichtsvorlage.

Um Berichte flexibler zu gestalten, können Parameter an das Berichts-Skript zu übergeben werden. Diese können z.B. dazu verwendet werden, eine Zeitspanne festzulegen, für die die Daten ermittelt werden.

Mit einem Klick auf das Icon + wird ein neuer Parameter angelegt <u>Abbildung 11.4, "Die Ansicht</u> "<u>Parameter""</u>.

≡ 👱 KLARO	DS TEST MAN			DE Finanz-Tracker 👕 📑		Q X	0 ~ ≗ ~
📝 Definieren	RT00002 - Test Run Ov	verview Report					GĐ
📇 Planen	Eigenschaften Parame	eter (5) Skript Layo	ut-Vorlage Revisionen Änderungen	Vorschau	Speichern	Verwerfen	Zurück
🏟 Ausführen	Name	Label	Beschreibung	Тур	Voreinstell	lung Verpflich	tend Aktion
	startDate	Start Date	The start date of the test runs to consider	Datum			1 🔟
🕒 Auswerten	endDate	End Date	The end date of the test runs to consider	Datum			1 🗇
	sortByTestRun	Test Run Sort Criteria	Choose an entry, how the test runs should be listed.	Mehrfachaufzählung	Test Result		1 🗓
🔑 Konfigurieren	showDetailedTestRuns	Show Test Run Details	Select if detailed test run results should be included	Boolescher Wert	\checkmark		1 🗓
- Konnguneren	showPassedResults	Show Passed Results	Also include passed results	Boolescher Wert	\checkmark		1 🗇
	Neuer Parameter			Vorschau	Speichern	Verwerfen	Zurück

Abbildung 11.4. Die Ansicht "Parameter"

Der Typ der Parameter kann durch Auswahl des entsprechenden Wertes aus der Typenliste angegeben werden. Unterstützte Typen sind Text, Number, Datum, Boolean, Auswahl und Mehrfachauswahl. Es können Vorgabewerte angegeben und ein Pflicht-Flag gesetzt werden. Mit Klick auf das Icon 前 wird der Parameter aus der Liste entfernt.

Auf die übergebenen Parameter kann das Berichts-Skript zugreifen, indem es entweder die Methode getParameterValue(String name) oder die Methode getParameter(String name) aufruft. Diese Methoden geben null zurück, wenn kein Parameter mit dem angegebenen Namen gefunden werden konnte.

11.2.1. Unterstützte Parametertypen

Die folgenden Parametertypen werden unterstützt:

Text Zahl Datum Auswahl Mehrfachauswahl

11.2.2. Umgang mit Parametern

Wie bereits erwähnt, können die übergebenen Parameter über die Methoden getParameterValue(-String name) und getParameter(String name) aus dem Kontext abgerufen werden.

Der folgende Code-Ausschnitt zeigt, wie auf Parameter zugegriffen werden kann und wie sie im Berichts-Skript verwendet werden können:

```
StringBuffer query = new StringBuffer();
query.append("select tcr from KlarosTestCaseResult tcr where tcr.executionTime < ");
query.append(context.getParameterValue("executionTime"));
List<?> tcr = context.executeQuery(query.toString());
```

Diese Abfrage ruft alle Objekte vom Typ KlarosTestCaseResult ab, die einen Ausführungszeitpunkt haben der vor dem mit dem Parameter executionTime übergebenen Wert liegen.

Alternativ kann, wie im folgenden Code-Ausschnitt dargestellt, in der Abfrage auch direkt auf die Parameter zugegriffen werden:

```
query.append("select tcr from KlarosTestCaseResult tcr where tcr.executionTime <: executionTime");
List<?> tcr = context.executeParameterizedQuery(query.toString());
```

Von der Layout-Vorlage aus kann auf Parameter wie folgt zugegriffen werden:

<p:text value="Test results for test runs with execution time < #{executionTime} ms" />



Anmerkung

Stellen Sie sicher, dass Zeichen wie &, < oder > oder > escaped werden, wenn sie in XML-Attributen in der Layout-Vorlage verwendet werden. In unserem Beispiel ist < das Escape-Zeichen für <.

11.3. Einen Bericht erzeugen

Um einen Bericht zu erzeugen, klicken Sie auf das Icon (); *Auswerten* und wählen Sie *Berichte* aus dem Menü auf der linken Seite. Es wird eine Liste der vorhandenen Berichtsvorlagen angezeigt.

≡	学 K L A R (DS TEST	MANAGEMENT	-	DE Finanz-Track	er 🖬 📕	8	Q ×	€ 0 × ≛×
ľ	Definieren	Berichtsvorla	gen						
-2:	Planen	Neu		_				Speichern	Verwerfen
				7 Einträge - Seite 1 von 1 🛛 🖌 🚽 🧴 🕨	10 🗸			Alle ar	nzeigen 🖧 🗮
~	Ausführen	D 🗘 🚺							Aktion
~	Austumen	RT00007 6	Sissue Overview Report	Lists the issues in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account	vor etwa einem Jahr	▯൨▯▯
¢	Auswerten	RT00006	Iteration Status Report	Summary of an iteration status in a project	Genehmigt PDF	1.1	System Account	vor etwa einem Jahr	
		RT00005 🗧	Job Excel Overview Report	Lists the jobs in a project	Genehmigt Excel	1.1	System Account	vor etwa einem Jahr	x Q 🖸 🛍
ء	V	RT00004 6	Job Progress Report	Lists the jobs in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account	vor 12 Monaten	[] Q [] ∭
	Konliguneren	RT00003 6	System under Test Status Report	Lists the test case results in a SuT sorted by selectable parameters	Genehmigt PDF	1.1	System Account	vor 12 Monaten	[]♀[□ @
		RT00002 6	Test Run Overview Report	Lists the test runs in a project sorted by selectable parameters	Genehmigt PDF	1.1	System Account	vor 12 Monaten	฿ୣୣୣୣ (, () ()
		RT00001 £	j Job Status Report	Lists the jobs in a project grouped by selectable parameters	Genehmigt PDF	1.1	System Account	vor 12 Monaten	[] Q [] ∅
		ID	Name	Beschreibung	Status Format	Revision	Geändert von	Geändert	Aktion
		Neu						Speichern	Verwerfen

Abbildung 11.5. Einen Bericht erzeugen

Mit Kick auf das Icon 🔁 oder 🖈 wird der Bericht als PDF- oder Excel-Datei gerendert. Wurden für die Berichtsvorlage Parametern definiert, so öffnet sich ein Dialog, in dem die Parameter eingegeben werden können.

≡	🞐 KLARO	d s test managem	ENT	DE Finanz-Tracker	∎
	Definieren	Berichtsvorlagen			
		Neu			Speichern Verwerfen
***	Planen	0 û	7 Einträge - Seite 1 von 1 🔣 ┥ 1 🕨 🕅	10 🗸	γ Alle anzeigen 🕹 \equiv
-	A	Name 🗘 🛛 Name	Beschreibung 🗢	Status 🗘 Format 🖨 Revision	Geändert von 🗢 Geändert 🗢 Aktion
	Austunren	🗌 👌 Issue Overview Report	Lists the issues in a project sorted by selectable parameters	Genehmigt PDF 1.4	System Account 28.04.21, 04:34 📙 📿 🗍 前
		🗌 🛆 Iteration Status Report	Summary of an iteration status in a project	Genehmigt PDF 1.3	System Account 17.07.21, 04:24 📙 📿 🗍
	Auswerten	🛛 🛆 Job Excel Overview Report	Lists the jobs in a project	Genehmigt Excel 1.2	System Account 14.04.21, 11:17 🗴 🔍 🛄 🔟
		🗌 🛆 Job Status Report	Lists the jobs in a project grouped by selectable parameters	Genehmigt PDF 1.2	System Account 14.04.21, 11:17 📙 📿 🛄 前
لكر ا	Konfigurieren	Job Status Report	Lists the jobs in a project sorted by selectable parameters	Genehmigt PDF 1.4	System Account 05.07.21, 04:20 🔀 📿 🗍 🗍
ſ [°]		📃 👌 System under Test Status Repor	: Lists the test case results in a SuT sorted by selectable parameters	Genehmigt PDF 1.4	System Account 07.07.21, 04:22 🔀 📿 🛄 🛍
		Test Run Overview Report	Lists the test runs in a project sorted by selectable parameters	Genehmigt PDF 1.3	System Account 15.06.21, 20:59 🗋 📿 🗍 🗊
		Bitte geben Sie die benötigten Param	eter ein		Speichern Verwerfen
		Start Date	曲		
		Fad Date	*		
		End Date			
		Test Run Sort Criteria Test Resu	it X V		
		Show Test Run Details 🖌			
		Show Passed Results 🖌			
				ОК	Abbrechen

Abbildung 11.6. Berichtsparameter eingeben

11.4. Beispielbericht

In diesem Abschnitt finden Sie einen Beispielbericht, der Testfallergebnisse abruft und sie abhängig von ihrem Status anzeigt.

11.4.1. Erzeugen des Skripts

Der folgende Code-Ausschnitt zeigt den Aufbau eines Java-Skripts mit den erforderlichen Import-Statements. Der Code zum Abrufen der Daten wird in der Methode execute implementiert. Eine detailliertere Beschreibung der Klaros-Testmanagement API finden Sie in <u>de.verit.klaros.core.model</u>.

```
import de.verit.klaros.scripting.*;
import de.verit.klaros.core.model.*;
import java.util.*;
public class TestScript implements KlarosScript {
    public void execute(KlarosContext context) {...
    }
}
```

Der nächste Schritt besteht darin, die erforderlichen Daten abzurufen. Der folgende Code-Ausschnitt zeigt, wie eine Abfrage zum Abrufen von Testfallergebnissen erstellt werden kann.

```
StringBuffer query = new StringBuffer();
query.append("select tcr from KlarosTestCaseResult tcr");
List<?> tcr = context.executeQuery(query.toString());
```

Die Testfallergebnisse werden in einem List-Objekt zurückgegeben. Diese Liste muss im Kontext zwischengespeichert werden, damit die Layout-Vorlage darauf zugreifen kann:

context.add("results", tcr);



Anmerkung

Weitere Informationen zum Erstellen von Abfragen finden Sie unter HQL-Dokumentation.

Das List-Objekt wird im Kontext unter dem Namen results gespeichert und kann von der Layout-Vorlage über diesen Namen abgerufen werden. Falls zusätzliche Daten benötigt werden, führen Sie die entsprechenden Abfragen aus und speichern Sie die verarbeiteten Daten im Kontext unter jeweils einem anderen Namen.

11.4.2. Eine PDF-Layout-Vorlage erzeugen

Die Code-Ausschnitte in diesem Abschnitt zeigen, wie man eine Layout-Vorlage erstellt. Die offizielle Dokumentation zur verwendeten Layout-Sprache finden Sie im SeamPDF Manual. Weitere Informationen zum Objekt-Modell finden Sie unter <u>de.verit.klaros.core.model</u>.

Der folgende Code-Ausschnitt zeigt die Grundstruktur einer PDF-Layout-Vorlage. Im Element <p:document>-Elements ist der Inhalt des Berichts enthalten und in Kapiteln und Abschnitten gruppiert.

```
<p:document xmlns:ui="http://java.sun.com/jsf/facelets" xmlns:f="http://java.sun.com/jsf/core"
xmlns:p="http://jboss.org/schema/seam/pdf" title="Klaros-Testmanagement Test Suite Report"
marginMirroring="true" author="#{user.name}" creator="#{user.name}" pageSize="A4">
...
```

```
</p:document>
```

Hier wird der immer im Kontext vorhandene Parameter <u>de.verit.klaros.core.model.KlarosUser</u> mittels #{user.name} verwendet, um den Author des PDF-Dokuments festzulegen.

Der nächste Code-Ausschnitt zeigt, wie die Kopf- und Fußzeilen des Berichts definiert werden können. Hier wird das aktuelle Datum und der <u>de.verit.klaros.core.model.KlarosUser</u>-Parameter im Kontext verwendet.

```
<f:facet name="header">
<p:font size="8">
<p:font size="8">
<p:header borderWidthBottom="0.1" borderColorBottom="black" borderWidthTop="0" alignment="center">
<p:header borderWidthBottom="0.1" borderColorBottom="black" borderWidthTop="0" alignment="center">
<p:text value="Example report - generated on #{date} by #{user.name}"/>
</p:header>
<p:footer borderWidthTop="0.1" borderColorTop="black" borderWidthBottom="0" alignment="center">
```

Als nächstes wird die Titelseite für den Bericht definiert, die eine kurze Zusammenfassung des Berichts zeigt. Um dieses Beispiel kurzzuhalten, wird nur ein Fragment mit dem Namen und der E-Mail-Adresse des Benutzers angezeigt. Für das vollständige Skript siehe <u>Abschnitt E.3, "Beispiel Layout-Vorlage"</u>.

```
<p:font style="bold" size="16">
  <p:font style="bold" size="16">
  <p:paragraph alignment="center" spacingAfter="5">
  <p:font</pre>
</p:font>
```

11.5. Ein Diagramm erstellen

Berichte können durch ein Diagramm mit einer grafischen Darstellung der abgerufenen Daten erweitert werden. Die Daten für diese Diagramme werden ebenfalls durch das Berichts-Skript aufbereitet und im Kontext gespeichert. Die Layout-Vorlage übergibt die Daten an die Diagrammkomponente von SeamPDF. In diesem Abschnitt wird gezeigt, wie ein Tortendiagramm wie in <u>Abbildung 11.7, "Ein Beispiel für ein Tortendiagramm"</u> erstellt wird.



Abbildung 11.7. Ein Beispiel für ein Tortendiagramm

11.5.1. Script für ein Tortendiagramm

Das Berichts-Skript wird nicht nur zum Abrufen der Daten verwendet, sondern auch zum Aufbereiten der Daten für die Layout-Vorlage. Der folgende Code-Ausschnitt zeigt, wie die Daten für ein Tortendiagramm aufbereitet werden. Für jeden möglichen Ergebnistyp wird ein leeres List-Objekt erstellt. Dann werden die zuvor abgerufenen <u>de.verit.klaros.core.model.KlarosTestCase-Result</u>-Objekte je nach ihrem Ergebnistyp zu einer der Listen hinzugefügt. Auf diese Listen kann dann später aus dem Kontext heraus über den entsprechenden Schlüssel zugegriffen werden.

```
List<KlarosTestCaseResult> error = new ArrayList<KlarosTestCaseResult>();
List<KlarosTestCaseResult> failure = new ArrayList<KlarosTestCaseResult>();
List<KlarosTestCaseResult> success = new ArrayList<KlarosTestCaseResult>();
// Iterate over the results and retrieve the status
Iterator<KlarosTestCaseResult> iter = (Iterator<KlarosTestCaseResult>) tcr.iterator();
while (iter.hasNext()) {
    KlarosTestCaseResult result = iter.next();
    if (result.isError()) error.add(result);
    else if (result.isFailure()) failure.add(result);
    else if (result.isPassed()) success.add(result);
  }
  context.add("error", error);
  context.add("failure", failure);
  context.add("success", success);
```

11.5.2. Pie Chart Report Template

Der folgende Code-Ausschnitt zeigt das Diagramm unter Verwendung der Daten, die durch das Berichts-Skript im vorigen Abschnitt vorbereitet wurden.

```
<p:paragraph horizontalAlignment="center">
<p:piechart title="Test Results" direction="anticlockwise" circular="true"
startAngle="30" labelGap="0.1" labelLinkPaint="#000000" plotBackgroundPaint="#ffffff"
labelBackgroundPaint="#ffffff" is3D="true">
<p:series key="results">
<p:data key="Error [#{error.size}]" value="#{error.size}" sectionPaint="#FF0A0A" />
```

```
<p:data key="Success [#{success.size}]" value="#{success.size}" sectionPaint="#33CC00"/>
        <p:data key="Failure [#{failure.size}]" value="#{failure.size}" sectionPaint="#FFCC00"/>
        </p:series>
        </p:piechart>
    </p:paragraph>
```

Das Element piechart definiert die Darstellung und den Inhalt des Diagramms. Für ein Tortendiagramm ist das Element series erforderlich. Die Daten werden aus den Listen abgerufen, die zuvor vom Skript im Kontext gespeichert wurden.

<p:data key="Error [#{error.size}]" value="#{error.size}" sectionPaint="#FF0A0A" />

Dieser Code-Ausschnitt ruft die Liste mit den Ergebnissen aus dem Kontext ab und ruft seine size() Methode auf, um die Anzahl der Testfälle zu bestimmen. Das Tortendiagramm wird dann für die drei Datenabschnitte gerendert, wie in <u>Abbildung 11.7, "Ein Beispiel für ein Tortendiagramm"</u> zu sehen ist.



Anmerkung

Detaillierte Informationen zu den verschiedenen Diagrammtypen finden Sie in der SeamPDF-Dokumentation.

11.5.3. Bilder einbinden

Dieser Abschnitt zeigt, wie ein Bild in einen Bericht aufgenommen werden kann. Um für die Layout-Vorlage zugänglich zu sein, muss die Bilddatei an einem von zwei möglichen Orten gespeichert werden:

- In einer .jar-Datei im Ordner .klaros/resources
- · Auf einem Webserver, welches über eine URL abgerufen werden kann

Wenn das Bild in einer .jar-Datei gespeichert wird, kann mit dem folgenden Code-Ausschnitt auf das Bild zugegriffen werden.

<p:image value="images/image.png"/>

Das Attribut value definiert den Bilddateinamen und den Ordner, in dem sich das Bild innerhalb der jar-Datei befindet.

Wenn ein Bild über eine URL bereitgestellt wird, kann darauf mit dem folgenden Code-Ausschnitt zugegriffen werden.

```
<p:html>
<img src="http://www.verit.de/images/logo-klaros-160.png" />
</p:html>
```

Das Attribut src definiert die URL der Bild-Datei.

11.5.4. Eine Layout-Vorlage für Excel erzeugen

Die in diesem Abschnitt vorgestellten Code-Ausschnitte zeigen, wie man eine Layout-Vorlage für Klaros-Testmanagement erstellt. Detaillierte Informationen finden Sie im SeamExcel Handbuch. Weitere Informationen über das Klaros-Testmanagement Objektmodell sind hier zu finden <u>de.verit.klaros.core.model</u>.

Das folgende Code-Segment zeigt den Aufbau einer Excel-Layout-Vorlage. as Element <e:workbook> enthält alle weiteren Berichtsdetails.

Beachten Sie die Verwendung der <u>de.verit.klaros.core.model.KlarosUser</u> und der <u>de.verit.kla-</u> <u>ros.core.model.KlarosTestCase</u> Parameter aus dem Kontext: #{user.name} und #{testCases}.

Ein Arbeitsblatt verwendet eine Liste von Objekten, um die Zeilen und Spalten der Excel-Tabelle zu erzeugen. Diese Liste muss an das Attribut value des worksheet-Elements übergeben werden.

Im Attribut var kann der Name des aktuellen Objekts definiert werden. Zum Beispiel: Um den Testfallnamen des aktuellen Objekts innerhalb des Arbeitsblattes auszugeben, muss die Anweisung #{testCase.name} verwendet werden.

Es ist ebenfalls möglich, mehrere worksheet-Elemente in einer Vorlage zu verwenden. In diesem Beispiel verwenden wir aber nur ein einziges Arbeitsblatt mit dem Benutzernamen als Arbeitsblattname.

Kapitel 12. Import/Export

Klaros-Testmanagement bietet zahlreiche Schnittstellen zum Importieren von Daten aus anderen Anwendungen sowie zum Exportieren eigener Daten in verschiedene Formate an.

Für alle Importvorgänge ist eine Benutzerauthentifizierung erforderlich. Die benötigte Benutzerrolle hängt von der Art der zu importierenden Daten ab. Für den Import von Testergebnissen ist die Rolle *Tester* ausreichend, für Anforderungen und Testfälle ist die Rolle *Manager* erforderlich.



LDAP-Authentifizierung

Ist die Einstellung *Login Voreinstellung* in der LDAP-Konfiguration aktiviert, so erfolgt die Authentifizierung ausschließlich über das LDAP-Verzeichnis. Lokale Authentifizierungsinformationen werden in diesem Fall immer ignoriert.

12.1. Testfälle aus Excel-Tabellen importieren

Um Testfälle korrekt aus Excel-Tabellen zu importieren, müssen die Daten in einem vorgegebenen Format vorliegen. Dieses Format ist in <u>Abschnitt 12.1.1, "Import-Format"</u> detailliert beschrieben.

In der Klaros-Testmanagement Enterprise Edition können neben den Standardfeldern auch Daten für benutzerdefinierte Felder importiert werden. Dafür müssen im entsprechenden Projekt einige Vorbedingungen erfüllt sein. Weitere Informationen hierzu finden Sie unter <u>Abschnitt 12.1.2, "Vorbedingungen"</u>.

Die Importschnittstelle besteht aus einem REST-Interface, dass sowohl per Kommandozeile als auch über andere Anwendungen erreicht werden kann. Detailinformationen dazu finden sich in <u>Abschnitt 12.1.3, "Ausführen"</u>.

12.1.1. Import-Format

Um Testfälle korrekt zu importieren, muss die Excel-Tabelle einem bestimmten Format entsprechen:

- 1. Jeder Testfall muss in einer separaten Tabelle definiert sein.
- 2. Jede Tabelle ist in drei Abschnitte unterteilt: allgemeine Eigenschaften, Testfallschritte und benutzerdefinierte Eigenschaften.
- 3. Dabei sind die Felder A:1 bis A:22 für die Standard-Attribute eines Testfalls reserviert. Die entsprechenden Werte werden in die Felder B:1 bis B:22 eingetragen. Es ist nicht zwingend notwendig, alle Felder zu befüllen.
- 4. Die Spalte G ist für die Namen von beliebig vielen benutzerdefinierten Eigenschaften vorgesehen. In Spalte H werden die entsprechenden Werte dazu eingegeben.
- 5. Das Feld Step in der Zelle A:25 **muss** bestehen bleiben. Ändern oder Löschen des Feldes A:25 führt dazu, dass der Import dieser Tabelle abgewiesen wird. Ab hier beginnt die Definition der Testschritte:

- Spalte A für die Testschrittnummer,
- Spalte B für die Aktion,
- Spalte C für die Vorbedingung,
- Spalte D für die Nachbedingung und
- Spalte E für das erwartete Ergebnis.

	A	В	С	D	E	F	G	Н	1
1	ID	TC00051	Wird automatisci	h erzeugt			Zyklus	Major	
2	Name	Login-Test		, i i i i i i i i i i i i i i i i i i i			Kategorie	Regressionstests	
3	Revision	1.0	Wird automatisci	h erzeugt					
4						Don	utzordofin	iarta Eiganaahaf	ton
5	Beschreibung	Prüft die grundlegende Funktionalität				Den	utzeruenn	lerte Eigenschaf	len
6	Vorbedingung	Server ist installiert							
7	Nachbedingung	Benutzer ist authentifiziert							
8	Erwartetes Ergebnis	Hauptseite öffnet sich							
9									
10	Anmerkung	Erforderlich für den Basisbetrieb							
11	Testart	FUNCTIONAL							
12	Entwurfsverfahren	BLACK_BOX		haften					
13	Ergebnisart	POSITIVE	emeine Eigenso	charten					
14	Ausführung	MANUAL							
15	Priorität	HIGH							
16	Status	APPROVED							
17	Team	QA							
18	Teststufe	SYSTEM							
19	Dokument	Requirements 1.5 (12.07.2020)							
20	Abhängigkeit	•							
21	Evaluierungsart	MANUAL							
22	Verfolgbarkeit	UC-002							
23	Geschätzte Dauer (ms)	3000							
24									
25	Step	Aktion	Vorbedingung	Nachbedingung	Erwartetes Ergebnis				
26		Login-Seite offnen			Login-Seite wird angezeigt				
21	2	Benutzername eingeben	Testschritte	e					
28	3	Passwort eingeben			Passwort darf nicht sichtbar sein				
29	4	OK klicken			Benutzer ist eingeloggt				
30	Feld A:25								
31	darf nicht entfernt								
32	werden!								
33	norden.								
-34	1		1		I contract of the second se				

Abbildung 12.1. Beispiel für einen Testfall in Excel

12.1.1.1. Allgemeine Eigenschaften

In der unten dargestellten Tabelle sind die Zellkoordinaten für die Werte der einzelnen allgemeinen Testfallattribute dargestellt. Die ID des Testfalls wird beim Import automatisch erzeugt. Keines dieser Felder muss zwingend befüllt werden. Einige Felder können nur vordefinierte Werte annehmen. Diese sind in Klammern aufgeführt.

Koordinate	Wert
B2	Name
B3	Revision (Verwendung nur beim Export)
B5	Beschreibung
B6	Vorbedingung
B7	Nachbedingung
B8	Erwartetes Ergebnis
B10	Bemerkung
B11	Testart (FUNCTIONAL, NON_FUNCTIONAL, STRUCTURAL, REGRESSION, RE_TEST)

Koordinate	Wert
B12	Entwurfsverfahren (BLACK_BOX, WHITE_BOX)
B13	Ergebnisart (POSITIVE, NEGATIVE)
B14	Ausführung (MANUAL, AUTO)
B15	Priorität (LOW, MEDIUM, HIGH)
B16	Status (NEW, APPROVED, LOCKED, INVISIBLE)
B17	Team
B18	Level (COMPONENT, INTEGRATION, SYSTEM, ACCEPTANCE)
B19	Dokument
B20	Abhängigkeit
B21	Auswertung
B22	Verfolgbarkeit
B23	Geschätzte Dauer (ms)

Tabelle 12.1. Koordinaten der allgemeinen Eigenschaften von Testfällen

12.1.1.2. Testschritte

Testfälle können eine beliebige Anzahl von Testschritten enthalten. Diese werden ab Zelle A:26 pro Zeile einzeln in aufsteigender Reihenfolge aufgelistet. Ist in einer Zeile keine Schrittnummer enthalten, wird die Verarbeitung der Excel-Tabelle beendet.

Koordinaten	Wert
A26 (ff)	Schritt-Nummer
B26 (ff)	Aktion
C26 (ff)	Vorbedingung
D26 (ff)	Nachbedingung
E26 (ff)	Erwartetes Ergebnis

Tabelle 12.2. Koordinaten der Testschritte

12.1.1.3. Benutzerdefinierte Eigenschaften

```
wittion Nur in der Klaros-Testmanagement Enterprise Edition verfügbar
```

Testfälle können benutzerdefinierte Eigenschaften enthalten. Diese werden als Name/Wert-Paare ab Zelle G:1 und H:1 abwärts aufgelistet.

Der Name der in Spalte G aufgelisteten Eigenschaft muss exakt mit dem Namen der benutzerdefinierten Eigenschaft im entsprechenden Projekt übereinstimmen.

Handelt es sich bei der benutzerdefinierten Eigenschaft um eine Aufzählung, so muss das Wertfeld exakt mit einem der dort definierten Aufzählungswerte übereinstimmen. Wird eine leere Namenszelle gefunden, stoppt die Verarbeitung.

Koordinaten	Wert
G1 (ff)	Eigenschafts-Name
H1 (ff)	Eigenschafts-Wert

Tabelle 12.3. Koordinaten der benutzerdefinierten Eigenschaften von Testfällen

12.1.2. Vorbedingungen

Um Testfälle aus Excel-Tabellen importieren zu können, müssen folgende Vorbedingungen erfüllt sein:

- 1. Die zu importierende Datei muss in den Formaten XLS oder XLSX vorliegen.
- 2. Das Projekt, in welches die Testfälle importiert werden sollen, muss bereits erstellt sein.
- 3. Benutzerdefinierte Eigenschaften (können nur in der Klaros-Testmanagement Enterprise Edition erstellt werden!) müssen bereits im entsprechenden Projekt definiert sein.

12.1.3. Ausführen

Testfälle aus Excel-Tabellen lassen sich über eine REST-Schnittstelle mit Hilfe eines Kommandozeilen- basierten Werkzeuges oder anderen beliebigen Anwendungen hochladen.

Das folgende Beispiel zeigt, wie eine Excel-Tabelle mit Testfällen in das Projekt mit der ID P00001 unter Verwendung der Curl-Kommandozeile importiert werden kann. Curl sollte in jeder Linux-Distribution enthalten sein. Für die Microsoft Windows-Betriebssystemfamilien ist es als Teil der Cygwin-Distribution http://www.cygwin.com/ oder als Kommandozeilenwerkzeug von http://curl. haxx.se/download.html erhältlich.



Inkompatibler Windows Powershell-Curl-Alias

Windows Powershell definiert ebenfalls einen Alias namens **curl**, der auch für diesen Zweck verwendet werden kann. Dieser benötigt allerdings andere Argumente. Wollen Sie das Linux-kompatible Curl-Programm unter Powershell verwenden, geben Sie **curl.exe** anstelle von **curl** ein.

```
curl -v -T TestCases.xls "<klaros-app-url>/seam/resource/rest/import/testcase/xls?config=P00001\
&username=user&password=secret"
```

Beispiel 12.1. Testfallimport aus Excel-Tabellen über die Kommandozeile



ID-Format

Alle Objekte, die während des Imports referenziert werden (wie Projekte oder Testfälle), enthalten fünf Ziffern in ihrer ID.

So ist P00001 eine gültige Projekt-ID, während P0001 und P000001 ungültig sind.

12.2. Testfälle aus XML-Dateien importieren

Beim Importieren von Testfällen aus XML-Dateien erzeugt jeder Importvorgang neue Instanzen der gelieferten Testfälle.

Das Format der XML-Datei ist in <u>Anhang C, Dateiformat für Testfall-Import</u> beschrieben. Das XML-Schema ist unter der folgenden URL erhältlich: https://www.klaros-testmanagement.com/files/ schema/klaros-testcases-1.0.xsd.

12.2.1. Vorbedingungen

Um Testfälle aus XML-Dateien importieren zu können, müssen folgende Vorbedingungen erfüllt sein:

- 1. Die zu importierende Datei muss im XML-Format vorliegen.
- 2. Das Projekt, in welches die Testfälle importiert werden sollen, muss bereits erstellt sein.
- 3. Wenn benutzerdefinierte Eigenschaften importiert werden sollen, müssen sie vor dem Importvorgang im Projekt angelegt werden.

12.2.2. Ausführen

Für den Import von XML-basierten Testfällen steht eine REST-Schnittstelle zur Verfügung, über die die Daten aus der Kommandozeile oder anderen Anwendungen hochgeladen werden können.

Das folgende Beispiel zeigt, wie eine XML-Datei mit einem Testfall in das Projekt mit der ID P00001 unter Verwendung der Curl-Kommandozeile importiert werden kann. Curl sollte in jeder Linux-Distribution enthalten sein. Für die Microsoft Windows-Betriebssystemfamilien ist es als Teil der Cygwin-Distribution http://www.cygwin.com/ oder als Kommandozeilenwerkzeug von http://curl. haxx.se/download.html erhältlich.

curl -v -T TestCases.xml "<klaros-app-url>/seam/resource/rest/import/testcase/xml?config=P00001\
&username=user&password=secret"

Beispiel 12.2. XML-Testfall-Import über die Kommandozeile

12.3. Anforderungen aus Excel-Tabellen importieren

Um Anforderungen korrekt aus Excel-Tabellen zu importieren, müssen die Daten in einem vorgegebenen Format vorliegen. Dieses Format ist im folgenden Abschnitt <u>Abschnitt 12.1.1, "Import-</u> <u>Format"</u> detailliert beschrieben.

In der Klaros-Testmanagement Enterprise Edition können neben den Standardfeldern auch Daten für benutzerdefinierte Felder importiert werden. Dazu ist es notwendig, das Projekt entsprechend vorzubereiten. Weitere Informationen dazu finden Sie unter <u>Abschnitt 12.1.2, "Vorbedingungen"</u>.

Die Importschnittstelle besteht aus einem REST-Interface, dass sowohl per Kommandozeile als auch über andere Anwendungen erreicht werden kann. Dies wird in Abschnitt <u>Abschnitt 12.1.3,</u> <u>"Ausführen"</u> detailliert beschrieben.

12.3.1. Import-Format

Um Anforderungen korrekt zu importieren, muss die Excel-Tabelle einem bestimmten Format entsprechen:

- 1. Jede Anforderung muss in einer separaten Tabelle definiert sein.
- 2. Jede Tabelle ist in zwei Abschnitte unterteilt: allgemeine Eigenschaften und benutzerdefinierte Eigenschaften.
- 3. Dabei sind die Felder A:1 bis A:9 für die Standard-Attribute einer Anforderung reserviert. Die entsprechenden Werte werden in die Felder B:1 bis B:9 eingetragen. Es ist nicht zwingend notwendig, alle Felder zu befüllen.
- 4. Die Spalte G ist für die Namen für beliebig viele benutzerdefinierte Eigenschaften vorgesehen. In Spalte H werden die entsprechenden Werte dazu eingegeben.

	A	В	С	D	E	F	G	н	- I
1	ID	R00001	Wird autom	Wird automatisch erzeugt			Abteilung	Intern	
2	Name	Authorisierung erforderlich					Kategorie	Sicherheit	
3	Revision	1.0	Wird autom	atisch erzeugt			Phase	BETA	
4									
5	Zusammenfassung	Unbefugte Benutzer dürfen nicht auf Dienste zugreifen können.				Ben	utzerdefinie	rte Eigensch	aften
6	Beschreibung	Während der Alpha-Entwicklungsphase war es möglich, auf Dienste zuzugreifen, ohne eine gültige Benutzername/Passwort-Kombination anzugeben. Der Server muss nun gegen unberechtigten Zugriff gesichert werden.							
7		Allgemeine	Eigenschaft	en					
8	Priorität	HIGH							
9	Status	APPROVED							
10									
11									

Abbildung 12.2. Beispiel für eine Excel-Tabelle mit Anforderungen

12.3.1.1. Allgemeine Eigenschaften

In der unten dargestellten Tabelle sind die Zellkoordinaten für die Werte der einzelnen allgemeinen Eigenschaften der Anforderungen dargestellt. Die ID der Anforderung wird beim Import automatisch erzeugt. Keines dieser Felder muss zwingend befüllt werden. Einige Felder können nur vordefinierte Werte annehmen. Diese sind in Klammern aufgeführt.

Koordinate	Wert
B1	ID (nur im Export verwendet)
B2	Name
B3	Revision (nur im Export verwendet)
B5	Zusammenfassung
B6	Beschreibung
B8	Priorität (LOW, MEDIUM, HIGH)
B9	Status (NEW, APPROVED, LOCKED, INVISIBLE)

Tabelle 12.4. Koordinaten für allgemeine Eigenschaften von Anforderungen

12.3.1.2. Benutzerdefinierte Eigenschaften

Anforderungen können benutzerdefinierte Eigenschaften enthalten. Diese werden als Name/ Wert-Paare ab Zelle G:1 und H:1 abwärts aufgelistet. Damit der Import erfolgreich ist, müssen die Namen der aufgelisteten Eigenschaften mit den benutzerdefinierten Eigenschaften übereinstimmen, die für das vom Import betroffene Projekt definiert wurden.

Der Name in der Namensspalte und der Name einer benutzerdefinierten Eigenschaft in Klaros-Testmanagement müssen exakt übereinstimmen, andernfalls kann der Wert nicht korrekt verarbeitet werden. Wird eine leere Namenszelle gefunden, stoppt die Verarbeitung. Das Wertfeld für Aufzählungseigenschaften muss exakt mit dem Namen der zu verarbeitenden Definition der Aufzählungseigenschaftswerte übereinstimmen.

Koordinaten	Wert
G1 (ff)	Eigenschaftsname
H1 (ff)	Eigenschaftswert

Tabelle 12.5. Koordinaten für benutzerdefinierte Eigenschaften

12.3.2. Vorbedingungen

Um Anforderungen aus Excel-Tabellen importieren zu können, müssen folgende Vorbedingungen erfüllt sein:

- 1. Die zu importierende Datei muss in den Formaten XLS oder XLSX vorliegen.
- 2. Das Projekt, in welches die Anforderungen importiert werden soll, muss bereits erstellt sein.
- 3. Benutzerdefinierte Eigenschaften (können nur in der Klaros-Testmanagement Enterprise Edition erstellt werden!) müssen bereits im entsprechenden Projekt definiert sein.

12.3.3. Ausführen

Für den Import von Excel-basierten Anforderungen steht eine REST-Schnittstelle zur Verfügung, über die Daten über die Kommandozeile oder anderen Anwendungen hochgeladen werden können.

Das folgende Beispiel zeigt, wie eine Excel-Tabelle mit Anforderungen in das Projekt mit der ID P00001 unter Verwendung der Curl-Kommandozeile importiert werden kann. Curl sollte in jeder Linux-Distribution enthalten sein. Für die Microsoft Windows-Betriebssystemfamilien ist es als Teil der Cygwin-Distribution http://www.cygwin.com/ oder als Kommandozeilenwerkzeug von http://curl.haxx.se/download.html erhältlich.

curl -v -T Requirements.xls "<klaros-app-url>/seam/resource/rest/import/requirement/xls?config=P00001\
&username=user&password=secret"

Beispiel 12.3. Anforderungen aus Excel-Tabellen über die Kommandozeile importieren

12.4. Importieren und Synchronisieren von Anforderungen aus XML

Mit Klaros-Testmanagement können Anforderungen sowohl importiert als auch aus XML-Dateien synchronisiert werden.

Ein Import ist auf eine einmalige Aktion beschränkt, die immer neue Instanzen der gelieferten Anforderungen erzeugt. Die Synchronisierung hingegen ermöglicht es, vorhandene Anforderungen mit dem Inhalt der XML-Datei zu aktualisieren und bei Bedarf neue Revisionen davon zu erstellen.

Das Format der XML-Datei ist in <u>Anhang D, Dateiformat für Anforderungs-Import</u> beschrieben. Das XML-Schema ist unter der folgenden URL erhältlich: https://www.klaros-testmanagement.com/files/schema/klaros-requirements-1.0.xsd.

12.4.1. Vorbedingungen

Um Anforderungen aus XML-Dateien importieren zu können, müssen folgende Vorbedingungen erfüllt sein:

- 1. Die zu importierende Datei muss im *.XML-Format vorliegen.
- 2. Das Projekt, in welches die Anforderungen importiert werden soll, muss bereits erstellt sein.
- 3. Wenn benutzerdefinierte Eigenschaften importiert werden sollen, müssen sie vor dem Importvorgang im Projekt angelegt werden.
- 4. Wenn die importierten Anforderungen via externalTestCaseId Testfällen zugeordnet werden sollen, müssen Testfälle mit dieser externen ID bereits vorliegen.

12.4.2. Importieren

Für den Import von XML-basierten Anforderungen steht eine REST-Schnittstelle zur Verfügung, über die Daten über die Kommandozeile oder anderen Anwendungen hochgeladen werden können.

Das folgende Beispiel zeigt, wie eine XML-Datei mit Anforderungen in das Projekt mit der ID P00001 unter Verwendung der Curl-Kommandozeile importiert werden kann. Curl sollte in jeder Linux-Distribution enthalten sein. Für die Microsoft Windows-Betriebssystemfamilien ist es als Teil der Cygwin-Distribution http://www.cygwin.com/ oder als Kommandozeilenwerkzeug von http://curl.haxx.se/download.html erhältlich.

curl -v -T Requirements.xml "<klaros-app-url>/seam/resource/rest/import/requirement/xml?config=P00001\ &username=user&password=secret"

Beispiel 12.4. Anforderungen aus XML-Dateien über die Kommandozeile importieren

12.4.3. Synchronisierung

Zur Synchronisierung von XML-basierten Anforderungen steht eine andere REST-Schnittstelle zur Verfügung (endet mit /sync/requirement/xml). Sie versteht dasselbe Importformat, wie es für das Importieren von Anforderungen definiert ist, erfordert jedoch das Vorhandensein zusätzlicher Elemente für den korrekten Betrieb.

Im Gegensatz zu einem Import dient eine Synchronisierungsaktion dazu, den in Klaros-Testmanagement gespeicherten Satz von Anforderungen aus beliebigen externen Anwendungen regelmäßig zu aktualisieren.

Die Synchronisation unterstützt das Erstellen und Aktualisieren sowie das Überarbeiten von Anforderungen. Änderungen an den Anforderungen werden in der Klaros-Testmanagement Datenbank für jedes Feld bei jeder Synchronisierungsaktion berücksichtigt.



Synchronisierung überschreibt lokale Änderungen!

Beachten Sie, dass bei der Synchronisierung bereits gespeicherte Änderungen überschrieben werden.



externalId

Das Element externalId (Siehe <u>Abschnitt D.11, "<externalId>"</u>) identifiziert eine Anforderung für nachfolgende Synchronisierungsversuche. Es ist ein Pflichtfeld bei Verwendung dieser Schnittstelle. Wird dieser Wert geändert, wird ein neues Anforderungsobjekt erstellt, anstatt den Inhalt der zuvor erstellten Anforderung zu ersetzen.



revision

Das Element revision (Siehe <u>Abschnitt D.15, "<revision>"</u>) identifiziert die Version einer Anforderung für nachfolgende Synchronisierungsversuche.

- Ist der Wert leer, so wird die aktuell ausgewählte Revision der Anforderung aktualisiert.
- Wird ein neue Revision eingegeben, so wird eine neue Version der Anforderung erstellt und aktualisiert.
- Falls eine ältere Revision eingegeben wird, wird die eingegebene Revision aktualisiert.

Das folgende Beispiel zeigt, wie eine XML-Datei mit Anforderungen in das Projekt mit der ID P00001 unter Verwendung der Curl-Kommandozeile synchronisiert werden kann. Curl sollte in jeder Linux-Distribution enthalten sein. Für die Microsoft Windows-Betriebssystemfamilien ist es als Teil der Cygwin-Distribution http://www.cygwin.com/ oder als Kommandozeilenwerkzeug von http://curl.haxx.se/download.html erhältlich.

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<r:container xmlns:r="http://klaros-testmanagement.com/export-requirements-1.0">
    <r:requirements>
    <r:requirement>
    <r:attributes/>
    <r:externalId>RTM-00001</r:externalId>
    <r:revision>1.0</r:revision>
    <r:priority>MEDIUM</r:priority>
    <r:shortname>Remote-controlled door panels / unlocking</r:shortname>
    <r:summary>Doors must me remote controllable.</r:summary>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </r:requirement>
    </reduct
    </reduct
```

Beispiel 12.5. XML-Anforderungssynchronisation über die Kommandozeile

curl -v -T Requirements.xml "<klaros-app-url>/seam/resource/rest/sync/requirement/xml?config=P00001\ &username=user&password=secret"

Beispiel 12.6. XML-Anforderungsimport über die Kommandozeile

12.5. Testergebnisse importieren

Testautomatisierungswerkzeuge erzeugen Testergebnisse typischerweise in Dateiform. Diese Dateien können über eine REST-Schnittstelle importiert und automatisch in Testergebnisse umgewandelt werden.

12.5.1. Das JUnit XML++ Format

JUnit XML++ ist das von Klaros-Testmanagement intern verwendete, generische Format für den Import von Testfallergebnissen aus externen Quellen. Dieses Format ist aufwärtskompatibel mit dem JUnit XML Format.

Durch Erweiterung des JUnit XML Format bietet das JUnit XML++ Format Möglichkeiten einzelne Testschritte eines Tests zu beschreiben, sowie binäre Anhänge in ein XML-Dokument einzubetten. Zudem bietet es mit dem Ergebnistyp inconclusive die Möglichkeit nicht eindeutige Ergebnisse darzustellen.

12.5.1.1. Das JUnit XML Format

Eine eindeutige formale JUnit XML-Format-Spezifikation existiert leider nicht, weil JUnit selbst keine XML-Berichte erstellt. Die XML-Berichterstellung resultiert gewöhnlich aus der Aufbereitung des JUnit Aufrufs innerhalb eines Build-Tools wie bspw. eines Ant JUnit-Tasks, des Maven Surefire Plugin oder Gradle. Das XML-Berichtsformat wurde erstmalig in Ant eingeführt und später von Maven und Gradle angepasst.

Das JUnit XML Format konnte sich mittlerweile über Tool- und Programmiersprachen- grenzen hinweg als ein generisches (Quasi-)Standardformat für Testfallergebnisse etablieren und wird von einer Vielzahl von Automatisierungswerkzeugen unterstützt. Trotz der Uneindeutigkeit des Formats, wurden in der Vergangenheit mehrfach Versuche vorgenommen, das Format in Form eines XML-Schemas zu dokumentieren. Ausführlich kommentierte Schemadefinitionen finden sich bspw. hier: https://llg.cubic.org/docs/junit/ und hier: https://github.com/windyroad/Junit-Schema/blob/master/JUnit.xsd.

12.5.1.2. Das JUnit XML++ Format

- Beschreiben und Bewerten von Tests auf Testschrittebene
- Ergänzen der Testergebnisse um binäre Anhänge (Bilder, Dokumente, etc.)
- Unterstützen von weiteren Testbewertungen (unklar)

Diese haben wir dem JUnit XML Format hinzugefügt, ohne den bestehenden Syntax zu ändern. Dieses neue Format nennen wir im weiteren JUnit XML++.

12.5.1.2.1. Testschrittergebnisse

Innerhalb eines testcase-Elements können nun beliebig viele teststep-Elemente vorkommen. Diese enthalten zwingend ein time-Attribut, das die Ausführungsdauer des Testschritts in Sekunden angibt.

Wie auch bei Testfällen, steuern error, failure oder skipped Elemente innerhalb eines Testschritts deren Bewertung. Ebenfalls analog zu den Testfällen werden system-out sowie system-err Elemente unterstützt. Enthalten diese Inhalte, so werden sie automatisch als binäre Anhänge dem beim Import erzeugten Testfallergebnis hinzugefügt.

Optional können dem Testschrittergebnis noch die Elemente action, precondition, postcondition und expectedResult untergeordnet werden.

Sind diese vorhanden, werden sie in Klaros-Testmanagement in die entsprechenden Felder eines Testschrittergebnisses übernommen. Die Angabe ist nicht zwingend erforderlich, aber zumindest für das Feld action zu empfehlen um einzelne Testschritte besser identifizieren zu können.

Beispiel 12.7. Beispiel Testschrittergebnis.xml

12.5.1.2.2. Anhänge

Für die Dokumentation eines Test(schritt)ergebnisses kann es sinnvoll sein, dieses neben der textuellen Beschreibung um eine Datei (Screenshot, Dokument etc.) zu ergänzen. Um dies zu ermöglichen, können in den Elementen system-out und system-err zusätzliche Inhalte definiert werden.

Nach erfolgreichem Extrahieren und Speichern werden diese Inhalte nicht in eventuell zusätzlich angelegte system-out/system-err Attachments übernommen. Die Inhalte werden in Anlehnung an das Jenkins Plugin JUnit Attachments definiert und können in drei verschiedenen Ausprägungen vorkommen:

Base64 Encoded Attachment	<pre>Syntax: [[ATTACHMENT_DATA «attached-file.xyz» «encoded data»]]</pre>
	In "encoded data" wird ein base64-encodierter binärer Inhalt erwartet. Dieser wird nach erfolgreicher Dekodierung unter dem Dateinamen attached-file.xyz als Anhang am Testfall- ergebnis oder Testschrittergebnis gespeichert.
	Bei dieser Methode wird also den gesamten Anhang als Teil der Ergebnisdatei gespeichert. Dies erfordert mehr Speicher- platz, ermöglicht aber jederzeit den Zugriff auf den Inhalt.
URL Attachment	Syntax: [[ATTACHMENT_URL /«url/to/attached-file.xyz»]]
	Es wird versucht, den Inhalt der URL url/to/attached-file. xyz herunterzuladen und unter dem Dateinamen atta- ched-file.xyz als Anhang zu speichern.

Bei dieser Methode wird erwartet, dass das Attachment im Netz bereitgestellt ist. Es ist zwingend notwendig, dass die Klaros-Instanz zum Importzeitpunkt Zugang auf diese URL hat, damit die Extraktion gelingt.

 File Attachment
 Syntax: [[ATTACHMENT]/«path/to/attached-file.xyz»]]

Es wird versucht, den Inhalt der Datei path/to/attached-file. xyz herunterzuladen und unter dem Dateinamen attached-file.xyz als Anhang zu speichern.

Bei dieser Methode wird erwartet, dass das Attachment im lokalen Filesystem bereitgestellt ist. Es ist zwingend notwendig, dass die Klaros-Instanz zum Importzeitpunkt Zugang auf diese Datei hat, damit die Extraktion gelingt.

12.5.1.2.3. Testergebnisse

Sowohl bei Testfallergebnissen (testcase) als auch bei Testschrittergebnissen (teststep), ist es möglich neben error, failure oder skipped auch inconclusive (unklar) als Ergebnistyp zu setzen, da dieser zwar in Klaros-Testmanagement nicht aber in JUnit unterstützt wird.

12.5.2. Vorbedingungen

Um Testergebnisse zu importieren müssen folgende Vorbedingungen erfüllt sein:

1. Das Projekt, die Iteration, die Testumgebung und das Testsystem, das die Ergebnisse des importierten Testfalls oder der Testsuite enthalten soll, müssen bereits erstellt sein.

12.5.3. Ausführen

Die URL der Importschnittstelle finden Sie unter http://localhost:18080/klaros-web/seam/ resource/rest/importer. Der Inhalt wird über eine HTTP-PUT-Anforderung unter Verwendung der obigen URL und verschiedener URL-Query-Parameter übertragen.



Der Ausdruck <klaros-app-url>

Die oben gezeigte URL http://localhost:18080/klaros-web ist die Standard-URL der Klaros-Anwendung, auf die der vom Host zugreift und kann je nach Ihrer Einrichtung variieren. In diesem Kapitel wird stattdessen der Begriff <klaros-app-url>verwendet, um dies wiederzugeben.

Die folgenden Parameter werden unterstützt:

config	Die ID des Projekts, in das die Ergebnisse importiert werden sollen (z.B. P0001).
iteration	Die ID der Iteration, zu der die Ergebnisse in Beziehung gesetzt werden sollen (z.B. ITR00001). Dieser Parameter ist optional.

env	Die ID der Testumgebung, in der den (z.B. ENV00001). Bitte stellen umgebung bereits im Projekt def Import beginnen.	die Tests durchgeführt wur- Sie sicher, dass diese Test- iniert ist, bevor Sie mit dem
sut	Die ID des Testsystems, in dem die (z.B. SUT00001). Bitte stellen Sie s bereits im Projekt definiert ist, bev nen.	e Tests durchgeführt wurden sicher, dass das Testsystem or Sie mit dem Import begin-
type	Der Typ des Importformats. Die unterstützt:	e folgenden Typen werden
	aunit	AUnit
	boost	Boost Test
	Check	Check
	cpptest	CppTest
	cppunit	CppUnit
	ctest	ctest (CMake)
	cunit	CUnit
	fitnesse	Fitnesse
	fpunit	Free Pascal Unit
	gtester	GLib/gtester
	googletest	GoogleTest
	jbehave	JBehave
	jmeter	JMeter
	jsunit	JsUnit
	jubula	Jubula / GUldancer
	junit	JUnit / Gauge / Robot Framework / Selenium / Squish / TestNG
	mbunit	MBUnit
	mstest	MSTest
	nunit	NUnit
	phpunit	PHPUnit
	qftest ²⁹⁴	QF-Test

	qtestlib	QTestLib		
	tessy	TESSY		
	testcomplete	TestComplete		
	tusar	TUSAR		
	uft	Unified Functional Testing (UFT) / QuickTest Profes- sional (QTP)		
	cpptestunit	UnitTest++		
	valgrind	Valgrind		
	xUnit.net	xunitdotnet		
job	Die optionale ID der Aufgabe, der dieser Testlauf zugeordnet werden soll (z.B. JOB00001). Falls angegeben, muss die Auf- gabe vor Beginn des Imports bereits im Projekt vorhanden sein.			
time	Der Zeitpunkt der Testausführung. Das Format für diese Zeit- angabe ist dd.MM.yyyyy_HH:mm. Dieser Parameter kann ver- wendet werden, um den Ausführungszeitpunkt des Testlaufs zu überschreiben. Ansonsten wird der Zeitpunkt des Imports verwendet.			
createTestSuiteResults	Wenn dieser Parameter auf true gesetzt ist, so werden auto- matisch Testsuiteergebnisse aus den in der Ergebnisdatei enthalten Informationen erzeugt. Diese Informationen können je nach Importformat variieren. Zusätzlich wird eine entspre- chende Testsuite für das Testsuiteergebnis erstellt, falls diese noch nicht vorhanden war.			
autoCreateTestCases	Ist dieser Parameter auf true gesetzt, so werden automatisch Testfälle aus den in der Ergebnisdatei enthalten Informatio- nen erzeugt, wenn kein zugehöriger Testfall gefunden werden konnte.			
username	Der Benutzername für den Import.			
password	Das Passwort für den Import.			

Ein vollständiges Beispiel für eine QF-Test-Import-URL würde wie folgt aussehen:

http://localhost:18080/klaros-web/seam/resource/rest/importer?\
config=P00001&env=ENV00001&sut=SUT00001&type=qftest&time=01.03.2011_12:00&username=me&password=secret

Beispiel 12.8. Beispiel QF-Test Import-URL

Die Ergebnisdatei ist im HTTP-Request-Body enthalten.

Mit dem Kommandozeilen-Werkzeug **curl** kann der Import unter Linux oder Cygwin unter Windows in einer einzigen Befehlszeile ausgelöst werden.

```
curl -v -H "Content-Type: text/xml" -T <test result file> \
    "<klaros-app-url>/seam/resource/rest/importer?config=P00001&env=ENV00001&sut=SUT00001&type=junit\
&time=23.05.2011_14:55"
```

Beispiel 12.9. Curl Kommandozeilen Beispiel



Inkompatibler Windows Powershell-Curl-Alias

Windows Powershell definiert ebenfalls einen Alias namens **curl**, der für diesen Zweck verwendet werden kann, aber einen völlig anderen Satz von Argumenten benötigt. Nutzen Sie das Linux-kompatible Curl-Programm unter Powershell, verwenden Sie **curl.exe** anstelle von **curl**.

```
curl -Method put -InFile <test result file> -Uri \
"<klaros-app-url>/seam/resource/rest/importer?config=P00001&env=ENV00001&sut=SUT00001&type=junit\
&time=23.05.2011_14:55"
```

Beispiel 12.10. Powershell curl Alias Kommandozeilen-Beispiel

12.5.4. Jenkins Plugin

Das Plugin integriert den kontinuierlichen Integrationsserver Jenkins mit Klaros-Testmanagement und spielt die Testergebnisse eines Jenkins-Builds in Klaros-Testmanagement ein. Die Testergebnisse werden in der Datenbank für weitere Auswertungen gespeichert. Die Installations- und Konfigurationsanleitung für das Plugin finden Sie auf der Seite Jenkins GitHub.

12.6. Testergebnisse exportieren

Klaros-Testmanagement besitzt eine JUnit XML Export Schnittstelle zum exportieren von Testergebnissen.

12.6.1. Ausführen

Die URL der Exportschnittstelle finden Sie unter http://localhost:18080/klaros-web/seam/ resource/rest/export/result/junit. Der Inhalt wird über eine HTTP-GET-Anforderung unter Verwendung der obigen URL und verschiedener URL-Query-Parameter übertragen.



Der Ausdruck <klaros-app-url>

Die oben gezeigte URL http://localhost:18080/klaros-web ist die Standard-URL der Klaros-Anwendung, auf die der vom Host zugreift und kann je nach Ihrer Einrichtung variieren. In diesem Kapitel wird stattdessen der Begriff <klaros-app-url>verwendet, um dies wiederzugeben.

Die folgenden Parameter werden unterstützt:

config

Die ID des Projekts, aus dem die Ergebnisse exportiert werden sollen (z.B. P0001).

env	Die ID der Testumgebung, in der die Tests durchgeführt wur- den (z.B. ENV00001).
sut	Die ID des Testsystems, in dem die Tests durchgeführt wurden (z.B. SUT00001).
job	Die ID der Aufgabe, dessen Ergebnisse exportiert werden sol- len (z.B. JOB00001).
testRun	Die ID des Testlaufs, dessen Ergebnisse exportiert werden sol- len (z.B. TRU0000001).
username	Der Benutzername für den Export.
password	Das Passwort für den Export.

Es besteht die Möglichkeit sowohl einzelne Testergebnisse einer Aufgabe oder eines Testlaufs als auch alle Testergebnisse eines Testsystems oder einer Kombination aus Testsystem und Testumgebung zu exportieren. Über die Angabe der Parameter kann dies gesteuert werden.

Mit dem Kommandozeilen-Werkzeug **curl** kann der Export unter Linux oder Cygwin unter Windows in einer einzigen Befehlszeile ausgelöst werden.

In diesem Beispiel werden die Ergebnisse der Aufgabe JOB00001 aus dem Projekt P00001 exportiert.

Beispiel 12.11. Curl Kommandozeilen Beispiel



Inkompatibler Windows Powershell-Curl-Alias

Windows Powershell definiert ebenfalls einen Alias namens **curl**, der für diesen Zweck verwendet werden kann, aber einen völlig anderen Satz von Argumenten benötigt. Nutzen Sie das Linux-kompatible Curl-Programm unter Powershell, verwenden Sie **curl.exe** anstelle von **curl**.

```
curl -Method get -Uri \
```

"<klaros-app-url>/seam/resource/rest/export/result/junit?config=P00001&testRun=TRU0000001&username=me&password=secret"

In diesem Beispiel werden die Ergebnisse des Testlaufs TRU0000001 aus dem Projekt P00001 exportiert.

Beispiel 12.12. Powershell curl Alias Kommandozeilen-Beispiel

12.7. Tabelleninhalte exportieren



Mit Klaros-Testmanagement kann der Inhalt aller Tabellen in die Formate Excel, PDF oder XML exportiert werden. Die aktuellen Filter- und Sortiereinstellungen werden dabei berücksichtigt.

= 👱 KLAR	o s test management	DE Finanz-Tracker 🖀 🗮	۹ 🚱 ۲ 🛓 ۲
C Definieren	Testsuiten		
Planen	Neu	N 20 V AV	Speichern Verwerfen
🏟 Ausführen	ID ≠ Q Revision Name ‡ TS00008 ⊕ 1.0 Überprüfen Sie die Verbindung zum Server	Testsystem	Excel-Export
🕒 Auswerten	T S00007 合 1.0 Daueraufträge bearbeiten T S00006 合 1.0 Übersicht T S00004 1.0 Verbingungen überprüfen	Finance Tracker 1.0.0	XML-Export 7 C 口 合 印 2 C 口 合 印
🖌 Konfigurieren	TS00003 1.0 Aktivitäten verschiedener Zelträume anzeigen TS00002 1.0 Überweisungen testen TS00002 1.0 Deschargerd efficielleren		
	ID Revision Name	Testsystem	Testfälle Aktion
	Neu		Speichern Verwerfen
		3	

Abbildung 12.3. Tabelleninhalte in Excel-Tabellen exportieren

12.8. Backup und Wiederherstellen

Das Verschieben von Daten zwischen verschiedenen Datenbankinstallationen oder zum selektiven Import von Daten kann über XML-Dateien erfolgen. <u>Abschnitt 10.6, "Sicherung"</u> erklärt die Import- und Exportfunktionalität im Detail.

12.8.1. Backup via REST

Klaros-Testmanagement Enterprise Edition enthält eine REST-basierte Schnittstelle, die Zugriff auf Sicherungsdateien einzelner Projekte gewährt. Der folgende Beispiel-URI gibt die XML-Datei der Sicherung des Projekts P00004 zurück: http://{host}:{port}/klaros-web/seam/resource/rest/ io/backup/project/P00004.



Anmerkung

Weitere Informationen darüber, wie Sie auf REST-basierte Schnittstellen zugreifen können, finden Sie Kapitel 13, *Die Remote API*.

Kapitel 13. Die Remote API



Nur in der Klaros-Testmanagement Enterprise Edition verfügbar

13.1. Überblick

Über eine REST-basierte Schnittstelle können Daten wie Testfälle, Ergebnisse oder Informationen über Benutzer entweder im XML- oder im JSON-Format abgerufen werden. Die API ist über die URL http://{host}:{port}/klaros-web/api/v1/resources/ erreichbar.

	larald Martens	:
● GET ● POST ● PUT ● PATCH ● DELETE ● HEAD ● OPT	IONS Other	
Raw Form Headers		
Authorization	Basic YWRtaW46YWRtaW4=	×
name	value	×
ADD NEW HEADER		
	Clear	Send
Status: 200: OK 🕐 Loading time:289ms		
Response headers (0)	Request headers (0) Redirects (0)	
	-	
Raw JSON Response		
COPY TO CLIPBOARD SAVE AS FILE		

Abbildung 13.1. JSON-Ausgabe nach der Suche nach einem Benutzernamen.

13.2. REST-Clients

Es gibt zahlreiche Möglichkeiten, eine Verbindung mit der Remote-API herzustellen. Die folgenden Listen geben einen Überblick über einige frei verfügbare Clients.

Browser Plugins

Advanced REST client plugin for Google Chrome

Google Chrome Web Store

RESTED for Firefox

Firefox Add-Ons

RestMan Add-On for Opera

Opera Add-Ons

Standalone Clients

• Wget

https://www.gnu.org/software/wget/

• cURL

http://curl.haxx.se/

13.3. Dokumentation

Klaros-Testmanagement wird mit einer interaktiven Dokumentation für die Remote-API geliefert, auf die über einen Browser unter http:/{host}:{port}/klaros-web/api-doc/resources/ zugegriffen werden kann.

Klaros-Testmanagement R	EST Serv	rice AP	: /api/v1/resources/systemsUnderTest/{id}/{testRuns}		Resourc	es Data Types	
GET /api/v1/resources/systemsU	Home / Resources / /api/v1/resources/systemsUnderTest/[id]/(testRuns}						
Back to Top	/api/v1/resources/systemsUnderTest/{id}/{testRuns}						
	Returns the test runs executed for this system under test.						
	GET	/api/\	r1/resources/systemsUnderTest/{id}/{testRuns} ♂				
	Returns Request	the test i Parame	uns executed for this system under test. ers				
	name	type	description		default	constraints	
	id	path	the system under test id				
	accept	query	The Media Type information accepted by the client. Currently supported values are: application and application/xml.	on/json	application/json		
	limit	query	Page limit when retrieving paged lists of objects. The default value is 0 and disables paging.		0	int	
	offset	query	Paging offset when retrieving paged lists of objects.		0	int	
	Respons	e Body					
	media	type	data type	descriptio	n		
	applicat	tion/json	TestRuns (JSON)	the test rur	ns		
	applicat	tion/xml	testRuns (XML)				

Abbildung 13.2. Remote-API Dokumentation

Anhang A. Zugriffsberechtigungen

A.1. Rollenberechtigungen – Community Edition

Aktionen/Rollen	Administrator	Manager	Tester	Gast
Testfälle ausführen	0	0		×
Berichte erzeugen	0	0	0	0
Testergebnisse ansehen	0	0	0	0
Sichern und Wiederherstellen ^a	0	0	×	×
Testobjekte erzeugen, bearbeiten und löschen	0	0	×	×
Benutzer mit der Rolle <i>Tester</i> anlegen, bear- beiten und löschen ^b	0	0	×	×
Benutzer mit den Rollen <i>Administrator</i> oder <i>Manager</i> anlegen, bearbeiten und löschen	0	×	×	×
Systemeinstellungen bearbeiten ^b	0	×	×	×

^a Globale Manager dürfen jedes Projekt wiederherstellen. Sie dürfen jedoch nur Projekte exportieren, für die keine projektspezifischen Rollen definiert sind oder in denen sie die Rolle *Manager* besitzen.

^bWird nicht durch die Projektrolle beeinflusst.

A.2. Rollenberechtigungen – Enterprise Edition

Aktionen/Rollen	Administrator	Manager	Tester	Gast
Dashboard konfigurieren	O	\bigcirc	\bigcirc	0
Aufgaben ausführen	O	\bigcirc	\bigcirc	×
Excel-Tabellen exportieren	O	0	\bigcirc	0
Benutzerdefinierte Berichte erzeugen	O	\bigcirc	\bigcirc	0
Generische Volltextsuche	O	0	\bigcirc	0
Testlaufausführung aufzeichnen	O	0	0	×
Aufgaben erzeugen und zuweisen	O	0	×	×
Benutzerdefinierte Berichte erzeugen ^a	O	0	×	×
Benutzerdefinierte Felder erstellen	O	\bigcirc	×	×
LDAP-Authentifizierung konfigurieren ^a	0	×	×	×

^aWird nicht durch die Projektrolle beeinflusst

Tabelle A.2. Rollenberechtigungen – Enterprise Edition

Tabelle A.1. Rollenberechtigungen – Community Edition

Anhang B. Die Klaros Object Model API Referenz

B.1. Klaros Object Model API Reference

- B.1.1. de.verit.klaros.core.model
- B.1.1.1. IKlarosAttachment

This interface provides access to a binary attachment.

B.1.1.1.1. Synopsis

```
public interface IKlarosAttachment
    extends IKlarosRepositoryEntity<KlarosAttachment> {
```

```
// Public Methods
public abstract long getSize();
}
```



B.1.1.1.2. getSize()

public abstract long getSize();

Gets the size in bytes of this attachment.

Parameters

return

the size

B.1.1.2. IKlarosCategoryNode

A Klaros Category Node.

B.1.1.2.1. Synopsis

}

public interface IKlarosCategoryNode extends IKlarosPersistentObject {

```
// Public Methods
public abstract CategoryTree getCategoryTree();
public abstract String getDescription();
public abstract KlarosCategoryNode getParent();
public abstract String getShortname();
public abstract List<KlarosCategoryNode> getSubCategories();
```



B.1.1.2.2. getCategoryTree()

public abstract CategoryTree getCategoryTree();

Gets the category tree.

Parameters

return

the category tree

B.1.1.2.3. getDescription()

public abstract String getDescription();

Gets the description. Parameters the description return B.1.1.2.4. getParent() public abstract KlarosCategoryNode getParent(); Gets the parent category node. Parameters the parent return B.1.1.2.5. getShortname() public abstract String getShortname(); Gets the shortname. Parameters the shortname return B.1.1.2.6. getSubCategories() public abstract List<KlarosCategoryNode> getSubCategories(); Gets the sub categories. Parameters the sub categories return B.1.1.3. IKlarosCategoryTree A category tree. B.1.1.3.1. Synopsis public interface IKlarosCategoryTree extends IKlarosNamedEntity<KlarosCategoryTree> { // Public Methods

// Public Methods
public abstract String getDescription();
public abstract KlarosCategoryNode getRootNode();
public abstract String getShortname();

}



B.1.1.3.2. getDescription()

public abstract String getDescription();

Gets the description.

return

the description

B.1.1.3.3. getRootNode()

public abstract KlarosCategoryNode getRootNode();

Gets the root category node.

Parameters

return

the root

Gets the shortname.

Parameters

return

the shortname

B.1.1.4. IKlarosConfiguration

This interface provides access to data of a test project.

B.1.1.4.1. Synopsis

public interface IKlarosConfiguration
 extends IKlarosLabeledObject<KlarosConfiguration> {

```
// Public Methods
public abstract String getDescription();
public abstract Set<KlarosTestEnvironment> getEnvs();
public abstract Set<KlarosIssueManagement> getIssueManagementSystems();
public abstract Set<KlarosIssue> getIssues();
public abstract Set<KlarosCategoryTree> getIterationCategoryTrees();
public abstract Set<KlarosIteration> getIterations();
public abstract Set<KlarosJob> getJobs();
public abstract Set<KlarosCategoryTree> getRequirementCategoryTrees();
public abstract Set<KlarosRequirement> getRequirements();
public abstract KlarosRequirementsManagement getRequirementsManagementSystem();
public abstract Set<KlarosSUTImplementation> getSuts();
public abstract Set<KlarosCategoryTree> getSystemUnderTestCategoryTrees();
public abstract Set<KlarosCategoryTree> getTestCaseCategoryTrees();
public abstract Set<KlarosTestCase> getTestCases();
public abstract Set<KlarosCategoryTree> getTestEnvironmentCategoryTrees();
public abstract Set<KlarosTestRun> getTestRuns();
public abstract Set<KlarosCategoryTree> getTestSegmentCategoryTrees();
public abstract Set<KlarosCategoryTree> getTestSuiteCategoryTrees();
public abstract Set<KlarosTestSuite> getTestSuites();
public abstract Boolean isSecured();
```

}


B.1.1.4.2. getDescription()

public abstract String getDescription();

Returns the project description.

Parameters

return

The description of the project.

B.1.1.4.3. getEnvs()

public abstract Set<KlarosTestEnvironment> getEnvs();

Returns the project test environments.

Parameters

return

Set containing the test environments of the project.

B.1.1.4.4. getIssueManagemen public abstract Set <kla< th=""><th>tSystems() rosIssueManagement> getIssueManagementSystems();</th></kla<>	tSystems() rosIssueManagement> getIssueManagementSystems();
Returns the related issue m	anagement systems.
	Parameters
return	Set containing the related issue management systems of the project.
B.1.1.4.5. getlssues() public abstract Set <kla< td=""><td>rosIssue> getIssues();</td></kla<>	rosIssue> getIssues();
Returns the issues.	
	Parameters
return	KlarosSet containing the issue objects of the project.
B.1.1.4.6. getIterationCategory] public abstract Set <kla< td=""><td><pre>Frees() rosCategoryTree> getIterationCategoryTrees();</pre></td></kla<>	<pre>Frees() rosCategoryTree> getIterationCategoryTrees();</pre>
Returns the project iteratior	n category trees.
	Parameters
return	Set containing the iteration category trees of the project.
B.1.1.4.7. getIterations() public abstract Set <kla< td=""><td><pre>rosIteration> getIterations();</pre></td></kla<>	<pre>rosIteration> getIterations();</pre>
Returns the project iteratior	۱.
	Parameters
return	KlarosSet containing the iteration objects of the project.
B.1.1.4.8.getJobs() public abstract Set <kla< td=""><td>rosJob> getJobs();</td></kla<>	rosJob> getJobs();
Returns the project jobs.	
	Parameters
return	KlarosSet containing the job objects of the project.
B.1.1.4.9. getRequirementCated public abstract Set <kla< td=""><td>goryTrees() rosCategoryTree> getRequirementCategoryTrees();</td></kla<>	goryTrees() rosCategoryTree> getRequirementCategoryTrees();
Returns the project requirer	nent category trees.
	Parameters
return	Set containing the requirement category trees of the project.
B.1.1.4.10. getRequirements() public abstract Set <kla< td=""><td><pre>rosRequirement> getRequirements();</pre></td></kla<>	<pre>rosRequirement> getRequirements();</pre>

	Returns the project requirements.		
		Parameters	
	return	Set containing the requirement objects of the project.	
B.1.1.4.11. getRequirementsManagementSystem() public abstract KlarosRequirementsManagement getRequirementsManagementSystem();			
	Returns the related requirement n	nanagement system.	
		Parameters	
	return	Set containing the related requirement management system of the project.	
B.1.1	1.4.12.getSuts() public abstract Set <klarossu< td=""><td>TImplementation> getSuts();</td></klarossu<>	TImplementation> getSuts();	
	Returns the project SUTs (system	is under test).	
		Parameters	
	return	KlarosSet containing the SUT objects of the project.	
B.1.1.4.13. getSystemUnderTestCategoryTrees() public abstract Set <klaroscategorytree> getSystemUnderTestCategoryTrees();</klaroscategorytree>			
	Returns the project system under test category trees.		
		Parameters	
	return	Set containing the system under test category trees of the pro- ject.	
<pre>B.1.1.4.14. getTestCaseCategoryTrees() public abstract Set<klaroscategorytree> getTestCaseCategoryTrees();</klaroscategorytree></pre>			
	Returns the project test case cate	egory trees.	
		Parameters	
	return	Set containing the test case category trees of the project.	
B.1.1	1.4.15.getTestCases() public abstract Set <klaroste< td=""><td>stCase> getTestCases();</td></klaroste<>	stCase> getTestCases();	
	Returns the project test cases.		
		Parameters	
	return	Set containing the test case objects of the project.	
B.1.1	1.4.16. getTestEnvironmentCated public abstract Set <klarosca< td=""><td>goryTrees() tegoryTree> getTestEnvironmentCategoryTrees();</td></klarosca<>	goryTrees() tegoryTree> getTestEnvironmentCategoryTrees();	
	Returns the project test environm	ent category trees.	

Parameters

	return	Set containing the test environment category trees of the pro- ject.
B.1.1	.4.17.getTestRuns() public abstract Set <klarostes< td=""><td>tRun> getTestRuns();</td></klarostes<>	tRun> getTestRuns();
	Returns the project test runs.	
		Parameters
	return	Set containing the test run objects of the project.
B.1.1	.4.18. getTestSegmentCategory ⁻ public abstract Set <klaroscat< td=""><td>Trees() egoryTree> getTestSegmentCategoryTrees();</td></klaroscat<>	Trees() egoryTree> getTestSegmentCategoryTrees();
	Returns the project test segment of	category trees.
		Parameters
	return	Set containing the test segment category trees of the project.
B.1.1	.4.19. getTestSuiteCategoryTree public abstract Set <klaroscat< td=""><td>S() egoryTree> getTestSuiteCategoryTrees();</td></klaroscat<>	S() egoryTree> getTestSuiteCategoryTrees();
	Returns the project test suite cate	gory trees.
		Parameters
	return	Set containing the test suite category trees of the project.
B.1.1	.4.20. getTestSuites() public abstract Set <klarostes< td=""><td>tSuite> getTestSuites();</td></klarostes<>	tSuite> getTestSuites();
	Returns the project test suites.	
		Parameters
	return	Set containing the test suite objects of the project.
B.1.1	.4.21.isSecured() public abstract Boolean isSec	ured();
	Returns whether this project is see	cured or not.
		Parameters
	return	true if this project is secured, false if not.
B.1.1	.5. IKlarosEnumValue	
	A Custom property enumeration v	alue.
B.1.1	.5.1. Synopsis	
	public interface IKlarosEnumVa	lue extends IKlarosPersistentObject {
	// Public Methods	

```
public abstract String getValue();
}
```



B.1.1.5.2. getValue()

public abstract String getValue();

Gets the enumeration value.

Parameters

return

B.1.1.6. IKlarosExternalLink

the value

This interface provides access to externally stored information about an object.

B.1.1.6.1. Synopsis

public interface IKlarosExternalLink extends IKlarosPersistentObject {

```
// Public Methods
public abstract String getReference();
}
```



B.1.1.6.2. getReference()

public abstract String getReference();

Get the reference to the externally stored information.

Parameters

return

The reference to the information.

B.1.1.7. IKlarosExternalServer

This interface provides access to external systems.

B.1.1.7.1. Synopsis

}

public interface IKlarosExternalServer<T> extends IKlarosLabeledObject<T> {

```
// Public Methods
public abstract String getDescription();
public abstract String getType();
public abstract String getUrl();
```



B.1.1.7.2. getDescription()

public abstract String getDescription();

Get the description of the external system.

Parameters

return

The description of the external system.

B.1.1.7.3. getType()
 public abstract String getType();

Get the type of this system.

Parameters

return

The type

B.1.1.7.4. getUrl()

public abstract String getUrl();

Get the base url of this system.

Parameters

return

The url

B.1.1.8. IKlarosIssue

This interface provides access to a software issue.

B.1.1.8.1. Synopsis

}

public interface IKlarosIssue extends IKlarosNamedEntity<KlarosIssue> {

```
// Public Methods
@Deprecated(since="5.3.3")
public abstract KlarosSUTImplementation getAcceptedIn();
public abstract Date getCreationDate();
public abstract String getDescription();
public abstract String getExternalId();
public abstract KlarosIssueManagement getIssueManagement();
public abstract Date getLastSynched();
public abstract String getOwner();
public abstract String getPriority();
public abstract String getRemoteBrowseUrl();
public abstract String getReporter();
public abstract String getResolution();
public abstract String getState();
public abstract String getSubject();
public abstract Set<KlarosSUTImplementation> getSystemsUnderTest();
public abstract Set<KlarosTestCase> getTestCases();
public abstract boolean isResolved();
```



B.1.1.8.2. getAcceptedIn()

@Deprecated(since="5.3.3")
public abstract KlarosSUTImplementation getAcceptedIn();

Gets the system under test this issue has been accepted/detected in.

Parameters

return

the system under test

B.1.1.8.3. getCreationDate()

public abstract Date getCreationDate();

Gets the creation date of this issue.

Parameters

return

the creation date of this issue.

B.1.1.8.4. getDescription()
 public abstract String getDescription();

Get the description.

Parameters

return

The description of this issue.

B.1.1.8.5. getExternalId()

public abstract String getExternalId();

Gets the external id of this issue, if available. This id is generated by the external issue management system upon creation of the issue.

Parameters

return

the external id

B.1.1.8.6. getIssueManagement()

public abstract KlarosIssueManagement getIssueManagement();

Gets the issue management this issue belongs to.

Parameters

return

the issue management

Gets the last synchronization date of this issue.

Parameters

return

the last synchronization date of this issue.

B.1.1.8.8. getOwner()

public abstract String getOwner();

Gets the name of the owner/assignee of this issue.

Parameters

return

the owner name of this issue.

B.1.1.8.9. getPriority()

public abstract String getPriority();

Gets the priority of this issue. The format and content of this value varies with the issue management system and its configuration.

Parameters

return

the priority of this issue.

B.1.1.8.10. getRemoteBrowseUrl()

public abstract String getRemoteBrowseUrl();

Returns an url to browse the issue in the remote issue management system.

Parameters

return

the remote browse url

B.1.1.8.11. getReporter()

public abstract String getReporter();

Gets the name of the reporter of this issue.

Parameters

return

the reporter name of this issue.

B.1.1.8.12. getResolution()

public abstract String getResolution();

Gets the resolution of this issue. The format and content of this value varies with the issue management system and its configuration.

Parameters

return

the resolution of this issue.

B.1.1.8.13. getState()

public abstract String getState();

Gets the state of this issue. The format and content of this value varies with the issue management system and its configuration.

Parameters

return

the state string of this issue.

B.1.1.8.14. getSubject()

public abstract String getSubject();

Gets the subject of this issue.

Parameters

return

the subject string of this issue.

B.1.1.8.15. getSystemsUnderTest()

public abstract Set<KlarosSUTImplementation> getSystemsUnderTest();

Get the related systems under test.

Parameters

return

Set of systems under test which have detected this issue.

B.1.1.8.16. getTestCases()

public abstract Set<KlarosTestCase> getTestCases();

Get the related test cases.

Parameters

return

Set of test cases which have detected this issue.

B.1.1.8.17. isResolved()

public abstract boolean isResolved();

Checks if this issue is resolved.

Parameters

return

true, if resolved

B.1.1.9. IKlarosIssueManagement

The interface for accessing issue management systems.

B.1.1.9.1. Synopsis

public interface IKlarosIssueManagement
 extends IKlarosExternalServer<KlarosIssueManagement> {

```
// Public Methods
public abstract Set<KlarosIssue> getIssues();
public abstract String getProject();
}
```



B.1.1.9.2. getIssues()

public abstract Set<KlarosIssue> getIssues();

Gets the referenced issues of this system.

Parameters

return

the issues

B.1.1.9.3. getProject()

public abstract String getProject();

Get the optional project of this system.

Parameters

return

The project

B.1.1.10. IKlarosIteration

An iteration in a project.

B.1.1.10.1. Synopsis

public interface IKlarosIteration extends IKlarosNamedEntity<KlarosIteration> {

```
// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract Set<KlarosCategoryNode> getCategories();
public abstract StateDef getCurrentState();
public abstract String getDescription();
public abstract Date getDueDate();
public abstract Set<KlarosTestEnvironment> getEnvs();
public abstract Set<KlarosJob> getJobs();
public abstract String getShortname();
public abstract String getSuccessCriteria();
public abstract Set<KlarosTestRun> getTestRuns();
}
```



B.1.1.10.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this iteration.

Parameters

return

the attachments

B.1.1.10.3. getCategories()

public abstract Set<KlarosCategoryNode> getCategories();

Gets the categories this object belongs to. Each category node will belong to a different category tree.

		Parameters	5
	return	he category node	28
B.1.1	.10.4. getCurrentState() public abstract StateDef getCu	<pre>rrentState();</pre>	
	Gets the current state.		
		Parameters	5
	return	he current state	
B.1.1	.10.5. getDescription() public abstract String getDesc	<pre>ription();</pre>	
	Gets the description.		
		Parameters	5
	return	he description	
B.1.1	.10.6. getDueDate() public abstract Date getDueDa	e();	
	Gets the due date.		
		Parameters	5
	return	he due date	
B.1.1	.10.7.getEnvs() public abstract Set <klarostes< td=""><td>Environment> ge</td><td>tEnvs();</td></klarostes<>	Environment> ge	tEnvs();
	Gets the test environments associ	ted with this itera	ation.
		Parameters	5
	return	he test environm	ents
B.1.1	.10.8. getJobs() public abstract Set <klarosjob< td=""><td>getJobs();</td><td></td></klarosjob<>	getJobs();	
	Gets the jobs associated with this	eration.	
		Parameters	5
	return	he jobs	
B.1.1	.10.9. getShortname() public abstract String getSho	tname();	
	Gets the short name.		
		Parameters	5
	return	he short name	

B.1.1.10.10. getStartDate()

public abstract Date getStartDate();

Gets the start date.

Parameters

return

the start date

Gets the success criteria.

Parameters

return

the success criteria

B.1.1.10.12. getSuts()

public abstract Set<KlarosSUTImplementation> getSuts();

Gets the systems under test associated with this iteration.

Parameters

return

the systems under test

B.1.1.10.13. getTestRuns()

public abstract Set<KlarosTestRun> getTestRuns();

Gets the test runs associated with this iteration.

Parameters

the test runs

return

B.1.1.11. IKlaros Job

This interface provides access to the data of a job.

B.1.1.11.1. Synopsis

public interface IKlarosJob extends IKlarosNamedEntity<KlarosJob> {

// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract KlarosConfiguration getConfiguration();
public abstract Set<KlarosJobDependency> getDependentJobs();
public abstract String getDescription();
public abstract Date getDueDate();
public abstract KlarosTestEnvironment getEnv();
public abstract String getEstimatedTime();
public abstract Long getEstimatedTimeInMilliseconds();
public abstract JobPriority getJobPriority();
public abstract JobStatus getJobStatus();
public abstract KlarosJob getParent();
public abstract KlarosJob getParent();
public abstract Integer getProgress();

public abstract Set<KlarosJobDependency> getRequiredJobs(); public abstract Date getStartDate(); public abstract List<KlarosJob> getSubJobs(); public abstract Integer getSuccessRate(); public abstract String getSummary(); public abstract KlarosSUTImplementation getSut(); public abstract KlarosTestCase getTestCase(); public abstract Set<KlarosTestRun> getTestRuns(); public abstract KlarosTestSuite getTestSuite(); public abstract List<KlarosJobUpdateAction> getUpdateAction(); public abstract List<KlarosJobTimeBlock> getWork(); }



B.1.1.11.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this job.

Parameters

return

the attachments

B.1.1.11.3. getDependentJobs()

public abstract Set<KlarosJobDependency> getDependentJobs();

Gets the jobs depending on this jobs state.

Parameters

return

the dependent jobs

B.1.1.11.4. getRequiredJobs()

public abstract Set<KlarosJobDependency> getRequiredJobs();

Gets the jobs on which states this job is depending on.

Parameters

return

the required jobs

B.1.1.11.5. getTestRuns()

public abstract Set<KlarosTestRun> getTestRuns();

Gets the test runs executed with this job.

Parameters

return

the test runs

Gets the update action list.

Parameters

return

the update action

B.1.1.11.7. getWork()

public abstract List<KlarosJobTimeBlock> getWork();

Gets the list of work items done on this job.

Parameters

return

the work items

B.1.1.12. IKlarosJobDependency

This interface provides access to the data of a job.

B.1.1.12.1. Synopsis

public interface IKlarosJobDependency extends IKlarosPersistentObject {

```
// Public Methods
public abstract Set<JobStatus> getJobStates();
public abstract UUID getKey();
public abstract Integer getProgress();
public abstract KlarosJob getRequiredBy();
public abstract KlarosJob getRequires();
public abstract Integer getSuccessRate();
public abstract Set<Verdict> getVerdicts();
}
```



B.1.1.12.2. getJobStates()

public abstract Set<JobStatus> getJobStates();

Gets the job states.

Parameters

return

the job states

B.1.1.12.3. getKey()

public abstract UUID getKey();

Get the internal key of the object. This key is globally unique.

Parameters

The internal key.

Gets the job this dependency is required by.

Parameters

return

return

the job this dependency is required by

B.1.1.12.5. getRequires()

public abstract KlarosJob getRequires();

Returns the job this dependency requires for evaluation.

Parameters

return

the job this dependency requires

B.1.1.12.6. getVerdicts()

public abstract Set<Verdict> getVerdicts();

Gets the verdicts.

Parameters

return

the verdicts

B.1.1.13. IKlarosJobTimeBlock

The job time block defines a certain amount of time the job has been worked on.

B.1.1.13.1. Synopsis

```
public interface IKlarosJobTimeBlock
  extends IKlarosLabeledObject<KlarosJobTimeBlock> {
```

```
// Public Methods
public abstract String getDescription();
public abstract Date getDoneAt();
public abstract String getDuration();
public abstract Long getDurationInMilliseconds();
public abstract String getEstimatedTimeLeft();
public abstract Long getEstimatedTimeLeftInMilliseconds();
public abstract KlarosJob getJob();
public abstract KlarosTestRun getTestRun();
}
```



B.1.1.13.2. getDescription()

public abstract String getDescription();

Gets the work description.

Parameters

return

the work description

B.1.1.13.3. getDoneAt()

public abstract Date getDoneAt();

Gets the date this work was done at.

Parameters

return

the start date

B.1.1.13.4. getDuration()

public abstract String getDuration();

Gets the work duration as a String.

Parameters

return

the duration

B.1.1.13.5. getDurationInMilliseconds()

public abstract Long getDurationInMilliseconds();

Gets the work duration in milliseconds.

Parameters

	return	the work duration in hours
B.1.1	.13.6. getEstimatedTimeLeft() public abstract String getEst	imatedTimeLeft();
	Gets the estimated time left for thi	s job as a String.
		Parameters
	return	the estimated time left
B.1.1	.13.7. getEstimatedTimeLeftInM public abstract Long getEstim	illiseconds() atedTimeLeftInMilliseconds();
	Gets the estimated time left for thi	s job in milliseconds.
		Parameters
	return	the estimated time left in hours
B.1.1	.13.8.getJob() public abstract KlarosJob get.	Job();
	Gets the job this job time block be	longs to.
		Parameters
	return	the job
B.1.1	.13.9.getTestRun() public abstract KlarosTestRun	<pre>getTestRun();</pre>
Gets the test run this job time block belongs to.		k belongs to.
		Parameters
	return	the test run
B.1.1	.14. IKlarosJobUpdateAction	
	This interface provides access to t	he job update action.
B.1.1	.14.1. Synopsis	

public interface IKlarosJobUpdateAction
 extends IKlarosLabeledObject<KlarosJobUpdateAction> {

// Public Methods



```
// Public Methods
public abstract Iterable<T> asIterable();
public abstract Date getCreated();
public abstract KlarosUser getCreator();
public abstract UUID getKey();
public abstract KlarosUser getLastEditor();
public abstract Date getLastUpdated();
public abstract String getName();
public abstract boolean isEnabled();
```





B.1.1.15.2. aslterable()

public abstract Iterable<T> asIterable();

Return this object as an iterable containing just this object.

Parameters

return

the iterable container

B.1.1.15.3. getKey()

public abstract UUID getKey();

Get the internal key of the object. This key is globally unique.

Parameters

return

The internal key.

B.1.1.15.4. getName()
 public abstract String getName();

Get the name of the object. This matches the id field visible in the UI.

Parameters

return

The name of the object.

B.1.1.15.5. isEnabled()

public abstract boolean isEnabled();

Returns the value of the enabled flag of this object.

Parameters

return

true if the entity is enabled, else false

B.1.1.16. IKlarosNamedEntity

This interface provides access to data of a properties owner.

B.1.1.16.1. Synopsis

}

public interface IKlarosNamedEntity<T> extends IKlarosLabeledObject<T> {

```
// Public Methods
public abstract KlarosUser getAssignee();
public abstract List<IKlarosProperty> getProperties();
public abstract String getPropertyValue(String propertyName);
public abstract boolean isDefinedProperty(String propertyName);
```



B.1.1.16.2. getProperties()

public abstract List<IKlarosProperty> getProperties();

Gets the list of properties.

Parameters

	return	the properties
B.1.	1.16.3. getPropertyValue(String) public abstract String getPro	opertyValue(String propertyName);
	Gets the property value for the give	/en property name.
		Parameters
	propertyName	the property name
	return	the property value or null if not name is not present
B.1.	1.16.4. isDefinedProperty(String) public abstract boolean isDef	<pre>finedProperty(String propertyName);</pre>
	Check if a property identified by g	iven name is a defined property.
		Parameters
	propertyName	The name of the property to check.
	return	true if the identified property is a defined property, false else.

B.1.1.17. IKlarosPersistentObject

Base interface.

B.1.1.17.1. Synopsis

public interface IKlarosPersistentObject {
}

«interface» IKlarosPersistentObject

B.1.1.18. IKlarosProperty

A user defined property.

B.1.1.18.1. Synopsis

public interface IKlarosProperty extends IKlarosPersistentObject {

```
// Public Methods
public abstract String getName();
public abstract String getValue();
}
```



B.1.1.18.2. getName()

public abstract String getName();

Gets the property name.

Parameters

return

the property name

Gets the property value.

Parameters

return

the property value

B.1.1.19. IKlarosRepositoryEntity

This interface provides access to binary content.

B.1.1.19.1. Synopsis

public interface IKlarosRepositoryEntity<T> extends IKlarosLabeledObject<T> {

// Public Methods
public abstract String getMimeType();
public abstract String getUuid();
public abstract String getVersion();

}



B.1.1.19.2. getMimeType()

public abstract String getMimeType();

Gets the mime type of the attachment.

Parameters

return

the mime type

B.1.1.19.3. getUuid()

public abstract String getUuid();

Get the unique id of this attachment. This may be resolved by calling KlarosContext.getAttachmentURL() to an URL pointing to the attachment data.

Parameters

return

The uuid of this attachment.

B.1.1.19.4. getVersion()

public abstract String getVersion();

Gets the version of this attachment.

Parameters

return

the version

B.1.1.20. IKlarosRequirement

This interface provides access to data of a requirement.

B.1.1.20.1. Synopsis

}

```
public interface IKlarosRequirement<T,S extends Revision<?>>
    extends IKlarosRevision<T, S> {
```

```
// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract Set<KlarosCategoryNode> getCategories();
public abstract KlarosConfiguration getConfiguration();
public abstract Set<KlarosTestCase> getCoveringTestCases();
public abstract String getDescription();
public abstract Set<String> getExternalNames();
public abstract Set<KlarosIteration> getIterations();
public abstract Set<KlarosIteration> getIterations();
public abstract String getShortname();
public abstract String getState();
public abstract String getState();
public abstract String getSummary();
```



B.1.1.20.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this requirement.

		Parameters
	return	the attachments
B.1.1	.20.3. getCategories() public abstract Set <klarosca< td=""><td>tegoryNode> getCategories();</td></klarosca<>	tegoryNode> getCategories();
	Gets the categories this object be tree.	longs to. Each category node will belong to a different category
		Parameters
	return	the category nodes
B.1.1	.20.4. getConfiguration() public abstract KlarosConfigu	uration getConfiguration();
	Get configuration.	
		Parameters
	return	The related configuration.
B.1.1	.20.5. getCoveringTestCases() public abstract Set <klarostes< td=""><td><pre>stCase> getCoveringTestCases();</pre></td></klarostes<>	<pre>stCase> getCoveringTestCases();</pre>
	Get test cases covering the requir	rement.
		Parameters
	return	Set of test cases which cover this test requirement.
B.1.1	.20.6. getDescription() public abstract String getDes	<pre>scription();</pre>
	Get the description.	
		Parameters
	return	the string
B.1.1.20.7. getExternalNames() public abstract Set <string> getExternalNames();</string>		
	Gets the external requirement na names are used to match importe	mes this requirement is associated with. External requirement ed requirements to existing requirements.
		Parameters
	return	the external names
B.1.1.20.8. getExternalStatus() public abstract String getExternalStatus();		
	Gets the external status.	
		Parameters
	return	the external status

B.1.1.20.9. get public	<pre>lterations() abstract Set<klarositeration> getIterations();</klarositeration></pre>
Get the it	erations this requirement is assigned to.
	Parameters
return	Set of iterations this requirement is assigned to.
B.1.1.20.10. ge public	etPriority() abstract RequirementPriority getPriority();
Gets the	priority.
	Parameters
return	the priority
B.1.1.20.11. ge public	etShortname() abstract String getShortname();
Gets the	shortname.
	Parameters
return	the shortname
B.1.1.20.12. ge public	etState() abstract String getState();
Gets the	state.
	Parameters
return	the state
B.1.1.20.13. ge public	etSummary() abstract String getSummary();
Gets the	summary.
	Parameters
return	the summary
B.1.1.21. IKlar	osRequirementsManagement
The inter	ace for accessing requirements management systems.
B.1.1.21.1. Syr	nopsis
public in extends	iterface IKlarosRequirementsManagement IKlarosExternalServer <klarosrequirementsmanagement> {</klarosrequirementsmanagement>
// Publ public	ic Methods abstract String getProject();

}



B.1.1.21.2. getProject()

public abstract String getProject();

Get the optional project of this system.

Parameters

return

The project

B.1.1.22. IKlarosResult

This interface provides access to data of generic result.

B.1.1.22.1. Synopsis

public interface IKlarosResult<T> extends IKlarosNamedEntity<T> {

```
// Public Methods
public abstract KlarosConfiguration getConfiguration();
public abstract String getDescription();
public abstract long getExecutionTime();
public abstract KlarosJob getJob();
public abstract String getSummary();
public abstract KlarosTestRun getTestRun();
public abstract String getVerdict();
```

```
public abstract String getVerdict(Locale locale);
public abstract boolean isError();
public abstract boolean isFailure();
public abstract boolean isInconclusive();
public abstract boolean isPassed();
public abstract boolean isSkipped();
}
```



B.1.1.22.2. getConfiguration()

public abstract KlarosConfiguration getConfiguration();

Get configuration.

Parameters

return

The related configuration

B.1.1.22.3. getDescription()

public abstract String getDescription();

Get the test result description. This is usually set for failed/error status results.

Parameters

return

The test result description.

Get the test execution time in ms.

Parameters

return

The test execution time.

B.1.1.22.5. getJob()

public abstract KlarosJob getJob();

Gets the job this result belongs if one exists.

Parameters

return

the associated job

Get the test result summary. This is usually set for failed/error status results.

Parameters

return

The test result summary.

B.1.1.22.7. getTestRun()

public abstract KlarosTestRun getTestRun();

Get the associated test run.

Parameters

return

The test run that created this result.

B.1.1.22.8. getVerdict()

public abstract String getVerdict();

Returns the Test Case Result verdict as String. P = Passed U = Unknown S = Skipped E = Error F = Failed

Parameters

return

The Test Case Result verdict as String.
B.1.1.22.9. getVerdict(Locale)

public abstract String getVerdict(Locale locale);

Returns the Test Case Result verdict as String with localization. P = Passed U = Unknown S = Skipped E = Error F = Failed

Parameters

locale set location Information

return The Test Case Result verdict as String.

B.1.1.22.10. isError()

public abstract boolean isError();

Check if this is an error result. It is assumed, that error results have a property 'type' with the value 'E' or 'error'.

Parameters

return

true if this results represents an error.

B.1.1.22.11. isFailure()

public abstract boolean isFailure();

Check if this is a failure result. It is assumed, that failure results have a property 'type' with the value 'F' or 'failure'.

Parameters

return

true if this results represents a failure.

B.1.1.22.12. isInconclusive()

public abstract boolean isInconclusive();

Check if this is an inconclusive result. It is assumed, that inconclusive results have a property 'type' with the value 'l' or 'inconclusive'.

Parameters

return

true if this results represents an inconclusive result.

B.1.1.22.13. isPassed()

public abstract boolean isPassed();

Check if this is a result of a passed test case. It is assumed, that passed results have a property 'type' with the value 'P' or 'error'.

Parameters

return

true if this results represents an error.

B.1.1.22.14. isSkipped()

public abstract boolean isSkipped();

Check if this is a skipped result. It is assumed, that skipped results have a property 'type' with the value 'S' or 'skipped'.

```
Parameters
```

return

true if this results represents a skipped result.

```
B.1.1.23. IKlarosRevision
```

This interface provides access to a revisionable Klaros object.

```
B.1.1.23.1. Synopsis
```

```
public interface IKlarosRevision<T,S extends Revision<?>>
    extends IKlarosNamedEntity<T> {
```

```
// Public Methods
public abstract KlarosRevision<T, S> getPredecessor();
public abstract String getRevisionComment();
public abstract String getRevisionId();
public abstract KlarosRevision<T, S> getRoot();
public abstract KlarosRevision<T, S> getSuccessor();
}
```



B.1.1.23.2. getPredecessor() public abstract KlarosRevis:	ion <t, s=""> getPredecessor();</t,>
Get the predecessor of the revis	ion.
	Parameters
return	The revision object that is the predecessor of this revision.
B.1.1.23.3. getRevisionComment() public abstract String getRe	<pre>evisionComment();</pre>
Get comment.	
	Parameters
return	The comment of the revision.
B.1.1.23.4. getRevisionId() public abstract String getRe	evisionId();
Get the revision id.	
	Parameters
return	The revision id.
B.1.1.23.5. getRoot() public abstract KlarosRevis:	ion <t, s=""> getRoot();</t,>
Get the root of the revision hiera	rchy.
	Parameters
return	The root revision object.
B.1.1.23.6. getSuccessor() public abstract KlarosRevis:	ion <t, s=""> getSuccessor();</t,>
Get the successor of the revisior	٦.
	Parameters
return	The revision object that is the successor of this revision.
B.1.1.24. IKlarosSUTImplementation	٦
This interface provides access to	o data of a system under test version.
B.1.1.24.1. Synopsis	
public interface IKlarosSUTIm extends IKlarosNamedEntity<	plementation <larossutimplementation> {</larossutimplementation>

```
// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract Set<KlarosCategoryNode> getCategories();
public abstract KlarosConfiguration getConfiguration();
public abstract Set<KlarosIssue> getIssues();
public abstract Set<KlarosIteration> getIterations();
public abstract String getProductversion();
public abstract Set<KlarosTestRun> getTestRuns();
}
```



B.1.1.24.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this system under test.

Parameters

return

the attachments

B.1.1.24.3. getCategories()

public abstract Set<KlarosCategoryNode> getCategories();

Gets the categories this system under test belongs to. Each category node will belong to a different category tree.

Parameters

return

the category nodes

B.1.1.24.4. getConfiguration()

public abstract KlarosConfiguration getConfiguration();

Get configuration.

Parameters

return

The related configuration.

B.1.1.24.5. getIssues()

public abstract Set<KlarosIssue> getIssues();

Gets the issues related to this system under test.

Parameters

return

the issues

B.1.1.24.6. getIterations()

public abstract Set<KlarosIteration> getIterations();

Get the iterations this system under test is assigned to.

Parameters

return

Set of iterations this system under test is assigned to.

B.1.1.24.7. getProductversion()

public abstract String getProductversion();

Get product version.

Parameters

return

The version id of the system under test.

B.1.1.24.8. getTestRuns()

public abstract Set<KlarosTestRun> getTestRuns();

Get the test runs performed with this system under test.

Parameters

return

Set of test runs performed for this system under test.

B.1.1.25. IKlarosTestCase

This class provides access to data of a test case.

B.1.1.25.1. Synopsis

public interface IKlarosTestCase extends IKlarosRevision<KlarosTestCase, TestCase> {

```
// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract Set<KlarosCategoryNode> getCategories();
public abstract KlarosConfiguration getConfiguration();
public abstract Set<KlarosRequirement> getCovers();
public abstract String getDependency();
public abstract String getDescription();
public abstract TestDesignTechnique getDesignTechnique();
public abstract List<KlarosIssue> getDetectedIssues();
public abstract String getDocbase();
public abstract Long getEstimatedDuration();
public abstract TestEvaluationMethod getEvaluation();
public abstract TestExecutionMethod getExecution();
public abstract String getExpectedResult();
public abstract Set<String> getExternalNames();
public abstract List<KlarosJob> getJobs();
public abstract TestLevel getLevel();
public abstract String getNote();
public abstract String getPostcondition();
public abstract String getPrecondition();
public abstract TestPriority getPriority();
public abstract Set<KlarosTestCaseResult> getResults();
public abstract String getShortname();
public abstract String getState();
public abstract String getTeam();
public abstract List<KlarosTestCaseStep> getTestCaseSteps();
public abstract TestAreatopic getTestType();
public abstract String getTraceability();
public abstract TestVariety getVariety();
```

}



B.1.1.25.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this test case.

Parameters

return

the attachments

B.1.1.25.3. getCategories()

public abstract Set<KlarosCategoryNode> getCategories();

Gets the categories this object belongs to. Each category node will belong to a different category tree.

Parameters

return

the category nodes

B.1.1.25.4. getConfiguration()

public abstract KlarosConfiguration getConfiguration();

Get the project configuration this test case revision belongs to.

Parameters

return

The related configuration.

B.1.1.25.5. getCovers()

public abstract Set<KlarosRequirement> getCovers();

Get covered requirements.

Parameters

return

Set of requirements which are covered by this test case.

The dependency of this test case.

Parameters

return

The dependency.

The description of this test case.

Parameters

return

The description.

B.1.1.25.8. getDesignTechnique()

public abstract TestDesignTechnique getDesignTechnique();

The design technique of this test case.

Parameters

return

The type.

Get detected issues.

Parameters

return

List of issues which have been detected by this test case.

The docbase of this test case.

Parameters

return

The docbase.

The estimated duration of this test case.

Parameters

return

The estimatedDuration.

The evaluation of this test case.

Parameters

return

The evaluation.

B.1.1.25.13. getExecution()

public abstract TestExecutionMethod getExecution();

The execution method of this test case.

Parameters

return

The execution method.

B.1.1.25.14. getExpectedResult()

public abstract String getExpectedResult();

The expected result of this test case.

Parameters

return

The expected result.

B.1.1.25.15. getExternalNames()

public abstract Set<String> getExternalNames();

Gets the external test case names this test case is associated with. External test case names are used to match imported test case result to existing test cases.

Parameters

return

the external names

B.1.1.25.16. getJobs()

public abstract List<KlarosJob> getJobs();

Gets the jobs associated with this test case.

Parameters

return

the jobs

The level of this test case.

Parameters

return

The level.

B.1.1.25.18. getNote()
 public abstract String getNote();

The note of this test case.

Parameters

return

The note.

The postcondition of this test case.

Parameters

return

B.1.1.25.20. getPrecondition()

public abstract String getPrecondition();

The precondition of this test case.

Parameters

return

The precondition.

The priority of this test case.

Parameters

return

The priority.

B.1.1.25.22. getResults()

public abstract Set<KlarosTestCaseResult> getResults();

Get test case results.

Parameters

return

Set of results of executions of this test case.

The short name (title) of this test case.

Parameters

return

The short name.

The type of this test case.

Parameters

return

The type.

The team.

B.1.1.25.25.getTeam()
 public abstract String getTeam();

The team responsible for this test case.

Parameters

return

Get test case steps.

Parameters

return

list of steps of this test case.

B.1.1.25.27. getTestType()
 public abstract TestAreatopic getTestType();

The test type of this test case.

Parameters

return

The test type.

The traceability of this test case.

Parameters

return

The traceability.

The variety of this test case.

Parameters

return

The variety.

B.1.1.26. IKlarosTestCaseResult

This interface provides access to data of a test case result.

B.1.1.26.1. Synopsis

}

public interface IKlarosTestCaseResult extends IKlarosResult<KlarosTestCaseResult> {

// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract List<KlarosTestCaseStepResult> getStepResults();
public abstract KlarosTestCase getTestCase();
public abstract Integer getTestSuitePosition();
public abstract KlarosTestSuiteResult getTestSuiteResult();
public abstract boolean isPending();



B.1.1.26.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this result.

Parameters

return

the attachments

B.1.1.26.3. getStepResults()

public abstract List<KlarosTestCaseStepResult> getStepResults();

Get the associated step results.

Parameters

return

List of test case step results.

B.1.1.26.4. getTestCase()

public abstract KlarosTestCase getTestCase();

Get the associated test case.

Parameters

return

The test case that has been executed to get this result.

B.1.1.26.5. getTestSuitePosition()

public abstract Integer getTestSuitePosition();

Return the position in the test suite result this test case result belongs to or null if this test case has not been executed by executing a test suite.

Parameters

return the position of the test case in the test suite, or null

B.1.1.26.6. getTestSuiteResult()

public abstract KlarosTestSuiteResult getTestSuiteResult();

Return the test suite result this test case result belongs to or null if this test case has not been executed by executing a test suite.

Parameters

return

the test suite result, or null

B.1.1.26.7. isPending()

public abstract boolean isPending();

Checks if the result is pending. This means that it has not yet been completely executed.

Parameters

return

true, if the result is pending

B.1.1.27. IKlarosTestCaseStep

This interface provides access to a test case step.

B.1.1.27.1. Synopsis

}

public interface IKlarosTestCaseStep<T> extends IKlarosTestStepContainer<T> {

// Public Methods
public abstract Set<KlarosAttachment> getAttachments();



B.1.1.27.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this test case.

Parameters

return

the attachments

B.1.1.28. IKlarosTestCaseStepResult

This interface provides access to data of a test case step result.

B.1.1.28.1. Synopsis

public interface IKlarosTestCaseStepResult
 extends IKlarosResult<KlarosTestCaseStepResult> {

```
// Public Methods
public abstract String getAction();
public abstract Set<KlarosAttachment> getAttachments();
public abstract String getExpectedResult();
public abstract String getPostcondition();
public abstract KlarosTestCase getTestCase();
public abstract KlarosTestCaseResult getTestCaseResult();
}
```



B.1.1.28.2. getActio	on()
Get the action	
	Parameters
return	The action of this test case step
B.I.I.28.3. getAtta public abst	chments() ract Set <klarosattachment> getAttachments();</klarosattachment>
Gets the attac	hments associated with this test case.
	Parameters
return	the attachments
B.1.1.28.4. getExpe public abst	<pre>ectedResult() ract String getExpectedResult();</pre>
Get the expec	ted result.
	Parameters
return	The expected result of this test case step.
B.1.1.28.5. getPost public abst	condition() ract String getPostcondition();
Get the postco	ondition.
	Parameters
return	The postcondition of this test case step.
B.1.1.28.6. getPrec public abst	ondition() ract String getPrecondition();
Get the precor	ndition.
	Parameters
return	The precondition of this test case step.
B.1.1.28.7.getTest public abst	Case() ract KlarosTestCase getTestCase();
Get test case.	
	Parameters
return	The test case that has been executed to get this result.
B.1.1.28.8. getTest public abst	CaseResult() ract KlarosTestCaseResult getTestCaseResult();
Get test case	result.

Parameters

return

The test case result that has been executed to get this result.

B.1.1.29. IKlarosTestEnvironment

This interface provides access to data of a test environment.

B.1.1.29.1. Synopsis

```
public interface IKlarosTestEnvironment
    extends IKlarosNamedEntity<KlarosTestEnvironment> {
```

```
// Public Methods
public abstract Set<KlarosAttachment> getAttachments();
public abstract Set<KlarosCategoryNode> getCategories();
public abstract String getDescription();
public abstract Set<KlarosIteration> getIterations();
public abstract Set<KlarosTestRun> getTestRuns();
}
```



B.1.1.29.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this test environment.

Parameters

return

the attachments

B.1.1.29.3. getCategories()

public abstract Set<KlarosCategoryNode> getCategories();

Gets the categories this test environment belongs to. Each category node will belong to a different category tree.

Parameters

return

the category nodes

B.1.1.29.4. getIterations()

public abstract Set<KlarosIteration> getIterations();

Get the iterations this test environment is assigned to.

Parameters

return

Set of iterations this test environment is assigned to.

B.1.1.29.5. getTestRuns()

public abstract Set<KlarosTestRun> getTestRuns();

Gets the test runs associated with this test environment.

Parameters

return

Collection of test runs executed in the environment.

B.1.1.30. IKlarosTestExecutable

This class provides access to a test executable. This may wrap a test case or a test suite.

B.1.1.30.1. Synopsis

public interface IKlarosTestExecutable extends IKlarosPersistentObject {
}



B.1.1.31. IKlarosTestRun

This interface provides access to data of a test run.

B.1.1.31.1. Synopsis

}

public interface IKlarosTestRun extends IKlarosNamedEntity<KlarosTestRun> {

```
// Public Methods
public abstract KlarosConfiguration getConfiguration();
public abstract KlarosTestEnvironment getEnv();
public abstract KlarosIteration getIteration();
public abstract KlarosJob getJob();
public abstract int getNumberErrors();
public abstract int getNumberFailures();
public abstract int getNumberInconclusive();
public abstract int getNumberPassed();
public abstract int getNumberSkipped();
public abstract String getRelatedSummary();
public abstract Set<KlarosTestCaseResult> getResults();
public abstract String getRunId();
public abstract KlarosSUTImplementation getSut();
public abstract KlarosTestSuite getTestSuite();
public abstract Date getTimestamp();
public abstract Set<KlarosJobTimeBlock> getWork();
public abstract boolean isPending();
```



B.1.1.31.2. getConfiguration()

public abstract KlarosConfiguration getConfiguration();

Get configuration.

Parameters

return

The related configuration

B.1.1.31.3. getEnv()

public abstract KlarosTestEnvironment getEnv();

Get test environment.

Parameters

return

The test environment in which the test cases have been executed.

B.1.1.31.4. getIteration()

public abstract KlarosIteration getIteration();

Get the iteration in which the test run has been created, if available.

Parameters

return

The iteration in which the test run has been created

B.1.1.31.5. getJob()

public abstract KlarosJob getJob();

Get the job that initiated the test run, if available.

Parameters

return

The job that initiated the test run

> Get number of test cases with errors of this test run. It is assumed, that error results have a property 'type' with the value 'E'.

> > Parameters

return

The number of error test cases

B.1.1.31.7. getNumberFailures()

public abstract int getNumberFailures();

Get number of failed test cases of this test run. It is assumed, that failed results have a property 'type' with the value 'F'.

Parameters

return

The number of failed test cases

B.1.1.31.8. getNumberInconclusive()

public abstract int getNumberInconclusive();

Get number of inconclusive test cases of this test run. It is assumed, that inconclusive results have a property 'type' with the value 'l'.

Parameters

The number of skipped test cases

B.1.1.31.9. getNumberPassed()

public abstract int getNumberPassed();

Get number of passed test cases of this test run. It is assumed, that passed results have a property 'testCasePassed' with value 'true'.

Parameters

return

return

The number of passed test cases

B.1.1.31.10. getNumberSkipped()

public abstract int getNumberSkipped();

Get number of skipped test cases of this test run. It is assumed, that skipped results have a property 'type' with the value 'S'.

Parameters

return

The number of skipped test cases

B.1.1.31.11. getRelatedSummary()

public abstract String getRelatedSummary();

Gets the summary of the job, test suite or test case related to this test run.

Parameters

return

the related summary

B.1.1.31.12. getResults()

public abstract Set<KlarosTestCaseResult> getResults();

Get results.

Parameters

return

Set of results of test case executions.

Get id of test run.

Parameters

return

The id of the test run

B.1.1.31.14. getSut()

public abstract KlarosSUTImplementation getSut();

Get the tested system v	version.
	Parameters
return	The SUT version which has been tested
B.1.1.31.15. getTestSuite() public abstract Klan	rosTestSuite getTestSuite();
Get the test suite execu	ited with this test run, if available.
	Parameters
return	The test suite executed with this test run
B.1.1.31.16. getTimestamp public abstract Date	() e getTimestamp();
Get timestamp.	
	Parameters
return	The time the test run has been executed as Date object
B.1.1.31.17.getWork() public abstract Set	<klarosjobtimeblock> getWork();</klarosjobtimeblock>
Get the job time blocks	of this test run.
	Parameters
return	Set of job time blocks.
B.1.1.31.18. isPending() public abstract bool	lean isPending();
Checks if the test run is	pending. This means that it has not yet been completely executed.
	Parameters
return	true, if the test run is pending
B.1.1.32. IKlarosTestSegme	ent
This interface provides	access to a test segment.
B.1.1.32.1. Synopsis	
public interface IKla extends IKlarosRevis	rosTestSegment <t,s extends="" revision<s="">> sion<t, s=""> {</t,></t,s>
// Public Methods	

public abstract String getDescription();
public abstract String getExpectedResult();

```
public abstract String getNote();
public abstract String getPostcondition();
public abstract String getPrecondition();
public abstract String getShortname();
public abstract List<KlarosTestCaseStep> getSteps();
}
```



B.1.1.32.2. getDescription()

public abstract String getDescription();

Get the description.

Parameters

		Parameters
	return	The description of this test segment.
B.1.1	.32.3. getExpectedResult() public abstract String getExp	<pre>pectedResult();</pre>
	Get the expected result.	
		Parameters
	return	The expected result of this test case step.
B.1.1	.32.4. getNote() public abstract String getNot	:e();
	Get the note.	
		Parameters
	return	The note of this test segment.
B.1.1	.32.5. getPostcondition() public abstract String getPos	stcondition();
	Get the postcondition.	
		Parameters
	return	The postcondition of this test case step.
B.1.1	.32.6. getPrecondition() public abstract String getPre	econdition();
	Get the precondition.	
		Parameters
	return	The precondition of this test case step.
B.1.1	.32.7. getShortname() public abstract String getSho	<pre>prtname();</pre>
	Get the shortname.	
		Parameters
	return	The shortname of this test case segment.
B.1.1	.32.8. getSteps() public abstract List <klaroste< td=""><td>estCaseStep> getSteps();</td></klaroste<>	estCaseStep> getSteps();
	Get the test case steps.	

Parameters

return

list of steps of this test segment.

```
B.1.1.33. IKlarosTestStepContainer
```

This interface provides access to a test step containers.

```
B.1.1.33.1. Synopsis
```

public interface IKlarosTestStepContainer<T> extends IKlarosNamedEntity<T> {

```
// Public Methods
public abstract String getAction();
public abstract String getExpectedResult();
public abstract String getPostcondition();
public abstract String getPrecondition();
}
```



B.1.1.33.2. getAction()

public abstract String getAction();

Get the action.		
	Parameters	
return	The action of this test case step.	
<pre>B.1.1.33.3. getExpectedResult() public abstract String getExpectedResult();</pre>		
Get the expected result		
	Parameters	
return	The expected result of this test case step.	
B.1.1.33.4. getPostcondition() public abstract String getPostcondition();		
Get the postcondition.		
	Parameters	
return	The postcondition of this test case step.	
B.1.1.33.5. getPrecondition() public abstract String getPrecondition();		
Get the precondition.		
	Parameters	
return	The precondition of this test case step.	
B.1.1.34. IKlarosTestSuite		
This interface provides	access to data of a test suite.	
B.1.1.34.1. Synopsis		
<pre>public interface IKla S> {</pre>	<pre>rosTestSuite<t,s extends="" revision<?="">> extends IKlarosRevision<t,< pre=""></t,<></t,s></pre>	
<pre>// Public Methods public abstract Set- public abstract Set- public abstract Klan public abstract Lis- public abstract Str: public abstract Klan public abstract Lis-</pre>	<pre>KlarosAttachment> getAttachments(); KlarosCategoryNode> getCategories(); rosConfiguration getConfiguration(); t<klarostestsuiteresult> getResults(); ing getShortname(); rosSUTImplementation getSut(); t<klarostestcase> getTestCases();</klarostestcase></klarostestsuiteresult></pre>	

public abstract Integer getTestSuiteResultCount();



B.1.1.34.2. getAttachments()

public abstract Set<KlarosAttachment> getAttachments();

Gets the attachments associated with this test suite.

Parameters

return

the attachments

B.1.1.34.3. getCategories()

public abstract Set<KlarosCategoryNode> getCategories();

	Gets the categories this object be tree.	longs to. Each category node will belong to a different category
		Parameters
	return	the category nodes
B.1.1	.34.4. getConfiguration() public abstract KlarosConfigu	uration getConfiguration();
	Get configuration.	
		Parameters
	return	The related configuration.
B.1.1	.34.5. getResults() public abstract List <klaroste< td=""><td>estSuiteResult> getResults();</td></klaroste<>	estSuiteResult> getResults();
	Get test suite results.	
		Parameters
	return	Set of results of executions of this test suite.
B.1.1	.34.6. getShortname() public abstract String getSho	prtname();
	Get the short name of the test sui	te.
		Parameters
	return	String of the description of the test suite.
B.1.1	.34.7. getSut() public abstract KlarosSUTImp	lementation getSut();
	Get the description of the test sui	te.
		Parameters
	return	String of the description of the test suite.
B.1.1	.34.8. getTestCases() public abstract List <klaroste< td=""><td>estCase> getTestCases();</td></klaroste<>	estCase> getTestCases();
	Get the executables of this test su	uite.
		Parameters
	return	Set of test cases of this test suite.
B.1.1	.34.9. getTestSuiteResultCount(public abstract Integer getTe	() estSuiteResultCount();
Return the number of test suite results in this test suite.		
Parameters		
	return	The number of test suite results in this test suite.

B.1.1.35. IKlarosTestSuiteResult

This interface provides access to data of a test suite result.

B.1.1.35.1. Synopsis

}

```
public interface IKlarosTestSuiteResult
  extends IKlarosResult<KlarosTestSuiteResult> {
```

```
// Public Methods
public abstract Set<KlarosTestCaseResult> getResults();
public abstract KlarosTestSuite getTestSuite();
public abstract boolean isPending();
```



B.1.1.35.2. getResults()

public abstract Set<KlarosTestCaseResult> getResults();

Gets the test case results for this test suite result. The list is sorted by the position in which the test cases appear in the test suite.

Parameters

return

the results

B.1.1.35.3. getTestSuite()

public abstract KlarosTestSuite getTestSuite();

Get the related test suite.

Parameters

return

The test suite that has been executed to get this result.

B.1.1.35.4. isPending()

public abstract boolean isPending();

Checks if the result is pending. This means that it has not yet been completely executed.

Parameters

return

true, if the result is pending

B.1.1.36. IKlarosUser

The user object.

B.1.1.36.1. Synopsis

public interface IKlarosUser extends IKlarosLabeledObject<KlarosUser> {

	// Public Methods
	<pre>public abstract String getEmail();</pre>
	<pre>public abstract UserRole getRole();</pre>
	<pre>public abstract String getUsername();</pre>
}	



B.1.1.36.2. getEmail()

public abstract String getEmail();

The email address of this user.

Parameters

return

the email address

B.1.1.36.3. getRole()

public abstract UserRole getRole();

The role name of this user.

Parameters

return

the role name

B.1.1.36.4. getUsername()

public abstract String getUsername();

The username of this user as used when logging in.

Parameters

return

the username

B.1.1.37. KlarosAttachment

This class provides access to binary attachment.

B.1.1.37.1. Synopsis

}

implements IKlarosAttachment {

```
// Public Methods
public long getSize();
```

Methods inherited from de.verit.klaros.core.model.KlarosRepositoryEntity: getMimeType, get-Name, getUuid, getVersion

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.38. KlarosCategoryNode

A generic category node.

B.1.1.38.1. Synopsis

public class KlarosCategoryNode extends KlarosWrapper<KlarosCategoryNode, CategoryNode>

```
implements IKlarosCategoryNode {
    // Public Constructors
    public KlarosCategoryNode();
    // Public Methods
    public CategoryTree getCategoryTree();
    public String getDescription();
    public KlarosCategoryNode getParent();
    public String getShortname();
    public List<KlarosCategoryNode> getSubCategories();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



Constructor.

B.1.1.39. KlarosCategoryTree

A category tree.

B.1.1.39.1. Synopsis

public class KlarosCategoryTree extends

KlarosNamedEntity<KlarosCategoryTree, CategoryTree>

```
implements IKlarosCategoryTree {
    // Public Constructors
    public KlarosCategoryTree();
    // Public Methods
    public String getDescription();
    public KlarosCategoryNode getRootNode();
    public String getShortname();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



Serialization constructor.
B.1.1.40. KlarosConfiguration

This class provides access to the information stored for project's configuration.

B.1.1.40.1. Synopsis

public class KlarosConfiguration extends

KlarosLabeledObject<KlarosConfiguration,

Configuration>

```
// Public Methods
 public int compareTo(KlarosConfiguration o);
 public boolean equals(Object o);
 public String getDescription();
 public Set<KlarosTestEnvironment> getEnvs();
 public Set<KlarosIssueManagement> getIssueManagementSystems();
 public Set<KlarosIssue> getIssues();
 public Set<KlarosCategoryTree> getIterationCategoryTrees();
 public Set<KlarosIteration> getIterations();
 public Set<KlarosJob> getJobs();
 public Set<KlarosCategoryTree> getRequirementCategoryTrees();
 public Set<KlarosRequirement> getRequirements();
 public KlarosRequirementsManagement getRequirementsManagementSystem();
 public Set<KlarosSUTImplementation> getSuts();
 public Set<KlarosCategoryTree> getSystemUnderTestCategoryTrees();
 public Set<KlarosCategoryTree> getTestCaseCategoryTrees();
 public Set<KlarosTestCase> getTestCases();
 public Set<KlarosCategoryTree> getTestEnvironmentCategoryTrees();
 public Set<KlarosTestRun> getTestRuns();
 public Set<KlarosCategoryTree> getTestSegmentCategoryTrees();
 public Set<KlarosCategoryTree> getTestSuiteCategoryTrees();
 public Set<KlarosTestSuite> getTestSuites();
 public int hashCode();
 public Boolean isSecured();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.41. KlarosEnumValue

A Custom property enumeration value.

```
B.1.1.41.1. Synopsis
```

```
public final class KlarosEnumValue extends KlarosWrapper<KlarosEnumValue, EnumValue>
implements IKlarosEnumValue {
```

// Public Methods
public String getValue();
}

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

 $\label{eq:methods} \textbf{Methods inherited from java.lang.Object} : \texttt{getClass,notify,notifyAll,toString,wait}$



B.1.1.42. KlarosExternalServer

This class encapsulates the (dynamic) properties of a klaros object.

B.1.1.42.1. Synopsis

}

public abstract class KlarosExternalServer<T extends KlarosExternalServer<T, S>,S
 extends ExternalServer> extends

```
KlarosLabeledObject<T, S>
```

implements IKlarosExternalServer<T> {

// Public Methods
public String getDescription();
public String getType();
public String getUrl();

Direct known subclasses: de.verit.klaros.core.model.KlarosIssueManagement , de.verit.klaros.core.model.KlarosRequirementsManagement

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.43. KlarosIssue

This class provides access to the information stored for detected issues.

B.1.1.43.1. Synopsis

```
public final class KlarosIssue extends
                                 KlarosNamedEntity<KlarosIssue, Issue>
  implements IKlarosIssue {
 // Public Methods
 public KlarosSUTImplementation getAcceptedIn();
 public Date getCreationDate();
 public String getDescription();
 public String getExternalId();
 public KlarosIssueManagement getIssueManagement();
 public Date getLastSynched();
 public String getOwner();
 public String getPriority();
 public String getRemoteBrowseUrl();
 public String getReporter();
 public String getResolution();
 public String getState();
```

```
public String getSubject();
public Set<KlarosSUTImplementation> getSystemsUnderTest();
public Set<KlarosTestCase> getTestCases();
public boolean isResolved();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.44. KlarosIssueManagement

An issue management system.

B.1.1.44.1. Synopsis

public class KlarosIssueManagement extends

KlarosExternalServer<KlarosIssueManagement,</pre>

IssueManagement>

implements IKlarosIssueManagement {

// Public Methods

```
public Set<KlarosIssue> getIssues();
public String getProject();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosExternalServer: getDescription , getType , getUrl

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.45. KlarosIteration

An iteration in a project.

B.1.1.45.1. Synopsis

```
public Set<KlarosCategoryNode> getCategories();
public StateDef getCurrentState();
public String getDescription();
public Date getDueDate();
public Set<KlarosTestEnvironment> getEnvs();
public Set<KlarosJob> getJobs();
public String getShortname();
public Date getStartDate();
public String getSuccessCriteria();
public Set<KlarosSUTImplementation> getSuts();
public Set<KlarosTestRun> getTestRuns();
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.46. KlarosJob

This class provides access to the information stored for jobs.

B.1.1.46.1. Synopsis

public final class KlarosJob extends

KlarosNamedEntity<KlarosJob, Job>

// Public Methods public int compareTo(KlarosJob o); public boolean equals(Object o); public Set<KlarosAttachment> getAttachments(); public KlarosConfiguration getConfiguration(); public Set<KlarosJobDependency> getDependentJobs(); public String getDescription(); public Date getDueDate(); public KlarosTestEnvironment getEnv(); public String getEstimatedTime(); public Long getEstimatedTimeInMilliseconds(); public JobPriority getJobPriority(); public JobStatus getJobStatus(); public JobType getJobType(); public KlarosJob getParent(); public Integer getProgress(); public Set<KlarosJobDependency> getRequiredJobs(); public Date getStartDate(); public List<KlarosJob> getSubJobs(); public Integer getSuccessRate(); public String getSummary(); public KlarosSUTImplementation getSut(); public KlarosTestCase getTestCase(); public Set<KlarosTestRun> getTestRuns(); public KlarosTestSuite getTestSuite(); public List<KlarosJobUpdateAction> getUpdateAction(); public List<KlarosJobTimeBlock> getWork(); public int hashCode(); 3

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.47. KlarosJobDependency

This class provides access to the information stored for jobs.

```
B.1.1.47.1. Synopsis
```

```
public final class KlarosJobDependency extends KlarosWrapper<KlarosJobDependency,
    JobDependency>
    implements IKlarosJobDependency,
        Comparable<KlarosJobDependency> {
    // Public Methods
    public int compareTo(KlarosJobDependency o);
    public boolean equals(Object o);
    public Set<JobStatus> getJobStates();
    public UUID getKey();
```

```
public Integer getProgress();
public KlarosJob getRequiredBy();
public KlarosJob getRequires();
public Integer getSuccessRate();
public Set<Verdict> getVerdicts();
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.48. KlarosJobTimeBlock

The implementation of a job time block, measuring a unit of work done for a job.

B.1.1.48.1. Synopsis

```
public final class KlarosJobTimeBlock extends
                                 KlarosLabeledObject<KlarosJobTimeBlock,</pre>
 JobTimeBlock>
  implements IKlarosJobTimeBlock,
          Comparable<KlarosJobTimeBlock> {
 // Public Methods
 public int compareTo(KlarosJobTimeBlock o);
 public boolean equals(Object o);
 public String getDescription();
 public Date getDoneAt();
 public String getDuration();
 public Long getDurationInMilliseconds();
 public String getEstimatedTimeLeft();
 public Long getEstimatedTimeLeftInMilliseconds();
 public KlarosJob getJob();
 public KlarosTestRun getTestRun();
```

```
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.49. KlarosJobUpdateAction

The model of a job comment.

B.1.1.49.1. Synopsis

public final class KlarosJobUpdateAction extends

KlarosLabeledObject<KlarosJobUpdateAction,

JobUpdateAction>

```
implements IKlarosJobUpdateAction,
        Comparable<KlarosJobUpdateAction> {
        // Public Methods
        public int compareTo(KlarosJobUpdateAction o);
```

```
public int compareTo(KlarosJobUpdateAction o);
public boolean equals(Object o);
public String getChanges();
public String getDescription();
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.50. KlarosLabeledObject

This class encapsulates the (dynamic) properties of a klaros object.

B.1.1.50.1. Synopsis

```
public abstract class KlarosLabeledObject<T,S extends LabeledObject>
extends KlarosWrapper<T, S>
    implements IKlarosLabeledObject<T> {
    // Public Methods
    public Iterable<T> asIterable();
    public Date getCreated();
    public KlarosUser getCreator();
    public UUID getKey();
    public KlarosUser getLastEditor();
    public Date getLastUpdated();
    public String getName();
    public boolean isEnabled();
}
```

Direct known subclasses: de.verit.klaros.core.model.KlarosConfiguration , de.verit.klaros.core.model.KlarosExternalServer , de.verit.klaros.core.model.KlarosJobTimeBlock , de.verit.klaros.core.model.KlarosJobUpdateAction , de.verit.klaros.core.model.KlarosNamedEntity , de.verit.klaros.core.model.KlarosCore.model.KlarosCore.model.KlarosUser

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.51. KlarosNamedEntity

This class encapsulates the (dynamic) properties of a klaros object.

B.1.1.51.1. Synopsis

```
implements IKlarosNamedEntity<T> {
```

```
// Public Methods
public KlarosUser getAssignee();
public List<IKlarosProperty> getProperties();
public String getPropertyValue(String propertyName);
public boolean isDefinedProperty(String propertyName);
}
```

```
Direct known subclasses: de.verit.klaros.core.model.KlarosCategoryTree , de.verit.klaros.core.model.KlarosIssue , de.verit.klaros.core.model.KlarosIteration , de.verit.klaros.core.model.KlarosResult , de.verit.klaros.core.model.KlarosSUTImplementation , de.verit.klaros.core.model.Klaros.core.model.KlarosTestEnvironment , de.verit.klaros.core.model.KlarosTestRun , de.verit.klaros.core.model.KlarosTestStepContainer
```

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.52. KlarosProperty

The Klaros property class.

B.1.1.52.1. Synopsis

```
public final class KlarosProperty extends KlarosWrapper<KlarosProperty, Property>
implements IKlarosProperty {
```

```
// Public Methods
public String getName();
public String getValue();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.53. KlarosRepositoryEntity

This class encapsulates the (dynamic) properties of a klaros object.

B.1.1.53.1. Synopsis

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.54. KlarosRequirement

This class provides access to the information stored for requirements.

B.1.1.54.1. Synopsis

```
public final class KlarosRequirement extends
                                KlarosRevision<KlarosRequirement, Requirement>
  implements IKlarosRequirement<KlarosRequirement, Requirement> {
 // Public Methods
 public Set<KlarosAttachment> getAttachments();
 public Set<KlarosCategoryNode> getCategories();
 public KlarosConfiguration getConfiguration();
 public Set<KlarosTestCase> getCoveringTestCases();
 public String getDescription();
 public Set<String> getExternalNames();
 public String getExternalStatus();
 public Set<KlarosIteration> getIterations();
 public KlarosRequirement getPredecessor();
 public RequirementPriority getPriority();
 public KlarosRequirement getRoot();
 public String getShortname();
```

```
public String getState();
public KlarosRequirement getSuccessor();
public String getSummary();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosRevision: getRevisionComment , get-RevisionId

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



 The external status of this requirement.

Parameters

return

the external status

B.1.1.55. KlarosRequirementsManagement

A requirement management system.

B.1.1.55.1. Synopsis

public class KlarosRequirementsManagement extends

KlarosExternalServer<KlarosRequirementsManagement,

RequirementsManagement>

implements IKlarosRequirementsManagement {

```
// Public Methods
public String getProject();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosExternalServer: getDescription , getType , getUrl

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.56. KlarosResult

This class provides access to the information stored for generic results.

B.1.1.56.1. Synopsis

```
implements IKlarosResult<T> {
    // Public Methods
    public KlarosConfiguration getConfiguration();
    public String getDescription();
    public long getExecutionTime();
    public KlarosJob getJob();
    public String getSummary();
    public String getVerdict();
    public String getVerdict(Locale locale);
    public boolean isError();
    public boolean isFailure();
    public boolean isInconclusive();
    public boolean isPassed();
    public boolean isSkipped();
```

```
}
```

Direct known subclasses: de.verit.klaros.core.model.KlarosTestCaseResult , de.verit.klaros.core.model.KlarosTestCaseStepResult , de.verit.klaros.core.model.KlarosTestSuiteResult

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.57. KlarosRevision

This class encapsulates the revision related information of a klaros object.

B.1.1.57.1. Synopsis

implements IKlarosRevision<T, S> {

// Public Methods

```
public final String getRevisionComment();
public final String getRevisionId();
}
```

Direct known subclasses: de.verit.klaros.core.model.KlarosRequirement , de.verit.klaros.core.model.KlarosTestCase , de.verit.klaros.core.model.KlarosTestSegment , de.verit.klaros.core.model.KlarosTestSuite

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.58. KlarosSUTImplementation

This class provides access to the information stored for systems under tests (SUT).

B.1.1.58.1. Synopsis

}

public final class KlarosSUTImplementation extends

KlarosNamedEntity<KlarosSUTImplementation,

SystemUnderTest>

```
// Public Methods
public int compareTo(KlarosSUTImplementation o);
public boolean equals(Object o);
public Set<KlarosAttachment> getAttachments();
public Set<KlarosCategoryNode> getCategories();
public KlarosConfiguration getConfiguration();
public Set<KlarosIssue> getIssues();
public Set<KlarosIteration> getIterations();
public String getProductversion();
public Set<KlarosTestRun> getTestRuns();
public int hashCode();
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.59. KlarosTestCase

This class provides access to the information stored for a test case.

B.1.1.59.1. Synopsis

public final class KlarosTestCase extends

KlarosRevision<KlarosTestCase, TestCase>

// Public Methods public int compareTo(KlarosTestCase o); public boolean equals(Object o); public Set<KlarosAttachment> getAttachments(); public Set<KlarosCategoryNode> getCategories(); public KlarosConfiguration getConfiguration(); public Set<KlarosRequirement> getCovers(); public String getDependency(); public String getDescription(); public TestDesignTechnique getDesignTechnique(); public List<KlarosIssue> getDetectedIssues(); public String getDocbase(); public Long getEstimatedDuration(); public TestEvaluationMethod getEvaluation(); public TestExecutionMethod getExecution(); public String getExpectedResult(); public Set<String> getExternalNames(); public List<KlarosJob> getJobs(); public TestLevel getLevel(); public String getNote(); public String getPostcondition();

```
public String getPrecondition();
public KlarosTestCase getPredecessor();
public TestPriority getPriority();
public Set<KlarosTestCaseResult> getResults();
public KlarosTestCase getRoot();
public String getShortname();
public String getState();
public KlarosTestCase getSuccessor();
public KlarosTestCase getSuccessor();
public List<KlarosTestCaseStep> getTestCaseSteps();
public TestAreatopic getTestType();
public String getTraceability();
public TestVariety getVariety();
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosRevision: getRevisionComment , get-RevisionId

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.60. KlarosTestCaseResult

This class provides access to the information stored for test case results.

B.1.1.60.1. Synopsis

```
// Public Methods
public int compareTo(KlarosTestCaseResult o);
public boolean equals(Object o);
public Set<KlarosAttachment> getAttachments();
public List<KlarosTestCaseStepResult> getStepResults();
public KlarosTestCase getTestCase();
public KlarosTestRun getTestRun();
public Integer getTestSuitePosition();
public KlarosTestSuiteResult getTestSuiteResult();
public int hashCode();
public boolean isPending();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosResult: getConfiguration , get-Description , getExecutionTime , getJob , getSummary , getVerdict , isError , isFailure , isInconclusive , isPassed , isSkipped

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.61. KlarosTestCaseStep

This class provides access to the information stored for test case steps.

B.1.1.61.1. Synopsis

```
public final class KlarosTestCaseStep extends
```

KlarosTestStepContainer<KlarosTestCaseStep,</pre>

TestStep>

}

implements IKlarosTestCaseStep<KlarosTestCaseStep> {

```
// Public Methods
public String getAction();
public Set<KlarosAttachment> getAttachments();
public String getExpectedResult();
public String getPostcondition();
public String getPrecondition();
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.62. KlarosTestCaseStepResult

This class provides access to the information stored for test case step results.

B.1.1.62.1. Synopsis

```
public final class KlarosTestCaseStepResult extends
                                 KlarosResult<KlarosTestCaseStepResult,
 TestCaseStepResult>
  implements IKlarosTestCaseStepResult,
          Comparable<KlarosTestCaseStepResult> {
 // Public Methods
 public int compareTo(KlarosTestCaseStepResult o);
 public boolean equals(Object o);
 public String getAction();
 public Set<KlarosAttachment> getAttachments();
 public String getExpectedResult();
 public String getPostcondition();
 public String getPrecondition();
 public KlarosTestCase getTestCase();
 public KlarosTestCaseResult getTestCaseResult();
 public KlarosTestRun getTestRun();
 public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosResult: getConfiguration , get-Description , getExecutionTime , getJob , getSummary , getVerdict , isError , isFailure , isInconclusive , isPassed , isSkipped

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.63. KlarosTestEnvironment

This class provides access to the information stored for test environments.

B.1.1.63.1. Synopsis

// Public Methods

```
public int compareTo(KlarosTestEnvironment o);
public boolean equals(Object o);
public Set<KlarosAttachment> getAttachments();
public Set<KlarosCategoryNode> getCategories();
public String getDescription();
public Set<KlarosIteration> getIterations();
public Set<KlarosTestRun> getTestRuns();
public int hashCode();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.64. KlarosTestExecutable

This class provides access to the information stored for test cases.

B.1.1.64.1. Synopsis

```
public final class KlarosTestExecutable extends KlarosWrapper<KlarosTestExecutable,
TestExecutable>
```

```
implements IKlarosTestExecutable {
```

```
}
```

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.65. KlarosTestRun

This class provides access to the information stored for test runs.

B.1.1.65.1. Synopsis

```
public final class KlarosTestRun extends
                                 KlarosNamedEntity<KlarosTestRun, TestRun>
  implements IKlarosTestRun,
          Comparable<KlarosTestRun> {
 // Public Methods
 public int compareTo(KlarosTestRun o);
 public boolean equals(Object o);
 public KlarosConfiguration getConfiguration();
 public KlarosTestEnvironment getEnv();
 public KlarosIteration getIteration();
 public KlarosJob getJob();
 public int getNumberErrors();
 public int getNumberFailures();
 public int getNumberInconclusive();
 public int getNumberPassed();
 public int getNumberSkipped();
 public String getRelatedSummary();
 public Set<KlarosTestCaseResult> getResults();
 public String getRunId();
 public KlarosSUTImplementation getSut();
 public KlarosTestSuite getTestSuite();
 public Date getTimestamp();
 public Set<KlarosJobTimeBlock> getWork();
 public int hashCode();
 public boolean isPending();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty **Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject**: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.66. KlarosTestSegment

This class provides access to the information stored for test segments.

```
B.1.1.66.1. Synopsis
```

```
public String getPostcondition();
public String getPrecondition();
public KlarosRevision<KlarosTestSegment, TestSegment> getPredecessor();
public KlarosRevision<KlarosTestSegment, TestSegment> getRoot();
public String getShortname();
public List<KlarosTestCaseStep> getSteps();
public KlarosRevision<KlarosTestSegment, TestSegment> getSuccessor();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosRevision: getRevisionComment , get-RevisionId

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code





This class encapsulates the (dynamic) properties of a klaros object.

B.1.1.67.1. Synopsis

}

```
implements IKlarosTestStepContainer<T> {
```

Direct known subclasses: de.verit.klaros.core.model.KlarosTestCaseStep

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code



B.1.1.68. KlarosTestSuite

This class provides access to the information stored for test suites.

B.1.1.68.1. Synopsis

implements IKlarosTestSuite<KlarosTestSuite, TestSuite> {

```
// Public Methods
public Set<KlarosAttachment> getAttachments();
public Set<KlarosCategoryNode> getCategories();
public KlarosConfiguration getConfiguration();
public KlarosTestSuite getPredecessor();
public List<KlarosTestSuiteResult> getResults();
public KlarosTestSuite getRoot();
public String getShortname();
public KlarosTestSuite getSuccessor();
public KlarosSUTImplementation getSut();
public List<KlarosTestCase> getTestCases();
public List<KlarosTestSuiteResultCount();
}</pre>
```

Methods inherited from de.verit.klaros.core.model.KlarosRevision: getRevisionComment , get-RevisionId

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

 $\label{eq:methods} \textbf{Methods inherited from java.lang.Object}: \texttt{getClass, notify, notifyAll, toString, wait}$


B.1.1.69. KlarosTestSuiteResult

This class provides access to the information stored for test suite results.

B.1.1.69.1. Synopsis

```
public boolean isPending();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosResult: getConfiguration , get-Description , getExecutionTime , getJob , getSummary , getVerdict , isError , isFailure , isInconclusive , isPassed , isSkipped

Methods inherited from de.verit.klaros.core.model.KlarosNamedEntity: getAssignee , get-Properties , getPropertyValue , isDefinedProperty

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.1.70. KlarosUser

The user object.

B.1.1.70.1. Synopsis

public class KlarosUser extends

KlarosLabeledObject<KlarosUser, User>

```
// Public Methods
public int compareTo(KlarosUser o);
public boolean equals(Object o);
public String getEmail();
public UserRole getRole();
public String getUsername();
public int hashCode();
public String toString();
}
```

Methods inherited from de.verit.klaros.core.model.KlarosLabeledObject: asIterable , get-Created , getCreator , getKey , getLastEditor , getLastUpdated , getName , isEnabled

Methods inherited from de.verit.klaros.core.model.KlarosWrapper: equals, getWrapped, hash-Code

Methods inherited from java.lang.Object: getClass, notify, notifyAll, toString, wait



B.1.2. Deprecated API

B.1.2.1. Deprecated Methods

de.verit.klaros.core.model.IKIa- use getSystemsUnderTest rosIssue.getAcceptedIn()

B.2. Scripting API Reference

B.2.1. de.verit.klaros.scripting

B.2.1.1. IKlarosQueryFactory

A factory for executing Klaros queries.

B.2.1.1.1. Synopsis

```
public interface IKlarosQueryFactory extends Serializable {
```

```
// Public Methods
public abstract List<?> execute(String query);
public abstract List<?> execute(String query, ParameterContext params);
}
```



B.2.1.1.2. execute(String)

public abstract List<?> execute(String query);

Prepares the given query string with our KlarosOdaQuery and returns a KlarosList with the results of the query.

	Parameters
	the query
	the list of query results
meter	Context)

```
B.2.1.1.3. execute(String, ParameterContext)
```

public abstract List<?> execute(String query, ParameterContext params);

Prepares the given query string with our KlarosOdaQuery and returns a KlarosList with the results of the query.

Parameters

query

query

return

the query

params

the parameters

return

the list

B.2.1.2. KlarosContext

Context to provide all methods to the user to add own objects to the event context.

B.2.1.2.1. Synopsis

public class KlarosContext implements IKlarosContext {

```
// Public Constructors
 public KlarosContext(Context context,
                   UUID authenticatedUserId,
                   UUID activeProjectId,
                   UUID activeIterationId,
                   String applicationUrl,
                   IKlarosQueryFactory factory,
                   ParameterContext parameters);
 public KlarosContext(Context context,
                   UUID authenticatedUserId,
                   UUID activeProjectId,
                   UUID activeIterationId,
                   String applicationUrl,
                   IKlarosQueryFactory factory,
                   ParameterContext parameters,
                   Locale locale);
 // Public Methods
 public void add(String name, Object value);
 public boolean containsParameter(String name);
 public List<?> executeParameterizedQuery(String query);
 public List<?> executeQuery(String query);
 public KlarosIteration getActiveIteration();
 public KlarosConfiguration getActiveProject();
 public String getAttachmentUrl(String attachmentId);
 public String getAttachmentUrl(String attachmentId, String version);
 public String getAttachmentUrl(UUID attachmentId);
 public String getAttachmentUrl(UUID attachmentId, String version);
 public String getBrowsePageUrl(KlarosLabeledObject<?, ?> artifact);
 public Locale getLocale();
 public Parameter getParameter(String name);
 public Object getParameterValue(String name);
 public String getPrintPageUrl(KlarosLabeledObject<?, ?> artifact);
 public void setLocale(Locale locale);
}
```

Methods inherited from java.lang.Object: equals, getClass, hashCode, notify, notifyAll, to-String, wait



B.2.1.2.2. KlarosContext(Context, UUID, UUID, UUID, String, IKlarosQueryFactory, Parameter-Context)

public KlarosContext(Context context,

UUID authenticatedUserId, UUID activeProjectId, UUID activeIterationId, String applicationUrl, IKlarosQueryFactory factory, ParameterContext parameters);

Create a KlarosContext.

Parameters

context

Gets passed by the relating servlet.

authenticatedUserId

the authenticated user id

activeProjectId	the active project id
activeIterationId	the active iteration id if available
applicationUrl	the application url
factory	the factory
parameters	the parameters

B.2.1.2.3. KlarosContext(Context, UUID, UUID, UUID, String, IKlarosQueryFactory, Parameter-Context, Locale)

Create a KlarosContext.

Parameters

context	Gets passed by the relating servlet.
authenticatedUserId	the authenticated user id
activeProjectId	the active project id
activeIterationId	the active iteration id
applicationUrl	the application url
factory	the factory
parameters	the parameters
locale	the locale

B.2.2. de.verit.klaros.scripting.context

B.2.2.1. IKlarosContext

The context object encapsulates the input/output interface a report script can access. <orderedlist> <listitem>

The parameters given before starting the report. </listitem> An HQL query interface for retrieving model objects from the database. </listitem> <listitem>

A writable context for storing values for template access. </listitem> <listitem>

Access to current Klaros settings (current project, current iteration, locale settings) </listitem>

URL conversions for generating URLs to attachments, page and print page objects inside the application. </listitem> </orderedlist>

The following entries are available

B.2.2.1.1. Synopsis

public interface IKlarosContext {

```
// Public Methods
public abstract void add(String name, Object value);
public abstract boolean containsParameter(String parameterName);
public abstract List<?> executeParameterizedQuery(String query);
public abstract List<?> executeQuery(String query);
public abstract KlarosIteration getActiveIteration();
public abstract KlarosConfiguration getActiveProject();
public abstract String getAttachmentUrl(String attachmentId);
public abstract String getAttachmentUrl(String attachmentId, String revision);
public abstract String getAttachmentUrl(UUID attachmentId);
public abstract String getAttachmentUrl(UUID attachmentId, String revision);
public abstract String getBrowsePageUrl(KlarosLabeledObject<?, ?> artifact);
public abstract Locale getLocale();
public abstract Parameter getParameter(String parameterName);
public abstract Object getParameterValue(String name);
public abstract String getPrintPageUrl(KlarosLabeledObject<?, ?> artifact);
public abstract void setLocale(Locale locale);
```

«interface» IKlarosContext
 + add(name: String, value: Object): void + getActiveProject(): KlarosConfiguration + getActiveIteration(): KlarosIteration + executeQuery(query: String): List<? > + executeParameterizedQuery(query: String): List<? > + containsParameter(parameterName: String): boolean + getParameter(parameterName: String): Parameter + getParameterValue(name: String): Object + getLocale(): Locale + setLocale(locale: Locale): void + getAttachmentUrl(attachmentId: UUID): String + getAttachmentUrl(attachmentId: String): String + getAttachmentUrl(attachmentId: UUID, revision: String): String + getAttachmentUrl(attachmentId: String, revision: String): String + getBrowsePageUrl(artifact: KlarosLabeledObject<?, ?>): String + getPrintPageUrl(artifact: KlarosLabeledObject<?, ?>): String

B.2.2.1.2. add(String, Object)

public abstract void add(String name, Object value);

Add a new object with the give key to the event context.

Parameters

name

the name

value

the value

B.2.2.1.3. containsParameter(String)

public abstract boolean containsParameter(String parameterName);

Checks if the parameter is available in the context.

Parameters

parameterName the parameter name

return

true, if successful

B.2.2.1.4. executeParameterizedQuery(String)

public abstract List<?> executeParameterizedQuery(String query);

Execute the given query with the Parameters from the ParameterContext.

Parameters

query

The HQL query to execute.

	return	A KlarosList of the selected objects.
<pre>B.2.2.1.5. executeQuery(String) public abstract List<?> executeQuery(String query);</pre>		
	Execute the given query.	
		Parameters
	query	The HQL query to execute.
	return	A KlarosList of the selected objects.
B.2.2.1.6. getActiveIteration() public abstract KlarosIteration getActiveIteration();		
	Gets the active iteration.	
		Parameters
	return	the active iteration
<pre>B.2.2.1.7. getActiveProject() public abstract KlarosConfiguration getActiveProject();</pre>		
	Gets the active project.	
		Parameters
	return	the active project
B.2.2.1.8. getAttachmentUrl(String) public abstract String getAttachmentUrl(String attachmentId);		
	Return an URL string to retrieve the attachment with given attachment id.	
		Parameters
	attachmentId	the attachment id
	return	the attachment url
B.2.2.1.9. getAttachmentUrl(String, String) public abstract String getAttachmentUrl(String attachmentId, String revision);		
	Return an URL string to retrieve the attachment with given attachment id string and revision.	
		Parameters
	attachmentId	the attachment id
	revision	the revision
	return	the attachment url

B.2.2.1.10. getAttachmentUrl(UUID)

public abstract String getAttachmentUrl(UUID attachmentId);

Return an URL string to retrieve the attachment with given attachment UUID.

Parameters

attachmentId the attachment id

return the attachment url

B.2.2.1.11. getAttachmentUrl(UUID, String)

public abstract String getAttachmentUrl(UUID attachmentId, String revision);

Return an URL string to retrieve the attachment with given attachment UUID and revision.

	Parameters
attachmentId	the attachment id
revision	the revision string
return	the attachment url

B.2.2.1.12. getBrowsePageUrl(KlarosLabeledObject<?, ?>)

public abstract String getBrowsePageUrl(KlarosLabeledObject<?, ?> artifact);

Gets the browse page url to the given klaros artifact. Will return null if the artifact type is not supported.

Parameters

artifact	the artifact

return

the browse page url

```
B.2.2.1.13. getLocale()
```

public abstract Locale getLocale();

/** Returns the active Java Locale setting.

Parameters

the Java Locale

B.2.2.1.14. getParameter(String)

public abstract Parameter getParameter(String parameterName);

Gets the parameter.

Parameters

parameterName	the parameter name
return	The parameter with the given name
	Exceptions
Klaros- Illegal-	if the parameter is not available.

Argument-Exception

Gets the parameter value.

Parameters

name

the name

return

The parameter with the given name

if the parameter is not available.

Exceptions

Klaros-Illegal-Argument-Exception

B.2.2.1.16. getPrintPageUrl(KlarosLabeledObject<?, ?>)

public abstract String getPrintPageUrl(KlarosLabeledObject<?, ?> artifact);

Gets the print page of the given klaros artifact. Will return null if the artifact type is not supported.

Parameters

artifact

the artifact

return

the print page url

B.2.2.1.17. setLocale(Locale)

public abstract void setLocale(Locale locale);

Sets the active Java Locale setting.

Parameters

locale

the new locale

B.2.3. de.verit.klaros.scripting.model

B.2.3.1. Parameter

The Parameter class encapsulates a report query parameter.

B.2.3.1.1. Synopsis

public class Parameter implements Serializable {

// Public Constructors
public Parameter(ScriptParameter source);

// Public Methods
public String getDefaultValue();

```
public List<String> getDefaultValues();
public String getDescription();
public String getLabel();
public String getName();
public List<String> getOptions();
public String getOptionsString();
public ScriptParameterType getType();
public String getValue();
public List<String> getValues();
public List<String> getValues();
public boolean isMandatory();
}
```

Methods inherited from java.lang.Object: equals,getClass,hashCode,notify,notifyAll,to-String,wait



B.2.3.1.2. Parameter(ScriptParameter)

public Parameter(ScriptParameter source);

Instantiates a new parameter.

Parameters

source

the source script parameter

Gets the default value.

		Parameters
	return	the default value
B.2.3	.1.4. getDefaultValues() public List <string> getDefaul</string>	tValues();
	Gets the default values.	
		Parameters
	return	the default values
B.2.3	.1.5. getDescription() public String getDescription());
	Gets the description.	
		Parameters
	return	the description
B.2.3	.1.6.getLabel() public String getLabel();	
	Gets the label.	
		Parameters
	return	the label
B.2.3	.1.7.getName() public String getName();	
	Gets the name.	
		Parameters
	return	the name
B.2.3	.1.8. getOptions() public List <string> getOption</string>	S();
	Gets the options.	
		Parameters
	return	the options
B.2.3	.1.9. getOptionsString() public String getOptionsStrin	g();
	Gets the options string.	
		Parameters
	return	the options string

<pre>B.2.3.1.10. getType() public ScriptParameterType getType();</pre>	
Gets the type.	
	Parameters
return	the type
B.2.3.1.11.getValue() public String getValue();	
Gets the value.	
	Parameters
return	the value
B.2.3.1.12. getValues() public List <string> getValues</string>	s();
Gets the value.	
	Parameters
return	the value
<pre>B.2.3.1.13. isMandatory() public boolean isMandatory();</pre>	
Checks if is mandatory.	
	Parameters
return	true, if is mandatory

Anhang C. Dateiformat für Testfall-Import

Die folgende Tabelle zeigt eine kurze Übersicht der verfügbaren Elemente:

Name	Beschreibung
<u><areatopic></areatopic></u>	(Testart) Eine Gruppe von Testaktivitäten basierend auf bestimmten - Testzielen mit dem Zweck, eine Komponente oder ein System auf spezi- fische Merkmale zu prüfen (FUNCTIONAL, NON_FUNCTIONAL, STRUCT- URAL, REGRESSION oder RE_TEST).
<attachment></attachment>	Ein Anhang.
<attachments></attachments>	Die Liste der Anhänge.
<attribute></attribute>	Ein benutzerdefiniertes Feld.
<attributes></attributes>	Die Liste der benutzerdefinierten Felder.
<category></category>	Ein Knoten eines Kategoriebaums.
< <u>categoryDescripti-</u> on>	Die Beschreibung der Kategorie.
<categoryname></categoryname>	Der Name der Kategorie.
<categorytree></categorytree>	Ein Kategoriebaum.
<container></container>	Das Wurzelelement dieses Schemas, das alle Testfälle und optional eine Testsuite/Kategorienstruktur enthält.
<content></content>	Der in base64 codierte Inhalt des Anhangs.
<depends></depends>	Die Abhängigkeiten des Testfalls in Textform.
<description></description>	Die Beschreibung des Testfalls in Textform.
<docbase></docbase>	Das Referenzdokument mit der Anforderung, auf der der Testfall basiert.
<pre><estimatedduration></estimatedduration></pre>	Die geschätzte Dauer in Millisekunden.
<evaluation></evaluation>	Das Auswertungsverfahren des Testfalls (MANUAL oder AUTO).
<execution></execution>	Das Ausführungsart des Testfalls (MANUAL oder AUTO).
<expectedresult></expectedresult>	Das erwartete Ergebnis.
<externalid></externalid>	Die externe ID dieses Testfalls.
<u><level></level></u>	Der Testlevel dieses Testfalls (COMPONENT, INTEGRATION, SYSTEM oder ACCEPTANCE).
<method></method>	(Entwurfsverfahren) Die Methode des Testdesigns für das zu testende - System (BLACK_BOX oder WHITE_BOX).
<u><name></name></u>	Der Name eines benutzerdefinierten Feldes.
<note></note>	Ein Platzhalter für beliebige, kurze Notizen zum Testfall.
<postcondition></postcondition>	Ein Text, der die Nachbedingung beschreibt.
<precondition></precondition>	Ein Text, der die Vorbedingung beschreibt.
<priority></priority>	Die Priorität des Testfalls (LOW, MEDIUM oder HIGH).
<revision></revision>	Die Revision des Testfalls.

Name	Beschreibung
<shortname></shortname>	Der Name des Testfalls.
<u><state></state></u>	Der Testfall-Status (nicht genutzt, für zukünftige Zwecke reserviert).
<step></step>	Ein Testschritt.
<steps></steps>	Die Liste der Testschritte.
<u><team></team></u>	Das Team, welches für diesen Testfall verantwortlich ist.
<testcase></testcase>	Ein Testfall.
<testcases></testcases>	Die Liste der Testfälle.
<testsuite></testsuite>	Wenn dieses Element vorhanden ist, wird beim Import eine Testsuite mit diesem Namen angelegt.
<traceability></traceability>	(Verfolgbarkeit) Eine Referenz zur zugehörigen Anforderung, Use Case oder Arbeitspaket.
<value></value>	Der Wert eines benutzerdefinierten Feldes.
<u><variety></variety></u>	(Ergebnisart) Das erwartete Ergebnis eines Testfalls (POSITIVE oder NE-GATIVE).

Tabelle C.1. Übersicht der Elemente

C.1. <step>

Ein Testschritt.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>
steps> .

Name	Vorkommen	Beschreibung
<description></description>	01	Die Aktion, die in diesem Testfallschritt durch- geführt werden soll.
<expectedresult></expectedresult>	01	Das erwartete Ergebnis.
<postcondition></postcondition>	01	Die Nachbedingung.
<precondition></precondition>	01	Die Vorbedingung.

Tabelle C.2. <step> -Elemente

C.2. <steps>

Die Liste der Testschritte.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

Name	Vorkommen	Beschreibung
<step></step>	0n	Ein Testschritt.

Tabelle C.3. <steps> -Elemente

C.3. <attachments>

Die Liste der Anhänge.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

Name	Vorkommen	Beschreibung
<attachment></attachment>	0n	Ein binärer Anhang.

Tabelle C.4. <attachments> -Elemente

C.4. <attachment>

Ein Anhang.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: statachments .

Name	Vorkommen	Beschreibung
<u><name></name></u>	11	Der Name des Anhangs.
< <u>content></u>	11	Der binäre Inhalt des Anhangs, kodiert im - Base-64-Format.

Tabelle C.5. <attachment> -Elemente

C.5. <attribute>

Ein benutzerdefiniertes Feld.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><attributes></u>.

Name	Vorkommen	Beschreibung
<name></name>	11	Der Name des benutzerdefiniertes Felds.
< <u>value></u>	11	Der Wert des benutzerdefiniertes Felds.

Tabelle C.6. <attribute> -Elemente

C.6. <attributes>

Die Liste der benutzerdefinierten Felder.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

Name	Vorkommen	Beschreibung
<attribute></attribute>	0n	

Tabelle C.7. <attributes> -Elemente

C.7. <category>

Ein Knoten eines Kategoriebaums. Er kann weitere Unterkategorien enthalten.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: .categoryTrees">.categoryTrees">category>.categoryTrees, categoryScategoryTrees, www.categoryScategoryTrees)

Name	Vorkommen	Beschreibung
<categoryname></categoryname>	11	Der Name des Kategorieknotens.
<categorydescription></categorydescription>	11	Die Beschreibung des Kategorieknotens.
<category></category>	01	Der übergeordnete Kategorieknoten.

Tabelle C.8. <category> -Elemente

C.8. <categoryTree>

Ein Kategoriebaum. Er enthält den Wurzelkategorieknoten mit einer beliebigen Anzahl von Unterkategorien.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><container></u>.

Name	Vorkommen	Beschreibung
<categoryname></categoryname>	11	Der Name des Kategoriebaums.
< <u>category></u>	11	Der Wurzelknoten des Kategoriebaums.

Tabelle C.9. <categoryTree> -Elemente

C.9. <container>

Das Wurzelelement dieses Schemas, das alle Testfälle und optional eine Testsuite/Kategorienstruktur enthält.

Name	Vorkommen	Beschreibung
<u><testsuite></testsuite></u>	01	Eine optionale Testsuite, die die aufgeführten - Testfälle enthält.
<testcases></testcases>	01	Die Liste der Testfälle.
<categorytree></categorytree>	01	Der optionale Kategorienbaum dieser - Anforderungen.

Tabelle C.10. <container> -Elemente

C.10. <testcases>

Die Liste der Testfälle.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>container> .

Name	Vorkommen	Beschreibung
<testcase></testcase>	0n	Ein Testfall.

Tabelle C.11. <testcases> -Elemente

C.11. <testcase>

Ein Testfall.

Name	Vorkommen	Beschreibung
<areatopic></areatopic>	01	Der Typ des Testfalls (FUNCTIONAL, NON_FU- NCTIONAL, STRUCTURAL, REGRESSION or R- E_TEST).
<attachments></attachments>	01	Die Liste der binären Anhänge.
<attributes></attributes>	01	Die benutzerdefinierten Felder, die dieser - Testfall besitzt.
<category></category>	01	Der Kategorieknoten dem dieser Testfall zuge- ordnet ist.
<depends></depends>	01	Die Abhängigkeiten des Testfalls in Textform.
<pre><description></description></pre>	01	Die Beschreibung des Testfalls in Textform.
<pre><docbase></docbase></pre>	01	Das Referenzdokument mit der Anforderung, auf der der Testfall basiert.
<pre><estimatedduration></estimatedduration></pre>	01	Die geschätzte Dauer in Millisekunden für die - Ausführung dieses Testfalls.

Name	Vorkommen	Beschreibung
<evaluation></evaluation>	01	Das Auswertungsverfahren des Testfalls (MA- NUAL oder AUTO).
<execution></execution>	01	Das Ausführungsart des Testfalls (MANUAL oder AUTO).
<expectedresult></expectedresult>	01	Das erwartete Ergebnis.
<externalid></externalid>	01	Die externe ID dieses Testfalls. Sie wird benö- tigt, wenn in Anforderungsimporten auf - Testfälle verwiesen wird.
<level></level>	01	Der Testlevel dieses Testfalls (COMPONENT, I- NTEGRATION, SYSTEM oder ACCEPTANCE).
<method></method>	01	(Entwurfsverfahren) Die Methode des - Testdesigns für das zu testende System (BLA- CK_BOX oder WHITE_BOX).
<note></note>	01	Beliebige, kurze Notiz zum Testfall.
<postcondition></postcondition>	01	Die Nachbedingung.
<precondition></precondition>	01	Die Vorbedingung.
<priority></priority>	01	Die Priorität des Testfalls (LOW, MEDIUM oder HIGH).
<revision></revision>	01	Die Revision des Testfalls.
<shortname></shortname>	01	Der Name des Testfalls.
< <u>state></u>	01	Der Testfall-Status (nicht genutzt, für zukünfti- ge Zwecke reserviert).
<steps></steps>	01	Die Liste der Testschritte.
<team></team>	01	Das Team, welches für diesen Testfall verant- wortlich ist.
<traceability></traceability>	01	(Verfolgbarkeit) Eine Referenz zur zugehörigen Anforderung, Use Case oder Arbeitspaket.
< <u>variety></u>	01	(Ergebnisart) Das erwartete Ergebnis eines - Testfalls (POSITIVE oder NEGATIVE).

Tabelle C.12. <testcase> -Elemente

C.12. <testsuite>

Wenn dieses Element vorhanden ist, wird beim Import eine Testsuite mit diesem Namen angelegt. Diese Testsuite enthält dann alle Testfälle aus diesem Container.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>container> .

C.13. <externalId>

Die externe ID dieses Testfalls. Sie wird benötigt, wenn in Anforderungsimporten auf Testfälle verwiesen wird.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.14. <areatopic>

(Testart) Eine Gruppe von Testaktivitäten basierend auf bestimmten Testzielen mit dem Zweck, eine Komponente oder ein System auf spezifische Merkmale zu prüfen (FUNCTIONAL, NON_FUNCTIONAL, STRUCTURAL, REGRESSION oder RE_TEST).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.15. <depends>

Die Abhängigkeiten des Testfalls in Textform.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.16. <description>

Die Beschreibung des Testfalls in Textform.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><step></u>, <u><testcase></u>.

C.17. <docbase>

Das Referenzdokument mit der Anforderung, auf der der Testfall basiert.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.18. <estimatedDuration>

Die geschätzte Dauer in Millisekunden.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.19. <evaluation>

Das Auswertungsverfahren des Testfalls (MANUAL oder AUTO).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.20. <execution>

Das Ausführungsart des Testfalls (MANUAL oder AUTO).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.21. <expectedResult>

Das erwartete Ergebnis.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><step></u>, <u><testcase></u>.

C.22. < |eve| >

Der Testlevel dieses Testfalls (COMPONENT, INTEGRATION, SYSTEM oder ACCEPTANCE).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.23. <method>

(Entwurfsverfahren) Die Methode des Testdesigns für das zu testende System (BLACK_BOX oder WHITE_BOX).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.24. <note>

Ein Platzhalter für beliebige, kurze Notizen zum Testfall.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.25. <postcondition>

Ein Text, der die Nachbedingung beschreibt.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><step></u>, <u><testcase></u>.

C.26. <precondition>

Ein Text, der die Vorbedingung beschreibt.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>step>, <a>testcase>.

C.27. <revision>

Die Revision des Testfalls. Dieses Element wird nur beim Export genutzt, beim Import wird automatisch die Revisions-Id "1.0" vergeben.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.28. <priority>

Die Priorität des Testfalls (LOW, MEDIUM oder HIGH).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.29. <shortname>

Der Name des Testfalls.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.30. <state>

Der Testfall-Status (nicht genutzt, für zukünftige Zwecke reserviert).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.31. <team>

Das Team, welches für diesen Testfall verantwortlich ist.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.32. <traceability>

(Verfolgbarkeit) Eine Referenz zur zugehörigen Anforderung, Use Case oder Arbeitspaket.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.33. <variety>

(Ergebnisart) Das erwartete Ergebnis eines Testfalls (POSITIVE oder NEGATIVE).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a><testcase>

C.34. <name>

Der Name eines benutzerdefinierten Feldes.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a href="https://www.attributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattributes-cattri

C.35. <value>

Der Wert eines benutzerdefinierten Feldes.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: cattributes- .

C.36. <content>

Der in base64 codierte Inhalt des Anhangs.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: attachmentscore</attachmentscore">attachmentscore</attachmentscore</attachmentscore">></attachmentscore</attachmentscore</attachmentscore</attachmentscore">attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore">attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore">attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachmentscore</attachm

C.37. <categoryName>

Der Name der Kategorie.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>category>, <categoryTree>

C.38. <categoryDescription>

Die Beschreibung der Kategorie.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>category> .

Anhang D. Dateiformat für Anforderungs-Import

Name	Beschreibung
<attachment></attachment>	Ein Anhang.
<attachments></attachments>	Die Liste der binären Anhänge.
<attribute></attribute>	Ein benutzerdefiniertes Feld.
<attributes></attributes>	Die Liste der benutzerdefinierten Felder.
<category></category>	Ein Knoten eines Kategoriebaums.
< <u>categoryDescripti-</u> on>	Die Beschreibung der Kategorie.
<categoryname></categoryname>	Der Name der Kategorie.
<categorytree></categorytree>	Ein Kategoriebaum.
<container></container>	Das Root-Element dieses Schemas, das alle Anforderungen enthält.
<content></content>	Der in base64 codierte Inhalt des Anhangs.
<description></description>	Die Beschreibung der Anforderung in Textform.
<externalid></externalid>	Die externe Id dieser Anforderung, deren Format frei gewählt werden kann.
<externalrevision></externalrevision>	Die externe Revision dieses Anforderung (nicht genutzt, für zukünftige - Zwecke reserviert).
< <u>externalTestCa-</u> <u>selds></u>	Die Liste der externen Testfall-IDs, die diese Anforderung abdecken.
<name></name>	Der Name eines Feldes.
<priority></priority>	Die Priorität der Anforderung (LOW, MEDIUM oder HIGH).
<requirement></requirement>	Eine Anforderung.
<requirements></requirements>	Die Liste der Anforderungen.
<revision></revision>	Die Revision der Anforderung.
<shortname></shortname>	Der Name der Anforderung.
<summary></summary>	Eine kurze Zusammenfassung der Anforderung.
<value></value>	Der Wert eines Feldes.

Die folgende Tabelle zeigt eine kurze Übersicht der verfügbaren Elemente:

Tabelle D.1. Übersicht der Elemente

D.1. <attachments>

Die Liste der binären Anhänge.

Name	Vorkommen	Beschreibung
<attachment></attachment>	0n	Ein Anhang.

Tabelle D.2. <attachments> -Elemente

D.2. <attachment>

Ein Anhang.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: sattachments .

Name	Vorkommen	Beschreibung
<name></name>	11	Der Name des Anhangs.
<content></content>	11	Der binäre Inhalt des Anhangs, kodiert im - Base-64-Format.

Tabelle D.3. <attachment> -Elemente

D.3. <attributes>

Die Liste der benutzerdefinierten Felder.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

Name	Vorkommen	Beschreibung
<u><attribute></attribute></u>	0n	Ein benutzerdefiniertes Feld.

Tabelle D.4. <attributes> -Elemente

D.4. <attribute>

Ein benutzerdefiniertes Feld.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <u><attributes></u>.

Name	Vorkommen	Beschreibung
<name></name>	11	Der Name des benutzerdefiniertes Felds.
<value></value>	11	Der Wert des benutzerdefiniertes Felds.

Tabelle D.5. <attribute> -Elemente

D.5. <category>

Ein Knoten eines Kategoriebaums. Er kann weitere Unterkategorien enthalten.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: .categoryTree">category>.categoryTree, category>.categoryTree, <a href="https://www.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categorys.categ

Name	Vorkommen	Beschreibung
<categoryname></categoryname>	11	Der Name des Kategorieknotens.
<categorydescription></categorydescription>	11	Die Beschreibung des Kategorieknotens.
<category></category>	01	Der übergeordnete Kategorieknoten.

Tabelle D.6. <category> -Elemente

D.6. <categoryTree>

Ein Kategoriebaum. Er enthält den Wurzelkategorieknoten mit einer beliebigen Anzahl von Unterkategorien.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>container> .

Name	Vorkommen	Beschreibung
<categoryname></categoryname>	11	Der Name des Kategoriebaums.
<category></category>	11	Der Wurzelknoten des Kategoriebaums.

Tabelle D.7. <categoryTree> -Elemente

D.7. <container>

Das Root-Element dieses Schemas, das alle Anforderungen enthält.

Name	Vorkommen	Beschreibung
<requirements></requirements>	01	Die Liste der Anforderungen.
<categorytree></categorytree>	01	Der optionale Kategorienbaum dieser - Anforderungen.

Tabelle D.8. <container> -Elemente

D.8. <requirements>

Die Liste der Anforderungen.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>container> .

Name	Vorkommen	Beschreibung
<requirement></requirement>	0n	Eine Anforderung.

Tabelle D.9. <requirements> -Elemente

D.9. <requirement>

Eine Anforderung.

Name	Vorkommen	Beschreibung
<attributes></attributes>	01	Die benutzerdefinierten Felder, die diese - Anforderung besitzt.
<category></category>	01	Der Kategorieknoten dem dies Anforderung zugeordnet ist.
<description></description>	01	Die ausführliche Beschreibung der - Anforderung in Textform.
<externalid></externalid>	01	Die externe Id dieser Anforderung.
<priority></priority>	01	Die Priorität der Anforderung.
<revision></revision>	01	Die Revision der Anforderung.
<shortname></shortname>	01	Der Name der Anforderung.
<u><summary></summary></u>	01	Eine kurze Zusammenfassung der - Anforderung.
<attachments></attachments>	01	Die Liste der binären Anhänge.
<externaltestcaseids></externaltestcaseids>	01	Die Liste der externen Testfall-IDs, die diese - Anforderung abdecken.

Tabelle D.10. <requirement> -Elemente

D.10. <externalTestCaseIds>

Die Liste der externen Testfall-IDs, die diese Anforderung abdecken.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<a> <a> <a> <a> <a></

Name	Vorkommen	Beschreibung
<externalid></externalid>	0n	Eine ID eines externen Testfalls, der die -
		Anforderung abdeckt.

Tabelle D.11. <externalTestCaseIds> -Elemente

D.11. <externalId>

Die externe Id dieser Anforderung, deren Format frei gewählt werden kann. Dieses Element wird nur für die Anforderungs-Synchronisation genutzt und beim Import nicht berücksichtigt. Wird vor einer Synchronisation die Id einer Anforderung geändert, wird in Klaros-Testmanagement eine neue Anforderung mit dieser Id angelegt. Sonst wird die vorhandene Anforderung mit den Daten aus dieser XML-Datei überschrieben.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

D.12. <externalRevision>

Die externe Revision dieses Anforderung (nicht genutzt, für zukünftige Zwecke reserviert).

D.13. <description>

Die Beschreibung der Anforderung in Textform.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

D.14. <priority>

Die Priorität der Anforderung (LOW, MEDIUM oder HIGH).

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

D.15. <revision>

Die Revision der Anforderung. Dieses Element wird nur beim Export und Synchronisieren genutzt, beim Import wird automatisch die Revisions-Id "1.0" vergeben.

D.16. <shortname>

Der Name der Anforderung.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

D.17. <summary>

Eine kurze Zusammenfassung der Anforderung.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>

<u><requirement></u> .

D.18. <name>

Der Name eines Feldes.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: . > . . > . . > . . > . ></attributes-catteriou

D.19. <value>

Der Wert eines Feldes.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a tribute> .

D.20. <content>

Der in base64 codierte Inhalt des Anhangs.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: </a href="https://www.attachmentscore">attachmentscore

D.21. <categoryName>

Der Name der Kategorie.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>category>, <categoryTree>

D.22. <categoryDescription>

Die Beschreibung der Kategorie.

Dieses Element kann innerhalb der folgenden Elemente vorkommen: <a>category> .

Anhang E. Die Reporting Context API

E.1. Der Klaros Report-Kontext

Name	Beschreibung	Тур
date	Der Zeitpunkt der - Berichtsausführung.	java.util.Date
locale	Die vom Benutzer ausge- wählte Lokalisierung.	java.util.Locale
activeProject	Das aktuell ausgewählte - Projekt oder null, wenn kein Projekt ausgewählt ist.	de.verit.klaros.model.KlarosConfiguration
activelteration	Die aktuell ausgewählte - Iteration oder null, wenn keine Iteration ausgewählt ist.	de.verit.klaros.model.KlarosIteration
user	Der aktuelle Benutzer, der den Report erzeugt.	de.verit.klaros.model.KlarosUser
parameters	Eine Map von Parameter Namen zu Parameter Objekten, die die beim - Ausführen des Berichts ein- gegebenen benutzerdefi- nierten Parameter enthält.	java.util.Map java.lang.String,de.verit. klaros.scripting.model.Parameter

Tabelle E.1. Kontextvariablen

Auf die Kontext-Variablen kann in SeamPDF beispielsweise so zugegriffen werden:

```
<p:text value="#{date}" />
```



Anmerkung

#{user.name} und #{user.username} enthalten verschiedene Werte. Ersterer gibt den vollen Namen des Benutzers an, während letzterer den Namen angibt, mit dem der Benutzer angemeldet ist.

E.2. KlarosScript Interface

```
package de.verit.klaros.scripting;
/**
 * Public interface for generating user defined reports.
 */
public interface KlarosScript {
    /**
    * This method starts the report generation.
    *
```

```
* @param context
* The event context to provide all needed functions, properties
* and objects.
* @throws KlarosScriptingException if executing the script fails
*/
void execute(KlarosContext context) throws KlarosScriptingException;
```

E.3. Beispiel Layout-Vorlage

```
<p:document xmlns:ui="http://java.sun.com/jsf/facelets"
 xmlns:f="http://java.sun.com/jsf/core" xmlns:p="http://jboss.org/schema/seam/pdf"
 title="Klaros-Testmanagement Test Suite Report" marginMirroring="true"
 author="#{user.name}" creator="#{user.name}" pageSize="A4">
 <f:facet name="header">
   <p:font size="8">
     <p:header borderWidthBottom="0.1" borderColorBottom="black" borderWidthTop="0" alignment="center">
       <p:text value="Example report - generated #{date} by #{user.name}"/>
     </p:header>
     <p:footer borderWidthTop="0.1" borderColorTop="black" borderWidthBottom="0" alignment="center">
       <p:text value="Page " />
       <p:pageNumber />
     </p:footer>
   </p:font>
 </f:facet>
 <!-- print the frontpage -->
 <p:paragraph alignment="center" spacingAfter="100">
   <p:text value="" />
 </p:paragraph>
 <p:font style="bold" size="32">
   <p:paragraph alignment="center" spacingAfter="75">
     <p:text value="Test Case Report" />
   </p:paragraph>
 </p:font>
 <p:font style="normal" size="12">
   <p:paragraph alignment="center" spacingAfter="5">
     <p:text value="Created by" />
   </p:paragraph>
 </p:font>
 <p:font style="bold" size="16">
   <p:paragraph alignment="center" spacingAfter="5">
     <p:text value="#{user.name} (#{user.email})"/>
   </p:paragraph>
 </p:font>
 <p:font style="normal" size="12">
   <p:paragraph alignment="center" spacingAfter="5">
     <p:text value="at" />
   </p:paragraph>
 </p:font>
 <p:font style="bold" size="16">
   <p:paragraph alignment="center" spacingAfter="75">
     <p:text value="#{date}" />
   </p:paragraph>
 </p:font>
```

```
<p:newPage/>
<ui:fragment rendered="#{results != null}">
  <p:font style="normal" size="12">
   <p:paragraph alignment="left" spacingAfter="10">
      <p:text value="The test results for " />
   </p:paragraph>
  </p:font>
</ui:fragment>
<!-- Test result table -->
<p:table columns="4" widths="1 1 3 3">
  <!-- create the headline with bold characters -->
  <p:font size="10" style="bold">
    <p:cell horizontalAlignment="center" verticalAlignment="top">
      <p:paragraph>
        <p:text value="Name" />
      </p:paragraph>
   </p:cell>
    <p:cell horizontalAlignment="center" verticalAlignment="top">
      <p:paragraph>
        <p:text value="Result" />
      </p:paragraph>
   </p:cell>
    <p:cell horizontalAlignment="center" verticalAlignment="top">
      <p:paragraph>
        <p:text value="Test run description" />
      </p:paragraph>
   </p:cell>
   <p:cell horizontalAlignment="center" verticalAlignment="top">
      <p:paragraph>
        <p:text value="Summary" />
      </p:paragraph>
   </p:cell>
  </p:font>
  <!-- display the attributes of the test results -->
  <p:font size="8">
    <ui:repeat value="#{results}" var="tcr">
      <p:cell verticalAlignment="top" horizontalAlignment="left">
        <p:paragraph>
          <p:text value="#{tcr.testCase.name}" />
        </p:paragraph>
      </p:cell>
      <!-- decide which color has to be displayed, based on the test result -->
      <ui:fragment rendered="#{tcr.error}">
       <p:cell backgroundColor="rgb(255,0,0)" verticalAlignment="top" horizontalAlignment="center">
          <p:paragraph>
            <p:text value="error" />
          </p:paragraph>
        </p:cell>
      </ui:fragment>
      <ui:fragment rendered="#{tcr.failure}">
        <p:cell backgroundColor="rgb(255,215,0)" verticalAlignment="top" horizontalAlignment="center">
          <p:paragraph>
```

```
<p:text value="failure" />
           </p:paragraph>
         </p:cell>
       </ui:fragment>
       <ui:fragment rendered="#{tcr.passed}">
         <p:cell backgroundColor="rgb(0,255,0)" verticalAlignment="top" horizontalAlignment="center">
           <p:paragraph>
             <p:text value="passed" />
           </p:paragraph>
         </p:cell>
       </ui:fragment>
       <p:cell verticalAlignment="top" horizontalAlignment="left">
         <p:paragraph>
           <p:text value="#{tcr.description}" />
         </p:paragraph>
       </p:cell>
       <p:cell verticalAlignment="top" horizontalAlignment="left">
         <p:paragraph>
           <p:text value="#{tcr.summary}" />
         </p:paragraph>
       </p:cell>
       <!-- Print the test case description below the result row.
            To differ from the next row use a bigger border for the bottom. -->
       <p:cell colspan="4" verticalAlignment="top" horizontalAlignment="left"
               borderWidthBottom="1" paddingBottom="3">
         <p:paragraph>
           <p:font size="6" style="bold">
             <p:text value="Testcase description:" />
           </p:font>
           <p:font size="6">
             <p:text value="#{tcr.testCase.description}" />
           </p:font>
         </p:paragraph>
       </p:cell>
     </ui:repeat>
   </p:font>
 </p:table>
</p:document>
```

Anhang F. Verzeichnis aller Icons

F.1. Bereiche

lcon	Beschreibung
ľ	Definieren
***	Planen
\$	Ausführen
¢	Auswerten
ير	Konfigurieren

Tabelle F.1. Bereiche

F.2. Aktionen

F.2.1. Aktionen

lcon	Beschreibung
\odot	Objekt aktivieren
Q	Objekt anzeigen
ľ	Objekt bearbeiten
D	Objekt duplizieren
⑪	Objekt löschen
Ð	Objekt wiederherstellen
\diamond	Objekt dauerhaft entfernen
Ð	Objekt drucken
S	Objekt aktualisieren
ų	Revision erzeugen
+	Item hinzufügen
_	Item entfernen
×	Item leeren
	Einer Kategorie zuweisen
0	Nur aktive Iteration anzeigen
Ľ	Externen Link öffnen
ଜ	Verbindung überprüfen
٦	URL auf Gültigkeit prüfen
දු	Benutzerrolle in einem Projekt ändern
	Authentifizierung prüfen
Д	Lesezeichen anlegen

lcon	Beschreibung
샰	Download beginnen
	E-Mail verschicken
V	Filter-Bereich öffnen
8	Objekte zusammenführen
₹≡	Duplikate entfernen
	Eine Aufgaben einem Testfall oder einer Testsuite zuweisen
(Alle Einträge anzeigen
ø	Nur bereits ausgeführte Einträge anzeigen
20	Nur noch nicht ausgeführte Einträge anzeigen
U	Nur die zuletzt ausgeführten Einträge anzeigen

Tabelle F.2. Allgemeine Aktionen

F.2.2. Ausführungsaktionen

lcon	Beschreibung
\$	Objekt ausführen
.	Objekt manuell ausführen
_ C	Objektausführung blockiert
.	Objektüberprüfung durchführen
20	Objekt erneut ausführen oder aktualisieren
-5	Testergebnis importieren

Tabelle F.3. Ausführungsaktionen

F.2.3. Aktionen bei der Testausführung

lcon	Beschreibung
G	Vorheriger Schritt
M	Schritt überspringen
M	Alle verbleibenden Schritte überspringen und Test beenden
0	Die Ausführung pausieren
ľ	Schrittergebnis bearbeiten
۲	Test-Review anfordern
Ð	Neuen Issue erzeugen
()	Vorhandenen Issue verlinken
\odot	Schritt bestanden
⚠	Schritt fehlgeschlagen
۲	Fehler bei der Schrittausführung
*	Schrittergebnis unklar
lcon	Beschreibung
------	----------------------
?	Schritt übersprungen

Tabelle F.4. Aktionen bei der Testausführung

F.2.4. Ordnende Aktionen

lcon	Beschreibung
<u>م</u>	Verschiebt Objekte auf der gleichen Hierarchieebene nach oben
▽	Verschiebt Objekte auf der gleichen Hierarchieebene nach unten
1	Verschiebt Objekte auf der gleichen Hierarchieebene zur gewählten Position
D	Verschiebt Objekte eine Hierarchieebene nach unten
٩	Verschiebt Objekte eine Hierarchieebene nach oben
KK	Fügt ein Testsegment vor dem Testschritt ein
KK	Fügt ein Testsegment nach dem Testschritt ein
2	Maximiert Fenster
*	Minimiert Fenster
[]	Maximiert Einträge
	Minimiert Einträge
>	Fügt einen Testschritt rechts ein
<	Fügt einen Testschritt links ein
»	Fügt ein Testsegment rechts ein
«	Fügt ein Testsegment links ein

Tabelle F.5. Ordnende Aktionen

F.3. Tabellenoperationen

lcon	Beschreibung
æ	Kategorien-Bereich öffnen
\mathbf{V}	Filtern / Sortieren
Q	Schnellfilter anwenden
公	Tabelle exportieren
Ξ	Spalten konfigurieren

Tabelle F.6. Tabellenoperationen

F.4. Information

lcon	Beschreibung
ନ	Information
⚠	Warnung

lcon	Beschreibung
8	Fehler
A	Löschen deaktiviert
S	Mit anderen Objekten verbunden
()	Issues für dieses Objekt vorhanden
	Objekt wird von externer Quelle synchronisiert
0	Synchronisiertes Objekt hat ausstehende Aktualisierungen
Ŷ	Hinweis
ු	Aufgabe ist blockiert
00	Aufgabe hat ausstehende Abhängigkeiten

Tabelle F.7. Information

F.5. Eigenschaften

F.5.1. Ergebnisse

lcon	Beschreibung
\odot	Bestanden
⚠	Fehlgeschlagen
Θ	Fehler
*	Schrittergebnis unklar
0	Übersprungen

Tabelle F.8. Results

F.5.2. Prioritäten

lcon	Beschreibung
\otimes	Blocker
*	Kritisch
^	Hoch
≈	Niedrig
~	Trivial

Tabelle F.9. Prioritäten

F.6. Dokumentenformate

lcon	Beschreibung
GV	CSV
×	Excel

lcon	Beschreibung
\$	HTML
ß	PDF
\$	XML

Tabelle F.10. Dokumentenformate

Glossar

А

Abdeckung, Anforderungs- abdeckung	Die Anforderungsabdeckung stellt das Verhältnis der durch Tests abgedeckten Anforderungen zu der Gesamtmenge der definierten Anforderungen dar.
	In Klaros-Testmanagement können Anforderungen direkt erfasst und Testfällen zugeordnet werden. Dabei können einzelne Anforderungen durch mehrere Tests abgedeckt sein und ein Test kann mehr als eine Anforderung abdecken.
Administrator	Eine Benutzerrolle, die vollständigen Zugriff auf alle Funktionen in Kla- ros-Testmanagement hat.
	Dazu gehören u.a. Installation und Wartung der Software, Updates, Sys- temeinstellungen, Benutzerverwaltung, Backups und alle Rechte um Tests anzulegen, auszuführen und auszuwerten.
Änderungsverfolgung	Lückenloses Nachvollziehen aller Änderungen an den verwalteten Objekten durch automatisches Protokollieren der Aktivitäten.
	In Klaros-Testmanagement werden alle Änderungen auf der Detailsei- te "Änderungen" des entsprechenden Objektes farbig und zeitlich dar- gestellt.
Anforderung	Eine vom Benutzer benötigte Eigenschaft oder Fähigkeit, die eine Soft- ware erfüllen oder besitzen muss, um einen Vertrag, einen Standard, eine Spezifikation oder ein anderes formales Dokument zu erfüllen. [Nach IEEE 610].
	Anforderungen lassen sich entweder in Klaros-Testmanagement direkt verwalten oder mit externen Quellen wie z.B. JIRA synchronisieren. Ein Verweis auf die entsprechende Anforderung, z.B. auf ein Dokument, kann Testfällen und Testsuiten hinzugefügt werden (Verfolgbarkeit).
Arbeitsprotokoll	Das Arbeitsprotokoll zeigt den zeitlichen Verlauf und die Dauer der Akti- vitäten zur Bearbeitung einer Aufgabe an. Die für die gesamte Test- ausführung aufgewendete Zeit ergibt sich aus der Summe der Ausfüh- rungszeiten.
Aufgabe	Mittels Aufgaben werden Testaktivitäten geplant und protokolliert. Die Ausführung und die Ergebnisse der ausgeführten Aufgaben werden automatisch mitprotokolliert und zur Auswertung herangezogen.
Authentifizierung	Die Authentifizierung der Benutzer erfolgt in der Regel über Benutzer- name und Passwort.
	In Klaros-Testmanagement kann das Verwalten der Zugangsdaten direkt in der Software oder über ein externes LDAP- oder Active Direc- tory-Verzeichnis erfolgen.

Automatisierte Testfälle	Die Testergebnisse aus Automatisierungstools werden in Form einer Ergebnisdatei in Klaros-Testmanagement eingespielt. Liegen sowohl manuelle als auch automatisierte Testergebnisse vor, können sie gemeinsam ausgewertet werden.
В	
Benachrichtigungsereignis	Benachrichtigungsereignisse sind Auslöser für eine automatische E- Mail-Benachrichtigung an bestimmte Benutzer. Diese werden ver- schickt, sobald das vorgesehene Ereignis eingetreten ist.
	Mögliche Benachrichtigungsereignisse sind z.B. "Aufgabe zugewie- sen", "Benutzerkonto angelegt", "Testausführung fehlgeschlagen", "Benutzerkonto-Passwort geändert" und "Aufgabe bereit zur Ausfüh- rung".
Berichte (Reports)	Berichte geben Auskunft über den Stand der Testaktivitäten, den Zustand der Software, Hinweise auf kritische Komponenten und ermöglichen eine Vorausplanung.
	Dazu werden die Daten aus der Klaros-Testmanagement-Datenbank nach bestimmten Kriterien geordnet und in textuelle und grafische Übersichten aufbereitet. Beispiele für Berichte sind "Projektübersicht" und "Testaktivität" sowie Übersichten über Testfortschritt, Testergeb- nisse und Testläufe.
	Klaros-Testmanagement enthält zahlreiche grundlegende Berichte bereits vordefiniert. Individuelle Berichte können selbst erstellt werden. Alle Berichte sind in verschiedene Formate wie PDF oder Excel (auch zu weiteren Bearbeitung) exportierbar.
Berichtsvorlagen	In Berichtsvorlagen sind bereits grundlegende Bestandteile vordefi- niert und können vom Benutzer nach Bedarf beliebig angepasst wer- den.
	Bereits integrierte, vordefinierte Berichtsvorlagen in Klaros sind: "Testumgebungsbericht", "Testsystembericht", "Testsuitebericht" und "Testlaufbericht".
Bestanden (Bewertung)	Ein Test wird als Bestanden bezeichnet, wenn das tatsächliche mit dem vorausgesagten Ergebnis übereinstimmt. (ISTQB)
Bewertung	Als Bewertung wird das Ergebnis des ausgeführten Testfalls, Test- schritts oder der Testsuite bezeichnet. Mögliche Bewertungen sind "Bestanden", "Fehlschlag", "Fehler", "Unklar", "Übersprungen".
Bugzilla	Bugzilla ist ein Open Source-Issue Management System. Ausführliche Informationen sind auf der Bugzilla-Webseite zu finden.
	Klaros-Testmanagement verfügt über eine Anbindung an Bugzilla.

С	
Community Edition	Die kostenlose Edition von Klaros-Testmanagement. Sie hat einen ein- geschränkten Feature-Umfang, kann aber frei auch für kommerzielle Zwecke eingesetzt werden. Sie kann von der Webseite unter Download heruntergeladen werden.
Continuous Integration	Kontinuierliche Integration (Continuous Integration, CI) ist die automa- tisierte Integration von Codeänderungen von mehreren Mitwirkenden in ein einzelnes Softwareprojekt. Es ist eine wichtige DevOps-Best- Practice, die es Entwicklern ermöglicht, Codeänderungen regelmäßig in einem zentralen Repository zusammenzuführen, in dem dann Builds und Tests ausgeführt werden.
	Ein Plug-in für den Jenkins Continuous Integration Server überträgt automatisch die Testergebnisse eines Builds an Klaros Test Manage- ment, wo sie ausgewertet werden können.
D	
Dashboard	Ein Dashboard zeigt eine Sammlung von Informationen, die typischer- weise in Form von Diagrammen dargestellt werden.
	In Klaros-Testmanagement zeigen Dashboards die Auswertungen zu einem bestimmten Testprojekt an. Die angezeigten Diagramme kön- nen individuell zu einem Dashboard zusammengestellt werden. Jeder Benutzer kann beliebig viele Dashboards zusammenstellen.
	Dashboards können privat (nur für den Ersteller sichtbar) oder öffent- lich sein. Administratoren können ein Standard-Dashboard angelegen.
Datenbanksystem	Eine Datenbank ist eine Sammlung von Informationen, organisiert in in Wechselbeziehung stehenden Tabellen von Daten und Spezifikationen von Datenobjekten.
	Klaros-Testmanagement benötigt ein Datenbanksystem zur Speiche- rung der Objekte und wird mit einer vorkonfigurierten Apache-Der- by-Datenbank ausgeliefert. Weitere unterstützte Datenbanksysteme sind MariaDB, Microsoft SQLServer, MySQL und PostgresSQL.
Detailseite	Jedes Objekt (wie Testfälle, Anforderungen, Iterationen u.s.w.) besitzt eine eigene Detailseite mit verschiedenen weiteren Unteransichten. Diese variieren je nach Objekt und zeigen detaillierte Informationen zu beispielsweise <i>Eigenschaften</i> , <i>Anhängen</i> , <i>Änderungen</i> und <i>Ergebnissen</i> an.
Docker	Docker ist eine freie Software zum Isolieren von Anwendungen durch Containervirtualisierung. Die Container enthalten alle nötigen Pakete und lassen sich so leicht als Dateien transportieren und installieren.
	Klaros-Testmanagement kann ebenfalls als Container innerhalb einer Docker-basierten Umgebung betrieben werden Finsatzbereite Docker-

	Images für verschiedene Datenbanken sind bereits verfügbar. Eine ausführliche Dokumentation dazu ist unter GitHub zu finden.
Druckansichten	Alle Objekte lassen sich in deiner druckfreundlichen Darstellung aus- geben. Mit verschiedenen Parametern kann der Detailgrad des Aus- drucks festgelegt werden.
E	
Enterprise Edition	Die kommerzielle Edition mit vollem Support ist pro Benutzer lizenziert. Die initiale Installation enthält 3 Benutzer. Eine kostenlose 30-Tage- Demo kann unter Demo angefordert werden.
Ergebnisart	Der erwartete Ausgang des Tests: Positiv oder Negativ.
Ergebnis, erwartetes, voraus- gesagtes	Das Verhalten eines Systems oder einer Komponente unter festgeleg- ten Bedingungen, das durch die Spezifikation oder durch eine andere Quelle festgelegt ist. (ISTQB)
Evaluierungsart	Die Art der Testergebnisauswertung: Manuell oder Automatisiert.
F	
Fehler (Bewertung)	Ein Fehler bei der Testausführung tritt auf, wenn das System den Test nicht korrekt auszuführen kann. Er sollte nicht mit einem Fehlschlag verwechselt werden.
Fehlschlag (Bewertung)	Ein Test schlägt fehl, wenn das aktuelle Ergebnis nicht mit dem voraus- gesagten Ergebnis übereinstimmt (ISTQB). Er sollte nicht mit einem Fehler verwechselt werden.
	Klaros-Testmanagement unterscheidet die Bewertungen "Bestanden", "Fehler", "Fehlschlag", "Übersprungen" und "Unklar".
Funktionales Testentwurfs- verfahren	Ein Verfahren zur Herleitung und Auswahl von Testfällen, das auf der Analyse der funktionalen Spezifikation einer Softwarekomponen- te oder eines Softwaresystems basiert, ohne Bezug auf dessen innere Struktur (ISTQB), auch bekannt als Black-Box- und White-Box-Tests.
G	
Gast	Eine Benutzerrolle mit vollständigem, aber nur lesenden Zugriff auf alle Daten. Ein <i>Gast</i> kann Berichte in verschiedenen Formaten anzeigen und abspeichern. Diese Rolle ist z.B. für Projektmanager, Kunden oder Reviewer/Gutachter gedacht.
I	
ID	Die ID eines Objekts wird beim Erstellen automatisch angelegt und ist nicht änderbar. Alle IDs bestehen aus einem Kürzel für das jeweilige Objekt (z.B. TC für Testfall, ITR für Iteration) und einer aufsteigend ver- gebenen Zahl. <i>SEG00025</i> wäre also das Testsegment 00025.

Integration	Als Integration wird die Anbindung externer Systeme bezeichnet. Klaros-Testmanagement verfügt über Schnittstellen zur Integration mit Issue- und Anforderungsmanagementsystemen, Testautomatisie- rungs- und Lasttest-Werkzeugen sowie Continuous-Integration-Ser- vern. Außerdem existieren Schnittstellen zu Authentifizierungs- und E- Mail-Servern.
Issue/Problem	Als Issue bezeichnet man einen Teilaspekt, der zur Vervollständigung einer Arbeit erforderlich ist. Dies kann ein Bug sein, ein verlangtes Feature, eine Aufgabe, fehlende Dokumentation o.ä. Ein Issue wird in einem Issue Management System erfasst.
	Klaros-Testmanagement besitzt Schnittstellen zu mehreren Issue Management Systemen. Das heißt, gefundene Fehler können direkt an das angeschlossene IMS übergeben werden, ohne die Anwendung ver- lassen zu müssen.
Issue Management System	Ein Issue Management System (Issue Tracking System) ist eine Software, um Issues zu verwalten.
Issue Tracking System	Siehe Issue Management System.
Iteration	Eine Iteration ist ein vollständiger Entwicklungszyklus, der eine (inter- ne oder externe) Freigabe eines ausführbaren Produkts ergibt. Dieses Produkt ist eine Teilmenge des zu entwickelnden Endprodukts. Die Ent- wicklung schreitet von Iteration zu Iteration bis zum Endprodukt hin fort (ISTQB).
	Klaros-Testmanagement unterstützt agile Entwicklungsprinzipien und verwaltet Iterationen als eigene Objekte.
J	
Java	Damit Klaros-Testmanagement ausgeführt werden kann, ist eine Java 64-Bit-Java-11 Laufzeitumgebung erforderlich. Diese wird mit ausge- liefert und bei der Erstinstallation automatisch verwendet.
JavaScript	JavaScript ist eine in Webbrowsern eingesetzte Programmiersprache und wird für das Ausführen von Klaros-Testmanagement benötigt. Beim Einsatz von JavaScript-Blockern wie NoScript o.ä., muss das Ausführen von JavaScript als Ausnahme zugelassen werden.
JIRA	JIRA ist eine populäre Anwendung zur Fehlerverwaltung, Problembe- handlung und dem Projektmanagement der Firma Atlassian Software.
	Klaros-Testmanagement verfügt über eine Anbindung an JIRA.
К	
Kategorie	Die meisten Objekte können zur besseren Übersicht in Kategorien angeordnet werden. Das Gruppieren der Objekte (Anforderungen, Ite-

	rationen, Testsysteme, Testumgebungen, Testfälle oder Testsuiten) in Kategoriebäume kann nach selbst gewählten Kriterien erfolgen. Auch Mehrfachkategorisierungen sind möglich.
Konformität	Die Fähigkeit eines Softwareprodukts, anwendungsspezifische Nor- men oder Vereinbarungen oder gesetzliche Bestimmungen und ähnli- che Vorschriften zu erfüllen. [ISO 9126]
Konformitätsrate	In Klaros-Testmanagement zeigt die Konformitätsrate an, wie viele einer Anforderung zugeordnete Testfälle mit dem Ergebnis "Bestan- den" ausgeführt wurden.
L	
Layout-Vorlage	Die Layout-Vorlage ist neben dem Skript ein elementarer Bestandteil eines Berichtes. Sie beschreibt den Aufbau des erzeugten Berichtsdo- kuments in einer XML-basierten Beschreibungssprache. Layout-Vorla- gen können an individuelle Anforderungen angepasst werden.
Letzte Erfolgsrate (Berichts- diagramm)	Der Bericht "Letzte Erfolgsrate" zeigt die neuesten Ergebnisse der aus- geführten Testfälle für die ausgewählte Kombination aus Testsystem (eins oder mehrere) und Testumgebung (eine oder mehrere) auf dem Dashboard an.
Log-Panel	Das Log-Panel zeigt Statusmeldungen wie Warnungen oder Informa- tionen an. Standardmäßig wird nur die letzte Statusmeldung angezeigt, kann aber mit Klick auf das +-Icon geöffnet werden.
Μ	
Mantis	Mantis (MantisBT, Mantis Bug Tracker) ist eine Open Source-Anwen- dung zur Fehlerverwaltung und Problembehandlung in der Software- entwicklung.
	Klaros-Testmanagement verfügt über eine Anbindung an Mantis.
Ν	
Nachbedingung	Zustand des Testobjekts/Testsystems (und/oder der Umgebung), in dem sich das Testobjekt/Testsystem (oder die Umgebung) nach Aus- führung eines Testfalls oder einer Testsequenz befinden muss. (IST- QB)
0	
Objekt	Als Objekte werden in Klaros-Testmanagement Projekt, Testfall, Test- schritt, Testsegment, Testschrittergebnis, Testfallergebnis, Testsuite, Testsuiteergebnis, Testumgebung, Testsystem, Testlauf, Anforderung, Iteration und Aufgabe bezeichnet.

_
)
_

Projektübersicht (Berichts- diagramm)	Der Bericht "Projektübersicht" zeigt die Anzahl der ausgeführten/nicht ausgeführten Testfälle oder Testsuiten und die Anzahl der geplan- ten/ausgeführten Aufgaben auf dem Dashboard an.
Projekt	Ein Projekt ist der Grundbaustein in Klaros-Testmanagement und ent- hält alle anderen Objekte, die zur Testausführung notwendig sind.
Q	
QFTest	QF-Test QF-Test der Firma Quality First Software ist ein plattformüber- greifendes Werkzeug zur GUI-Testautomatisierung.
	Klaros-Testmanagement verfügt über eine Anbindung an QF-Test und kann die automatisiert erzeugten Testergebnisse zur weiteren Verar- beitung importieren.
R	
Redmine	Redmine ist eine Open Source Anwendung zur Fehlerverfolgung, Pro- blembehandlung und Projektmanagement.
	Klaros-Testmanagement verfügt über eine Anbindung an Redmine.
REST (Representational Sta- te Transfer)	Representational State Transfer (REST) ist ein Software-Architekturstil, der eine Teilmenge von HTTP verwendet. Er wird häufig verwendet, um Webdienste bereitzustellen.
	Klaros-Testmanagement bietet mehrere Integrationsschnittstellen basierend auf REST. Dazu gehören Lesezugriffe auf alle gespeicherten Informationen sowie Importschnittstellen für Testergebnisse, Testfäl- le und Anforderungen.
Revision	Wird eine größere Änderung an einem Objekt vorgenommen und die vorherige Version wird noch benötigt, sollte eine neue Revsion erstellt werden. Dies wird durch Vergabe einer neuen Revisionsnummer reali- siert. Nur ausgewählte Objekte wie z.B. Testfälle oder Anforderungen unterstützen diesen Mechanismus.
Rolle	Eine Benutzerrolle definiert die Rechte des Benutzers. Rollen können global und projektspezifisch festgelegt werden.
	In Klaros-Testmanagement gibt es die Benutzerrollen Administrator, Manager, Tester und Gast.
S	
Selenium	Selenium ist ein Webbrowser-Automatisierungswerkzeug und wird hauptsächlich zum Testen von Web-Apps verwendet.

	Selenium produziert JUnit-XML-konforme Testergebnisse, die in Kla- ros-Testmanagement zur weiteren Verarbeitung importiert werden können.
Skript	Das Skript ist neben der ein elementarer Bestandteil einer . Es extra- hiert die für den Bericht vorgesehenen Daten und liegt als Java-Klasse vor. Skripte können an individuelle Anforderungen angepasst werden.
Sprachen	Die angezeigte Sprache der Benutzer-Oberfläche wird über die vor- handenen Sprachdateien gesteuert. Diese werden im Verzeichnis .kla- ros/resources/messages gespeichert und unter Konfigurieren / Sys- tem / Sprachen aktiviert.
SUT	Siehe Testsystem, Zu testendes System, Prüfling, Testobjekt.
System-Zugang	Ein System-Zugang ist für automatisierte Vorgänge wie beispielsweise den Import von Testergebnissen reserviert. Eine Anmeldung über die Login-Seite ist hiermit nicht möglich.
Т	
Testaktivität (Berichtsdia- gramm)	Der Bericht "Testaktivität" zeigt die Testfallergebnisse für eine ausge- wählte Kombination aus einem oder mehreren Testsystemen und Test- umgebungen in einem ausgewählten Zeitraum als Histogramm auf dem Dashboard an.
Testart	Eine Gruppe von Testaktivitäten basierend auf spezifischen Testzie- len mit dem Zweck, eine Komponente oder ein System auf bestimm- te Eigenschaften zu testen (Funktional, Nichtfunktional, Strukturell, Regression, Retest).
Testausführung, Testdurch- führung	Der Prozess der Ausführung eines Tests für eine Komponente oder ein System, der Ist-Ergebnisse erzeugt. (ISTQB)
	Klaros-Testmanagement führt und protokolliert automatisch die Test- ausführung, siehe Testrunner.
Tester	Eine Benutzerrolle in Klaros-Testmanagement, die Objekte anzeigen sowie Testfälle und Testsuiten ausführen darf.
Testfall	Ein Testfall umfasst eine Menge von Eingabewerten, Ausführungsbe- dingungen, erwarteten Ergebnissen und Nachbedingungen, welche für ein bestimmtes Testsystem definiert wurden. Sie werden im Hinblick auf ein bestimmtes Ziel bzw. auf eine Testbedingung wie z.B. die Über- einstimmung mit spezifischen Anforderungen entwickelt.
Testfallergebnis	Das Ergebnis der Ausführung eines Tests und seiner Bewertung wie "Bestanden", "Fehler", "Fehlschlag" "Übersprungen" oder "Unklar".
Testfall-Priorität	Die Priorität des Testfalls: "Hoch", "Mittel" oder "Niedrig".
Testall-Status	Der Status des Testfalls: "Gesperrt", "Genehmigt", "Entwurf" oder "Aus- lassen".

Testfortschritt gramm)	(Berichtsdia-	Der Bericht "Testfortschritt" zeigt die Rate der ausgeführten gegenüber den definierten Testfällen eines Projekts für eine bestimmte Menge von Testumgebungen und Testsystemen auf dem Dashboard an.
Testhistorie gramm)	(Berichtsdia-	Der Bericht "Testhistorie" zeigt die Rate der definierten Testfälle gegen- über den ausgeführten Testfällen eines Projekts für eine oder mehrere Testumgebungen und ein oder mehrere Testsysteme in einer bestimm- ten Zeitperiode auf dem Dashboard an.
Testlauf		Die Ausführung eines oder mehrerer Testfälle mit einer bestimmten Version des Testobjekts/Testsystems. (ISTQB)
		Er bezieht sich immer auf genau ein Testsystem und genau eine Test- umgebung.
Testmanager		Die Person, die für das Management der Testaktivitäten, der Testres- sourcen und für die Bewertung des Testobjekts verantwortlich ist. Zu den Aufgaben gehören Anleitung, die Steuerung, die Verwaltung, Pla- nung und Regelung der Aktivitäten zur Bewertung des Testobjekts. (ISTQB)
		Eine Benutzerrolle in Klaros-Testmanagement, die vollständigen Lese- und Schreibzugriff auf alle Daten eines Projekts hat. Testmanager kön- nen alle Arten von Objekten anlegen, ändern und löschen sowie Pro- jekteinstellungen verwalten.
Testrunner		Klaros-Testmanagement enthält einen webbasierten Client für die Aus- führung manueller Tests. Dieser führt den Tester durch die Testschrit- te, gibt ihm die Möglichkeit Anmerkungen und Anhänge zu erstellen und protokolliert automatisch den Testverlauf und die Testergebnisse.
Testschritt		Ein Testschritt in Klaros-Testmanagement ist die kleinste Aktion, die innerhalb eines Testfalls abgearbeitet wird. Er enthält Ausführungsvor- bedingungen, erwartete Ergebnisse und Nachbedingungen der Ausfüh- rung für jede einzelne Aktion während der Testausführung.
Testsegment		Ein Testabschnitt ist in Klaros-Testmanagement eine definierte Folge von einzelnen Testschritten, die in einen Testfall eingefügt werden kön- nen. Ein Testsegment kann in mehreren Testfällen gleichzeitig verwen- det werden. Damit ist es möglich, ein Modulkonzept auf der Basis von Testschritten zu realisieren.
Testsuite		Eine Testsuite in Klaros-Testmanagement ist ein Satz an Testfällen, die zusammen in einer Gruppe ausgeführt werden können.
Testsystem, Z System, Prüflin	u testendes g, Testobjekt	Ein Testsystem repräsentiert in Klaros-Testmanagement die Software- version oder die Produktversion, die getestet werden soll. Je nach Branche sind dafür auch die Bezeichnungen zu testendes System, SUT, Prüfling oder Testobjekt gebräuchlich.
Testsystem gramm)	(Berichtsdia-	Der Bericht "Testsystem" zeigt die Erfolgs- und Fortschrittsrate von Testumgebungen unter einem einzigen Testsystem in einem Radardia-

	gramm auf dem Dashboard an. Sind weniger als drei Testumgebungen konfiguriert, wird stattdessen ein Balkendiagramm angezeigt.
Testumgebung	Eine Testumgebung repräsentiert die äußeren Umstände, die das Test- ergebnis beeinflussen können. Beispiele für Testumgebungen sind z.B. das Betriebssystem oder ein Applikations-Server (z.B. Tomcat 9 mit Ubuntu 20.04.2) oder auch Parameter wie Temperatur oder Geschwin- digkeit.
Testumgebung (Berichtsdia- gramm)	Der Bericht "Testumgebung - Übersicht" zeigt die Erfolgs- und Fort- schrittsrate von Testsystemen unter einer einzigen Testumgebung in einem Radardiagramm auf dem Dashboard an. Sind weniger als drei Testsysteme konfiguriert, wird stattdessen ein Balkendiagramm ange- zeigt.
Trac	Trac ist eine Open Source-Anwendung zur Fehlerverfolgung und Problembehandlung in der Softwareentwicklung. Klaros-Testmanage- ment verfügt über eine Anbindung an Trac.
U	
Übersichtsseite	Jeder Objekttyp besitzt eine Übersichtsseite. Dort werden die einzel- nen Objekte, wie z.B. Testfälle oder Iterationen, tabellarisch angezeigt. Neue Objekte werden ebenfalls hier erstellt.
Übersprungen (Bewertung)	Ein Testfall oder ein Testschritt wurden bei der Ausführung ausgelas- sen.
Unklar (Bewertung)	Ein Test wird als Unklar bezeichnet, wenn unklar ist, ob das tatsächliche mit dem vorausgesagten Ergebnis übereinstimmt.
Umgebungsvariablen	Umgebungsvariablen sind dynamisch benannte Werte, die das Verhal- ten von laufenden Prozessen auf einem Computer beeinflussen kön- nen. Sie sind Teil der Umgebung, in der ein Prozess läuft.
	Durch Setzen der Umgebungsvariablen KLAROS_HOME vor dem Start der Anwendung, kann der Speicherort des Klaros Home-Verzeichnis- ses an einen anderen Ort verschoben werden.
V	
Vorbedingung	Bedingungen an den Zustand des Testobjekts/Testsystems und seiner Umgebung, die vor der Durchführung eines Testfalls oder Testablaufs erfüllt sein müssen. (ISTQB)
W	
Webbrowser	Klaros-Testmanagement ist eine Webanwendung und läuft innerhalb eines Webbrowsers. Unterstützte Browser sind Apple Safari, Google Chrome, Microsoft Edge und Mozilla Firefox.

Y

YouTrack

YouTrack ist eine populäre Anwendung zur Fehlerverwaltung, Problembehandlung und dem Projektmanagement der Firma JetBrains.

Stichwortverzeichnis

A

Änderungshistorie, 70 Anforderung, 9 Anforderungen, 98 Anforderungsmanagement, 259 Anhänge, 135 Hochladen, 68 Referenzieren, 76 API, 299 Aufgabe, 10, 145 Anordnen, 147 aus Testfällen, 155 aus Testsuiten, 157 nach Benutzer, 159 Aufzählungswerte, 65 Ausführen, 161 Aufgaben, 161 Review, 163 Schritt-für-Schritt-Ansicht, 165 Tabellarische Ansicht, 167 Testfall, 163 Testsuiten, 170 Authentifizierung Single Sign-on, 4 Automatische Installations-Skripte, 31 Automatisierte Testfälle, 7

В

Benachrichtigung, 245 Benachrichtigungsschema, 243 Benutzer, 235 Benutzerdefinierte Eigenschaften, 64 Benutzerrollen, 11 Berichte Beispiel, 277 benutzerdefiniert, 271 Issues, 215 Letzte Erfolgsrate, 188 Parameter, 275 Projektübersicht, 188 Testaktivität. 189 Testfortschritt, 192 Testhistorie, 193 Testsystem - Übersicht, 190 Testumgebung - Übersicht, 191 Typen, 187

vordefiniert, 3 Vorlagen, 233, 272

С

CAS. 267 Classes KlarosAttachment, 375 KlarosCategoryNode, 376 KlarosCategoryTree, 377 KlarosConfiguration, 379 KlarosContext, 419 KlarosEnumValue, 380 KlarosExternalServer, 381 KlarosIssue, 382 KlarosIssueManagement, 384 KlarosIteration, 385 KlarosJob, 387 KlarosJobDependency, 388 KlarosJobTimeBlock, 389 KlarosJobUpdateAction, 390 KlarosLabeledObject, 391 KlarosNamedEntity, 392 KlarosProperty, 393 KlarosRepositoryEntity, 394 KlarosRequirement, 395 KlarosRequirementsManagement, 397 KlarosResult. 398 KlarosRevision, 399 KlarosSUTImplementation, 401 KlarosTestCase, 402 KlarosTestCaseResult, 404 KlarosTestCaseStep, 406 KlarosTestCaseStepResult, 407 KlarosTestEnvironment, 408 KlarosTestExecutable, 409 KlarosTestRun, 410 KlarosTestSegment, 411 KlarosTestStepContainer, 413 KlarosTestSuite, 414 KlarosTestSuiteResult, 415 KlarosUser, 416 Parameter, 426

D

Dashboard, 185 Standard, 186 Deinstallation, 46 Diagramm erstellen, 278 Docker, 4 Beschreibung, 14 Druckansichten, 61

Ε

E-Mail, 261 Ergebnis Einzelner Testfall, 206 Testfall, 204 Testlauf, 201 Export Excel, 297

F

Filtern, 56

Η

Hilfemenü, 54

I

Icons, 449 Aktionen, 449 Aktionen bei der Testausführung, 450 Ausführungsaktionen, 450 Ordnende Aktionen, 451 Bereiche, 449 Dokumentenformate, 452 Eigenschaften Ergebnisse, 452 Prioritäten, 452 Informationen, 451 Tabellenoperationen, 451 IKlarosAttachment, 302 IKlarosCategoryNode, 303 IKlarosCategoryTree, 304 IKlarosConfiguration, 306 IKlarosContext, 421 IKlarosEnumValue, 310 IKlarosExternalLink, 311 IKlarosExternalServer, 312 IKlarosIssue, 313 IKlarosIssueManagement, 317 **IKlarosIteration**, 318 IKlarosJob, 322 IKlarosJobDependency, 326 IKlarosJobTimeBlock, 327 IKlarosJobUpdateAction, 329 IKlarosLabeledObject, 330 IKlarosNamedEntity, 332 IKlarosPersistentObject, 333

IKlarosProperty, 333 IKlarosQueryFactory, 417 IKlarosRepositoryEntity, 334 IKlarosRequirement, 336 IKlarosRequirementsManagement, 339 IKlarosResult, 340 IKlarosRevision, 344 IKlarosSUTImplementation, 345 IKlarosTestCase, 348 IKlarosTestCaseResult, 354 IKlarosTestCaseStep, 356 IKlarosTestCaseStepResult, 357 IKlarosTestEnvironment, 360 IKlarosTestExecutable, 361 IKlarosTestRun, 362 IKlarosTestSegment, 366 IKlarosTestStepContainer, 369 IKlarosTestSuite, 370 IKlarosTestSuiteResult, 373 IKlarosUser, 374 Import Excel, 282 Format, Anforderungen, 286 Format, Testfälle, 282 QF-Test, 295 XML, 286 XML, Anforderungen, 289 Installation, 17 Integration, 85, 250 Anforderungsmanagement, 86, 259, 259 Issue Management, 85 Testautomatisierung, 291 Interfaces IKlarosAttachment, 302 IKlarosCategoryNode, 303 IKlarosCategoryTree, 304 IKlarosConfiguration, 306 IKlarosContext, 421 IKlarosEnumValue, 310 IKlarosExternalLink, 311 IKlarosExternalServer, 312 **IKlarosIssue**, 313 IKlarosIssueManagement, 317 IKlarosIteration, 318 IKlarosJob, 322 IKlarosJobDependency, 326 IKlarosJobTimeBlock, 327 IKlarosJobUpdateAction, 329 IKlarosLabeledObject, 330

IKlarosNamedEntity, 332 IKlarosPersistentObject, 333 IKlarosProperty, 333 IKlarosQueryFactory, 418 IKlarosRepositoryEntity, 334 IKlarosRequirement, 336 IKlarosRequirementsManagement, 339 IKlarosResult, 340 IKlarosRevision, 344 **IKlarosSUTImplementation**, 345 IKlarosTestCase, 348 IKlarosTestCaseResult, 354 IKlarosTestCaseStep, 356 IKlarosTestCaseStepResult, 357 IKlarosTestEnvironment, 360 IKlarosTestExecutable, 361 IKlarosTestRun, 362 IKlarosTestSegment, 366 IKlarosTestStepContainer, 369 IKlarosTestSuite, 370 IKlarosTestSuiteResult, 373 IKlarosUser, 374 Issues, 12 Iteration, 10 Iterationen, 90

К

Kategorien, 57 KlarosAttachment, 375 KlarosCategoryNode, 376 KlarosCategoryTree, 377 KlarosConfiguration, 379 KlarosContext, 419 KlarosEnumValue, 380 KlarosExternalServer, 381 Klaroslssue, 382 KlarosIssueManagement, 384 KlarosIteration, 385 KlarosJob, 386 KlarosJobDependency, 388 KlarosJobTimeBlock, 389 KlarosJobUpdateAction, 390 KlarosLabeledObject, 391 KlarosNamedEntity, 392 KlarosProperty, 393 KlarosRepositoryEntity, 394 KlarosRequirement, 395 KlarosRequirementsManagement, 397 KlarosResult, 397

KlarosRevision, 399 KlarosSUTImplementation, 401 KlarosTestCase, 402 KlarosTestCaseResult, 404 KlarosTestCaseStep, 406 KlarosTestCaseStepResult, 407 KlarosTestEnvironment, 408 KlarosTestExecutable, 409 KlarosTestRun, 410 KlarosTestSegment, 411 KlarosTestStepContainer, 412 KlarosTestSuite, 414 KlarosTestSuiteResult, 415 KlarosUser, 416 Konfigurieren, 232 Benachrichtigungsschema, 243 Benutzer, 235 Anlegen, 238 Löschen, 238 Benutzeroberfläche, 248 E-Mail, 261 Sprachen, 249 System, 242

L

LDAP, 264 Lesezeichen, 62 Lizenzmodell, 14 Login, 50

Μ

Massenaktionen, 60 Monitoring, 42

Ν

Nachbedingung, 166

0

Oberfläche, 248 Objekte, 12 Zuordnen, 59

Ρ

Parameter, 426 Parametertypen, 275 Performanztests unterstützte Tools, 3 Planen, 145 Projekt, 5 Projekte, 78

Zugriffsberechtigungen, 301

R

Rechte und Rollen, 301 Remote-API, 299 REST, 299 Backup, 298 Revisionen, 66 Rollen und Rechte, 301

S

Schnell-Navigation, 53 Schnellauswahl, 52 Session Timeout, 242 Single Sign-on-Authentifizierung, 267 Sortieren, 57 Sprachen, 48, 249 SSL-Unterstützung, 45 Standard-URL, 33 Suche, 52 Volltextsuche, 52 Synchronisierung, 289 System, 242 Systemvoraussetzungen, 14

Т

Tabellenfilter, 55 Testautomatisierung unterstützte Tools, 3 Testfall, 5 Testfälle, 124 Testlauf, 9 Testschritt, 6 Testschrittergebnis, 7 Testsegment, 7 Testsegmente, 119 Testsuite, 8 Testsuiteergebnis, 214 Testsuiten, 137 Testsystem, 8 Testsysteme, 111 Testumgebung, 8 Testumgebungen, 105

V

Vorbedingung, 166

Ζ

Zitate, 49