

# Testmanagementstudie 2009

Software Quality Lab

## Studie

- Version: **1.0**
- Stand vom: **11.01.2009 14:28:00**
- Status: **Freigegeben**
- Autor: **Johannes Hochrainer, Johannes Bergsmann**
  
- Dokument-ID: **TM-Studie-2009**
- Klassifikation: **Öffentlich**

# Inhalt

<b>DOKUMENTENINFORMATION .....</b>	<b>II</b>
Allgemeines .....	II
Dokument-Qualitätssicherung .....	II
Zweck und Inhalt dieses Dokuments .....	II
Copyright-Vermerk .....	II
<b>1. EINFÜHRUNG IN DIE TOOL-EVALUIERUNG .....</b>	<b>1</b>
1.1. Durchführung einer Tool-Evaluierung .....	2
1.2. Tool Evaluation Center .....	4
1.3. Kriterien für die Beurteilung der Tools in dieser Studie .....	6
<b>2. ÜBERBLICK .....</b>	<b>8</b>

## Dokumenteninformation

### Allgemeines

Zusammenhang mit anderen Dokumenten:	keiner
--------------------------------------	--------

### Dokument-Qualitätssicherung

Rolle	Name	Erreichbarkeit (Org.+Tel.+E-Mail)	Tätigkeit	Datum	Unterschrift
Autor	Johannes Bergsmann, Johannes Hochrainer		erstellt	2009-01-08	
Prüfer	Johannes Hochrainer		geprüft	2009-01-11	
Freigabe	Johannes Bergsmann	Johannes.bergsmann@ software-quality-lab.at	freigegeben	2009-01-11	

### Zweck und Inhalt dieses Dokuments

Dieses Dokument beinhaltet die Studie von Software Quality Lab über aktuelle Testmanagement- und Testautomatisierungswerkzeuge.

In dieser Studie sind all jene Werkzeuge enthalten, die im Tool Evaluation Center (TEC) von Software Quality Lab installiert sind. Weitere Tools bzw. Tool-Hersteller sind herzlich willkommen und können sich gerne mit uns in Verbindung setzen unter [info@software-quality-lab.at](mailto:info@software-quality-lab.at). Die Teilnahme im TEC und an der Studie ist für Hersteller kostenlos.

### Copyright-Vermerk

**Diese Studie wurde erstellt von Software Quality Lab GmbH und ist dessen Eigentum.**

Nachdruck, Vervielfältigung und Veröffentlichungen in jeglicher Form sind auch auszugsweise nur mit dem schriftlichen Einverständnis von Software Quality Lab GmbH erlaubt.

**Zitate bzw. Verweise auf die Studienergebnisse sind ausdrücklich erlaubt, wenn Sie folgende Referenz aufweisen:**

*Quelle: Software Quality Lab Testmanagementstudie 2009, [www.software-quality-lab.at](http://www.software-quality-lab.at)*

## 1. Einführung in die Tool-Evaluierung

Längst genügt es nicht mehr, irgendwelche Tools einzusetzen, um die Entwicklungsprozesse zu unterstützen.

**Sind die Werkzeuge nicht optimal auf die Prozesse abgestimmt, können die Folgekosten und die Folgewirkungen sehr hoch und möglicherweise sogar geschäftskritisch sein!**

Dabei sind nicht nur die Aufwände für Anpassung, Wartung und Updates zu berücksichtigen, sondern auch für Effizienzminderung oder gar Produktionsausfall. Deshalb müssen die Anforderungen systematisch erfasst und die betrachteten Produkte/Werkzeuge auf ihre Leistungsfähigkeit in Bezug auf die Anforderungen geprüft werden.

Nicht selten werden bei Produktvorführungen durch die Lieferanten bzw. Hersteller Funktionalitäten präsentiert, die nur teilweise, fehlerhaft oder noch gar nicht implementiert sind.

Außerdem wird kaum ein Anbieter auf alternative Produkte anderer Hersteller aufmerksam machen.

Bei der Tool-Evaluierung ist ein wesentlicher Erfolgsfaktor, wie man die passenden Tools evaluiert und auswählt.

Leider wird dieses doch oft strategisch wichtige Thema häufig recht halbherzig angegangen.

Die Gründe dafür sind vielfältig:

- keine Zeit, da gerade viele andere wichtige Projekte laufen
- keine qualifizierten Leute, die das beurteilen können
- zu wenig Marktüberblick über die gängigen Tools
- etc.

Um den Aufwand der Evaluierung zu minimieren, holt man sich oft auch einen „billigen“ *Studenten oder Praktikanten*, der das dann erledigt und auf dessen Ergebnissen die strategische Entscheidung basiert.

- Dass diesen jungen Leuten oft der Überblick, die notwendige praktische Erfahrung sowie auch die strategisch/unternehmerische Sicht fehlt, wird dabei nicht bedacht.
- Außerdem wird oft zu wenig Zeit in die Evaluierung investiert, sodass nach ein bisschen „Herumspielen“ diese wichtige Entscheidung auf Basis einer wackeligen Grundlage getroffen wird.

Die Tool-Entscheidung kann auch effizienter und auf Basis einer fundierten Evaluierung getroffen werden.





## 1.2. Tool Evaluation Center

Zur Durchführung der Tool-Evaluierung kann auch ein „Tool Evaluation Center“ (TEC) herangezogen werden. Dies ist eine typischerweise ständig betriebene Umgebung, in der die aktuellsten Tools vorkonfiguriert und mit Testbeispielen hinterlegt evaluiert werden.

**Das Tool Evaluation Center kann den Tool-Interessenten die Schritte 2 bis 6 der Tool-Evaluierung abnehmen und die Schritte 7 bis 9 wesentlich erleichtern. Es kann damit viel Zeit sparen, da diese Aktivitäten üblicherweise mehrere Tage oder Wochen an Arbeitszeit je Tool erfordern.**

Nachfolgend wird beispielhaft das TEC (Tool Evaluation Center von Software Quality Lab) vorgestellt, dem bei der Erstellung dieser Studie eine wichtige Rolle zukommt:

In der speziellen Umgebung des Tool Evaluation Centers (TEC) werden die wichtigsten Test-Tools und Application Life Cycle Werkzeuge für Kunden und Interessenten für Evaluierungen zur Verfügung gestellt. Das TEC unterstützt Kunden und Interessenten bei der neutralen Auswahl von Tools für Ihren Application Life Cycle.

Mit einem TEC kann eine herstellerunabhängige Unterstützung beim Auswahlprozess geleistet werden. Somit kann man sich für die Softwarewerkzeuge entscheiden, welche die Organisation und deren Prozesse optimal unterstützen.

Das TEC begleitet dabei beim ganzen Auswahlverfahren.

### Ziele des herstellerneutralen Tool Evaluation Centers sind:

- ⇒ Kunden und Interessenten wird der direkte Vergleich von Produkten und Tools unterschiedlicher Anbieter ermöglicht.
- ⇒ Das mühsame Produkthanfordern, probeweise Installieren, hektische Ausprobieren, ... kann vermeiden werden und eine objektive Evaluierung der Produkte wird ermöglicht.
- ⇒ Durch sehr breit angelegte, bereits vorbereitete Evaluierungsbeispiele wird ein praxisgerechter Vergleich der Produkte ermöglicht.
- ⇒ Durch die Betreuung durch einen Tool-Spezialisten wird sicher gestellt, dass die Produktevaluierung zielgerichtet und effizient verläuft.

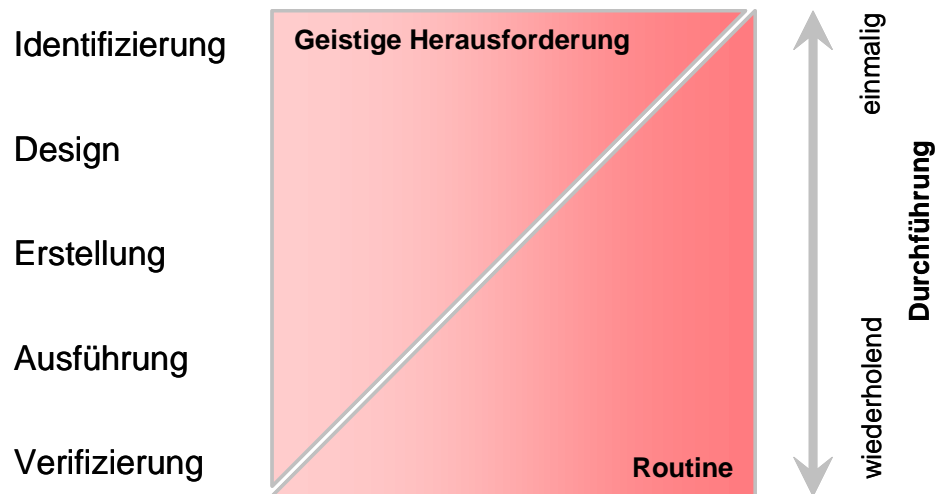
### Nutzen für die Kunden und Interessenten

- ✓ Es kann viel Zeitaufwand durch Wegfall der sonst üblichen Vorbereitungen gespart werden.
- ✓ Es können auf einfache Weise mehrere Produkte im direkten Vergleich evaluiert werden.
- ✓ Viele Interessenten wollen nicht vom Vertrieb des Herstellers „überredet“ werden, sondern wollen sich Produkte in Ruhe selbst ansehen und evaluieren.
- ✓ Es stehen von allen relevanten Produkten bereits die vorkonfigurierten und mit Beispielen versehenen Vollversionen der Produkte zur Verfügung.



## 1.3. Kriterien für die Beurteilung der Tools in dieser Studie

Nachfolgend sind die typischen Testaktivitäten dargestellt<sup>1</sup>:



Die Testaktivitäten Identifizierung, Design, Erstellung, Ausführung und Verifizierung bildeten eine wesentliche Grundlage für die Bewertung der Tools in dieser Studie.

Sie sind Teil des nachfolgenden Bewertungsschemas:

1. Allgemein (Hersteller, Geschichte, Grobbeschreibung, Einsatzgebiete, ...)
  - a. Hersteller
  - b. Produktportfolio
  - c. Programmteile
  - d. Integrationen
2. Unterstützte Testaktivitäten
  - a. Planung und Steuerung
    - Erstellung eines Testkonzepts
    - Planung von Ressourcen (Zeit, Mitarbeiter, Geräte, ...)
    - Überprüfung des Testfortschritts
    - Risikoanalyse
    - Planung von Reviews
    - Traceability von Anforderungen zu Testfällen

<sup>1</sup> Fewster, M., and Graham, D. Software Test Automation. Addison-Wesley Professional, 1999.

b. Analyse und Design

- Art der Testfallbeschreibung (formatierter Text, Data-Driven Testing, Keyword-Driven Testing, ...)
- Trennung logischer- von konkreten Testfällen
- Strukturierung und Kategorisierung der Testfälle
- Vor- und Nachbedingung
- Zusatzinformationen (Attribute, ...)
- Techniken zur Herleitung von Testfällen (Äquivalenzklassen, Grenzwertanalyse, ...)

c. Implementierung

- Wie erfolgt die Testautomatisierung
- Unterstützte Testautomatisierungswerkzeuge

d. Durchführung

- Bildung von Testsuiten (Filter, ...)
- Arten der Testausführung (manuell, automatisch, gemischt)
- Planung der Ausführung (Zeit, Ort, Priorisierung)
- Steuerung der Ausführung auf Remote-Clients

e. Auswertung und Bericht

- Anbindung von Fehlermanagementwerkzeugen
- Metriken
- Berichtgenerierung
- Vergleich von Testdurchläufen

3. Administration

- a. Benutzerverwaltung
- b. Versionsmanagement

4. Zusammenfassung

Diese Kriterien sind Basis für die Analyse der Tools in dieser Studie, als auch für die direkte Gegenüberstellung in der Bewertungsmatrix (siehe Kapitel 2).

## 2. Überblick

Auf den folgenden Seiten finden Sie eine Gegenüberstellung der in der Studie enthaltenen Testmanagementwerkzeuge.

Der Bewertungsschlüssel ist in weiterer Folge ebenfalls dargestellt.

In der Gesamt-Studie wird jedes der in der Übersicht angegebenen Werkzeug in weiterer Folge auf ca. 20-30 Seiten detailliert betrachtet und bewertet.

Zahlreiche Abbildungen und Erklärungen geben Ihnen wertvolle Informationen darüber, ob ein Testmanagementwerkzeug für Ihre Anforderungen geeignet ist.

Wenn Sie an der gesamten Studie oder auch einzelnen Werkzeugen interessiert sind, nehmen Sie bitte mit uns Kontakt auf unter [info@software-quality-lab.at](mailto:info@software-quality-lab.at).

Firma	Borland	HP	IBM Rational	Imbus	Mozilla	Orcanos	Polarion	Siemens	TestLink
<b>Produkt</b>	SilkCentral	Quality Center	Clear Quest TM	TestBench	Testopia	QPack	Polarion	SITEMPPO	TestLink
<b>Lizenz</b>	Floating / Seat	Floating / Seat	Floating / Seat	Floating	Frei (Netscape Public License)		Floating	Floating / Seat; Miet- oder Kaufmodell	Frei (GPL license)
<b>Datenbank</b>	Oracle, Microsoft SQL Server	Oracle, Microsoft SQL Server	DB2, Microsoft Access, Microsoft SQL Server, Oracle	Oracle, Microsoft SQL Server	MySQL	Microsoft SQL Server	n.v. (SVN)	Oracle, Microsoft Access und SQL Server, PostgreSQL	beliebig
<b>Client</b>	Web	Web	Web und Eclipse Rich Client	Java Client	Web	Web (Internet Explorer)	Web	Java Rich Client	Web
<b>Integrationen</b>									
<b>Anforderungsmanagement</b>	Borland CaliberRM IBM Rational Requisite Pro IBM Telelogic DOORS Microsoft Word und Excel	Borland CaliberRM IBM RequisitePro Microsoft Word/Excel Telelogic DOORS ...	Rational RequisitePro	Borland CaliberRM IBM Rational RequisitePro MS Excel TCP IrqA Telelogic DOORS	n.v.	n.v.	n.v.	Telelogic DOORS	Telelogic DOORS
<b>Testautomatisierung</b>	Borland Silk Performer Borland Silk Test HP QuickTest Professional HP WinRunner	HP WinRunner HP LoadRunner HP QuickTest Professional ...	Rational Functional Tester Rational Manual Tester Rational Robot Rational Performance Tester TPTP	LogicaCMG Contest net FrameWork National Instruments TestStand S.E.A. TestMaster Borland Silk Test HP WinRunner HP QuickTest Professional IBM Rational XDE / Functional Tester IBM Rational Robot qftestJUI	n.v.	n.v.	n.v.	Borland Silk Test Compuware TestPartner HP WinRunner HP QuickTest Professional	n.v.
<b>Fehlermanagement</b>	Borland SilkCentral Issue Manager (integriert in SilkCentral) Borland StarTeam IBM Rational ClearQuest Bugzilla		Rational Produkte	Atlassian JIRA Bugzilla IBM Rational ClearQuest Imbus Fehlerdatenbank MS Excel	Bugzilla	n.v.	n.v.	Bugzilla Telelogic Change	Bugzilla Eventum JIRA Mantis, TestPlus Track
<b>Offene API</b>	Ja (Java)	Ja	Ja	Ja (erweiterbar mit Plug-in)	Ja	Ja	Ja	Wenige	Ja
<b>Funktionsumfang</b>									
<b>Anforderungsmanagement</b>	Ja	Ja	Ja	Nein	Nein	Ja	Ja	Ja	Ja
<b>Testmanagement</b>	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Ja
<b>Fehlermanagement</b>	Ja	Ja	Ja	Ja	Ja	Ja	Ja	Nein	Nein

Firma	Borland	HP	IBM Rational	Imbus	Mozilla	Orcanos	Polarion	Siemens	TestLink
Produkt	SilkCentral	Quality Center	Clear Quest TM	TestBench	Testopia	QPack	Polarion	SITEMPPD	TestLink
<b>Unterstützte Testaktivitäten</b>									
<b>Planung und Steuerung</b>									
Anforderungsbasiertest Testen	+	+	o	o	n.v.	++	++	+	o
Überprüfen des Testfortschritts	++	++	+	+	++	++	++	+	++
Risikoanalyse	+	+	-	-	n.v.	++	+	-	-
Planung von Reviews	o	o	+	+	o	-	++	o	-
Testprozess	++	++	++	o	n.v.	++	++	o	o
Benutzerdefinierte Felder	+	+	++	o	n.v.	++	++	o	+
Filter	++	++	++	++	++	++	++	++	o
Traceability	++	++	++	+	-	++	++	o	o
<b>Analyse und Design</b>									
Testfallerstellung aus Anforderungen	++	++	o	o	-	o	o	o	++
Beschreibung manueller Tests	Tabelle	Tabelle	Text	Keyword-Driven / Text	Text	Text	Text	Tabelle	Text
Formattierungsmöglichkeiten manueller Vor- und Nachbedingung	++	++	++	++/-	++	++	+	o	++
Datengetriebene manuelle Tests	-	-	+	+	o	n.v.	n.v.	++	n.v.
Implementierung	n.v.	n.v.	++	(++)	n.v.	n.v.	n.v.	n.v.	n.v.
Unterstützte Programmierframeworks	JUnit, NUnit, Windows Scripting Host	VAPI-XP-Test (VBScript, JavaScript, Perl oder Python)	Eclipse TPTP (Manual-, JUnit-, URL-Test)	n.v.	n.v.	n.v.	(Ant, Maven,...)	JUnit	n.v.
Unterstützte Testautomatisierungswerkzeuge	o	o (aber: Drittanbieter)	o	++	n.v.	n.v.	n.v.	++	n.v.
Anbindung eigener Werkzeuge	+	+	+	++	+	+	+	++	+
<b>Ausführung</b>									
Bildung von Testsuiten	++	++	++	++	++	+	n.v.	++	(++)
Abhängigkeiten	o	+	n.v.	n.v.	-	-	-	n.v.	n.v.
Mischen von Testarten (manuell)	++	++	++	++	(++)	(++)	(++)	++	(++)
Durchführungsdauer	++	o	n.v.	+	+	-	-	n.v.	n.v.
Durchführungszeit	++	o	o	o	-	o	n.v.	n.v.	o
Durchführungsart	++	++	+	++	-	-	-	+	-
Assistent für manuelle Tests	++	++	++	++	o	o	n.v.	+	-
<b>Auswertung und Bericht</b>									
Diagramme	++	++	++	+	o	++	++	+	-
Berichte	++	++	++	+	o	++	++	+	-
Erstellung eigener Berichte	++	++	++	+	-	++	++	+	-
Live Analyse (Dashboard)	++	++	n.v.	n.v.	o	++	++	n.v.	-
<b>Administration</b>									
<b>Benutzerverwaltung</b>									
LDAP-Integration möglich	++	++	++	n.v.	++	n.v.	++	n.v. (aber: ab 5.6)	n.v.
Rollenmodell	++	++	++	o	n.v.	++	++	++	++
Granularität des Rechtemodells	++	++	++	-	o	++	++	++	+
<b>Versionsmanagement</b>									
Integriertes Versionsmanagementsystem	n.v.	n.v.	n.v.	++	++	++	++	++	n.v.
Umfang der Versionierung	n.v.	n.v.	n.v.	++	-	++	++	++	n.v.
Externes Versionsmanagement für Testautomatisierung integrierbar	++	++	++	n.v.	n.v.	++	++	n.v.	n.v.
Historisierungsfunktion	++	++	++	n.v.	++	++	++	++	n.v.
Sperrmechanismus für Mehrbenutzerbetrieb	n.v.	n.v.	n.v.	++	n.v.	++	n.v.	++	n.v.

## Bewertungsschlüssel für Testmanagementwerkzeuge

Bewertung	++	+	o	-	n.v.
Beschreibung	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
<b>Unterstützte Testaktivitäten</b>					
<b>Planung und Steuerung</b>					
Anforderungsbasiertes Testen	Benachrichtigung, wenn sich Anforderungen ändern	Analyse der Anforderungsabdeckung	Anforderungen können im Werkzeug definiert oder importiert werden	Angabe von Anforderungen ohne direkte Verbindung	nicht vorhanden
Überprüfen des Testfortschritts	Eigene Sicht für Projektmanager (z. B. Dashboard)	Vordefinierte Berichte	Vergleich geplanter mit tatsächlich ausgeführten Tests	Vergleich aller ausgeführten Tests mit allen vorhandenen Tests	nicht vorhanden
Risikoanalyse	Berichte und Prozess sind vorhanden	Anforderungen können bewertet werden und dienen als Kriterium für Testfallerstellung	Testfälle können bewertet werden	Kann mit benutzerdefinierten Feldern nachgebildet werden	nicht vorhanden
Planung von Reviews	Automatische Benachrichtigungsmöglichkeit vorhanden	Eigenes Eingabefeld für Auswahl des Reviewers, Reviewer erhält seine Aufgaben mit Filter	Statuswerte für Review sind vorhanden.	Kann mit benutzerdefinierten Feldern nachgebildet werden	nicht vorhanden
Testprozess	Prozess kann angepasst werden	Benachrichtigung, wenn Benutzer Aufgabe erfüllen soll	Mit vorgefertigten Filtern kann Benutzer Aufgaben finden	Benutzerdefinierte Felder sind vorhanden. Mit Filtern kann Benutzer Aufgaben herausfinden.	nicht vorhanden
Benutzerdefinierte Felder	Benachrichtigungen bei Änderung des Wertes	Für Anforderungen und Testfälle	Für Testfälle	Nur Textfeld	nicht vorhanden
Filter	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden

<b>Traceability</b>	Benachrichtigung bei Änderung eines verknüpften Elements	Status externer Elemente wird in Auswertungen berücksichtigt	Alle Elemente im System sind bidirektional verknüpft	Link zu Bericht im externen Fehlermanagementsystem	nicht vorhanden
<b>Analyse und Design</b>					
<b>Testfallerstellung aus Anforderungen</b>	Bei Anforderungen ist Abdeckungsgrad ersichtlich	Testfälle können direkt von den Anforderungen aus erzeugt werden	Zuerst Erstellung eines Testfalls, dem dann Anforderungen zugewiesen werden können	Keine direkte Referenz zu Anforderung. Es kann aber eine ID oder der Name eines Anforderungsdokuments eingegeben werden.	nicht vorhanden
<b>Formatierungsmöglichkeiten manueller Tests</b>	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
<b>Vor- und Nachbedingung</b>	Können von übergeordneten Elementen weitervererbt werden	Wieder verwendbar	Als separate Felder vorhanden	Mit benutzerdefinierten Feldern machbar	nicht vorhanden
<b>Datengetriebene manuelle Tests</b>	wird unterstützt				nicht vorhanden
<b>Implementierung</b>					
<b>Unterstützte Testautomatisierungswerkzeuge</b>	Mindestens 3 Fremdhersteller	Mindestens ein Fremdhersteller	Nur Werkzeuge aus vom selben Hersteller	Aufruf einer ausführbaren Datei	nicht vorhanden
<b>Anbindung eigener Werkzeuge</b>	Framework vorhanden	gut dokumentiert	dokumentiert	rudimentär	nicht vorhanden
<b>Ausführung</b>					
<b>Bildung von Testsuiten</b>	Auf Basis vorhergegangener Testdurchläufe (z. B. alle negativen Tests + nicht durchgeführten Tests)	Zusammenstellung mit komplexeren Bedingungen (z. B. Risiko = hoch + Systemtest = ja)	Zusammenstellung mit einfachen Bedingungen (z. B. Risiko = hoch)	Manuelle Zusammenstellung	nicht vorhanden
<b>Abhängigkeiten</b>	Zwischen Testfällen und Testsuiten	Zwischen Testfällen	Zwischen Testpaketen	Definition möglich, haben aber keine Auswirkung auf Testdurchführung	nicht vorhanden

Mischen von Testarten (manuell /automatisiert)	Möglich				nicht vorhanden
Durchführungsdauer	Spezielle Berichte für Vergleich der Durchführungszeiten.	Automatische Aufzeichnung der Durchführungszeiten	Hochrechnen der geplanten Zeit auf Testordner	Angabe der geplanten Zeit	nicht vorhanden
Durchführungszeit	Angabe von Perioden mit Ausnahmen	Angabe von Perioden	Angabe des Datums	Mit Meilensteinen	nicht vorhanden
Durchführungsart	Durchführungsserver steuert gleichzeitige Testdurchführung auf mehreren Computern	Online auf einen Computer	Offline auf anderen Computern	Nur manuelle Testfälle	nicht vorhanden
Assistent für manuelle Tests	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
<b>Auswertung und Bericht</b>					
Diagramme	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
Berichte	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
Erstellung eigener Berichte	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
Live Analyse (Dashboard)	außergewöhnlich	gut	ausreichend	rudimentär	nicht vorhanden
<b>Administration</b>					
<b>Benutzerverwaltung</b>					
LDAP-Integration möglich	Ja				Nein
Rollenmodell	Rollen können selbst definiert werden	Festes Rollenmodell, wobei Rechte angepasst werden können	Festes Rollenmodell	Es gibt keine Rollen	nicht vorhanden
Granularität des Rechtemodells	sehr fein	fein	ausreichend	grob	nicht vorhanden
<b>Versionsmanagement</b>					
Integriertes Versionsmanagementsystem	Ja				Nein
Umfang der Versionierung	Alle Testobjekte, für jede Testdurchführung kann Zustand des Testprojekts wiederhergestellt werden	Alle Testobjekte	Manche Testobjekte	Versionierung nur von Teilen einzelner Testobjekte (z. B. nur Testschritte)	nicht vorhanden

Externes Versionsmanagement für Testautomatisierung integrierbar	Ja				Nein
Historienfunktion	Ja				Nein
Sperrmechanismus für Mehrbenutzerbetrieb	Ja				Nein