

Ausgabe 2004 / 3

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

Software-Testen

weitere in dieser Ausgabe ...

- ⇒ Software-Testen - ein Überblick
- ⇒ Literatur, Links, Zitate

Kurzdefinition / Glossar ...

- ⇒ **Konstruktives Qualitätsmanagement** sind Methoden, Sprachen, Werkzeuge, Richtlinien, Standards, Checklisten, Vorgehensmodelle und Prozesse, die dafür sorgen, dass das entstehende Produkt bzw. der Erstellungsprozess a priori bestimmte Eigenschaften besitzt. Konstruktive Maßnahmen gliedern sich grundsätzlich in technische und organisatorische Maßnahmen.
- ⇒ **Analytisches Qualitätsmanagement** sind diagnostische Maßnahmen, durch die das existierende Qualitätsniveau gemessen wird und Ausmaß und Ort der Defekte festgestellt werden. Sie bringen in das Produkt bzw. den Entwicklungsprozess keine Qualität per se. Das Ziel ist die Prüfung und Bewertung der Qualität. *nach Balzert*
- ⇒ **Testen** ist das Ausführen eines Programms, um Fehler zu finden. Testen zählt zu analytischen Qualitätsmanagement-Methoden. *nach Liggesmeyer*



Testen als Verantwortung und als Sicherheit für Lieferant und Kunde!

Die Lieferanten und Software-Hersteller sehen sich einem enormen Preisdruck ausgesetzt und müssen natürlich sehr knapp kalkulieren.

Themen, die nicht vertraglich eindeutig festgeschrieben werden (wie ausführliches nachvollziehbares Testen, gute Dokumentation oder Qualität der Produkte, ...) sind hier oftmals die Opfer im Preiskampf.

Die Kunden wiederum verlassen sich auf Verträge und die Gewährleistung.

Diese Sicherheit ist jedoch oft sehr trügerisch, da in die Verträge und Spezifikationen zumeist ebenfalls wenig investiert wird und damit die Qualität der vertraglichen Basis in vielen Projekten äußerst schlecht ist.

Um Probleme und Streitfälle möglichst zu vermeiden, ist es sowohl für den Kunden als auch für den Lieferanten ratsam, analytisches QM (Testen, Reviews, ...) zu betreiben, um nicht erst im Betrieb die Fehler zu finden und teuer beheben zu müssen. Im Idealfall sollte auch der Kunde intensiv in diese Aktivitäten involviert werden und entsprechende Mitarbeiter bereit stellen.

Auch der Auftraggeber hat Verantwortung zu tragen und muss dem Lieferanten klar mitteilen, was er als Ergebnis erwartet.

Wenn er dies schon nicht in Form einer guten Spezifikation tut, so sollte er zumindest noch vor der Inbetriebnahme intensiv an der Test- und Abnahmephase teilnehmen und Rückmeldungen geben.

Solange jemand nicht bereit ist, entsprechend Zeit und finanzielle Mittel zur Verfügung zu stellen, wird er meist auch keine qualitativ hochwertige Ergebnisse erhalten.

Dipl.-Ing. Johannes Bergmann

allgemein gerichtlich beideter und
zertifizierter Sachverständiger für Informatik

Analytisches QM: Nur sinnvoll im Rahmen einer Gesamt-QM-Strategie!

"Wir beschäftigen genau so viele Tester wie Entwickler. Wenn wir ein neues Windows-Release vorbereiten, geht über die Hälfte des Budgets allein in die Qualitätskontrolle!"
Bill Gates-Zitat aus der Infomationweek 12/02

Wenn man in einem Unternehmen nur den Bereich des analytischen QM betrachtet (ohne den weiteren Kontext) gibt es verschiedene Interpretationsmöglichkeiten:

- bei einem hohen Anteil von analytische QM könnte das Unternehmen ein sehr qualitätsorientiertes Unternehmen sein und deshalb ein umfassendes analytisches Qualitätsmanagement betreiben oder
- die Entwickler des Unternehmens machen so viele Fehler, dass dadurch ein entsprechend großer Testaufwand notwendig wird.

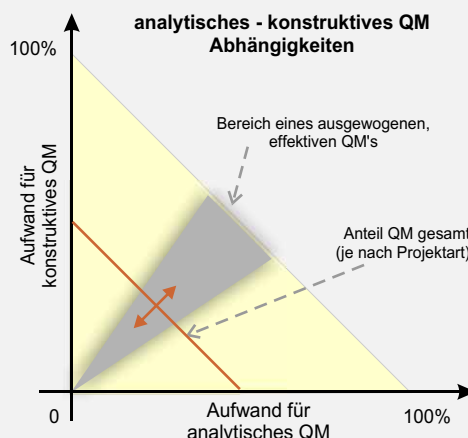
Daraus wird ersichtlich, dass der analytischen QM-Teil sinnvollerweise immer im Kontext des konstruktiven Qualitätsmanagements zu betrachten ist.

Zumeist ist jedoch der Bereich des konstruktiven Qualitätsmanagements der schwieriger umzusetzende QM-Bereich. Dadurch wird in vielen Unternehmen das Augenmerk auf den Bereich des analytischen Qualitätsmanagements und hier vorwiegend auf das Testen gelegt. Dies geschieht vor allem dann, wenn QM nicht von innen heraus angestrebt wird, sondern der Kundendruck oder mögliche Marketing-Effekte die primäre Motivation sind.

Dadurch kommt es oft zu Situationen, dass Unternehmen große Aufwände in Testtools und den Aufbau von Testteams investieren, die jedoch dann durch Mängel im Bereich des konstruktiven QM (vor allem in den frühen Phasen des Projekts) nicht effizient arbeiten können.

Die im analytischen QM-Bereich getätigten Investitionen gehen dann zum großen Teil wieder verloren.

Für ein effizientes Qualitätsmanagement sollten QM-Maßnahmen sowohl im konstruktiven wie auch im analytischen QM-Bereich gleichermaßen umgesetzt werden. Die QM-Gesamtstrategie sollte dies noch durch zusätzliche Maßnahmen im personellen, kulturellen und im Management-Bereich ergänzen.



Der Quality-Knowledgeletter ist ein periodisches Informationsmedium von Software Quality Lab und dessen Partnern mit den Schwerpunkten IT-Qualitätsmanagement, Projekt- und Prozess-Management.

Inhalt: fachliche Beiträge und Schwerpunktthemen, Vorstellung neuer Produkte und Leistungen, neue wissenschaftliche Erkenntnisse, ...
Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.at.
Weitere Infos zu diesem und anderen Themen finden Sie auf <http://www.software-quality-lab.at>.

Software-Testen

Ein Überblick von Dipl.-Ing. Johannes Bergmann

Software-Systeme werden immer komplexer und umfangreicher. Dabei treten vermehrt signifikante Qualitätsprobleme auf. Softwaresysteme ohne Mängel sind derzeit praktisch kaum zu finden. Da jedoch Softwaresysteme in immer mehr Bereichen des Lebens eine maßgebliche Rolle spielen, werden die Auswirkungen von Fehlern auch zunehmend größer und nehmen in manchen Fällen schon katastrophale Ausmaße an.

Testen ist ein Ansatz, um die Fehler und das damit verbundene Risiko in Softwaresystemen zu reduzieren. Tatsache ist jedoch leider, dass das Testen in vielen Fällen vernachlässigt wird.

Dieser Artikel soll die Wichtigkeit des Testens darstellen und zusätzlich einen kurzen Überblick über die Vorgehensweisen und Methodiken des Testens geben.

➔ Vertragliche, finanzielle und rechtliche Aspekte

Eine Vernachlässigung des Testens kann für den Softwarehersteller im Schadensfall erhebliche rechtliche und finanzielle Konsequenzen haben. Wenn Schäden durch fehlerhafte Software verursacht werden, wird im Nachhinein oftmals der Schuldige gesucht und teilweise wird auch gerichtlich vorgegangen.

Der Sachverständige des Gerichts wird dann natürlich auch den Testprozess dieser Software genauer untersuchen. Wenn hier fahrlässig gehandelt wurde („wenn die gewöhnlich erforderliche Sorgfalt außer acht gelassen wird“), ist das mit einem erheblichen Haftungs- und Prozessrisiko für den Schuldigen verbunden.

Und wenn, wie das oft der Fall ist, schon einfache Standardabläufe in einer Software fehlerbehaftet sind, woraus angenommen werden kann, dass der Hersteller die Software nicht oder kaum sorgfältig getestet hat, kann zumeist von einer fahrlässigen Handlungsweise des Software-Herstellers ausgegangen werden.

Im einfachsten Fall hat damit der Kunde oft die Möglichkeit, Preisabschläge auf den vereinbarten Kaufpreis geltend zu machen. Dabei können in vielen Fällen auch interne Zeitaufwände oder Zeitverluste, die nachweislich durch die Fehler in der Software entstehen, eingerechnet werden.

Derzeit machen jedoch viele Kunden noch nicht Gebrauch von dieser Möglichkeit und tragen diese internen Aufwände selbst. Daher sehen es viele Software-Hersteller immer noch als ‚Kavaliersdelikt‘, wenn Sie oft auch bewusst - fehlerhafte Software an die Kunden ausliefern und so den Testaufwand großteils an die Kunden abwälzen.

Mit der zunehmenden Industrialisierung der IT wird in Zukunft jedoch auch die Qualitätserwartung der Kunden steigen und der Testprozess auf Herstellerseite ein größeres Gewicht erhalten.

Doch auch der Kunde hat entsprechende Testverpflichtungen (primär mit Auswirkungen auf die Gewährleistungsverpflichtungen) und muss die Software in einer angemessenen Zeit nach der Lieferung überprüfen (testen) und abnehmen bzw. etwaige Mängel rügen.

Bei der Beauftragung von individuellen Softwarelösungen hat der Kunde darüber hinaus noch weitere Verpflichtungen, auf die jedoch im Rahmen dieses Artikels nicht weiter eingegangen wird.

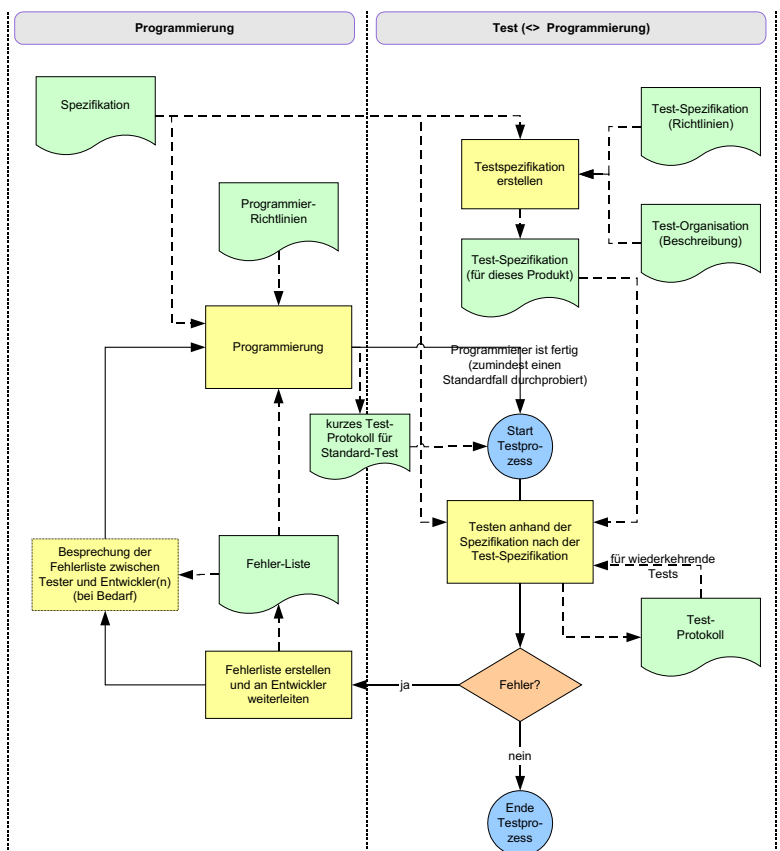
➔ Auswirkungen des Testens

Testen ist daher eine wesentliche und von jedem sorgfältig handelnden Software-Hersteller wahrzunehmende Verpflichtung und hat folgende wichtige Dimensionen:

- 1.) Fachlich/Technisch: Reduktion bzw. Vermeidung von Fehlern in der Software
- 2.) Rechtlich: Reduktion von Haftungs- und Prozessrisiken bei durch fehlerhafte Software verursachten Schäden
- 3.) Finanziell/Marktorientiert: Vermeidung von Preisabschlägen oder Schadenersatzforderungen bei fehlerbehafteter Software sowie Erhöhung der Kundenzufriedenheit und bessere Marktchancen durch qualitativ hochwertige Software-Produkte

➔ Der Testablauf

Nachfolgend wird ein einfacher Testablauf im Rahmen der Software-Programmierung dargestellt:



Beispielhafte Darstellung eines einfachen Test-Prozesses
Quelle: Bergmann

Fortsetzung auf nächster Seite >>>

Fortsetzung - Software-Testen >>>

➔ **Beschreibung eines einfachen Testprozesses**

Der Programmierer entwickelt die Software unter Verwendung der Spezifikation.

Zusätzlich werden noch unternehmenseinheitliche Programmierrichtlinien als Grundlage verwendet. Die Einhaltung der Programmierrichtlinien kann in Form von systematischen Code-Reviews geprüft werden.

Der Programmierer testet zwingend die Standard-Fälle für das erstellte Programm und dokumentiert dies in einem kurzen Test-Protokoll.

Anschließend wird das Programm inkl. Test-Protokoll an den Tester übergeben. Der Tester ist nicht identisch mit der Person des Entwicklers, sondern immer eine andere Person!

Spätestens zum Zeitpunkt der Übergabe des Programms an den Tester (vorzugsweise schon früher z.B. nach Fertigstellung der Spezifikation) erstellt der Tester (bzw. der Testmanager) auf Basis von entsprechenden Vorlagen und Richtlinien die Testspezifikation für dieses Programm. Dabei werden die allgemeinen Vorgaben aus der Test-Organisation ebenfalls berücksichtigt.

Der Tester testet anhand der entsprechenden Test-Spezifikation und Testfall-Vorgaben und protokolliert die durchgeführten Tests in einem ausführlichen Test-Protokoll.

Das erstellte Test-Protokoll kann dann für wiederholte Tests dieses Programms verwendet werden. Dieses Test-Protokoll ist für die Tester selbst und wird im Normalfall nicht an die Entwickler weitergegeben.

Wenn Fehler gefunden wurden, werden die Fehler in einer eigenen Fehlerliste protokolliert. Diese Fehlerliste wird nach der Testdurchführung an die Entwickler weitergegeben.

Bei Bedarf wird ev. eine zusätzliche persönliche Besprechung zwischen dem Tester und Entwickler stattfinden, in welcher die Fehler detaillierter besprochen werden.

Der Entwickler behebt die Fehler und testet wiederum zwingend die Standard-Fälle für das erstellte Programm und dokumentiert dies entsprechend in einem kurzen Test-Protokoll.

Dann beginnt der Testprozess erneut, wobei je nach Testspezifikation und Test-Richtlinien eventuell wiederum alle Testfälle durchgetestet werden müssen (Regressionstest).

Wenn keinen Fehler mehr durch den Tester gefunden wurden, ist der Testprozess beendet.

Ergänzend dazu gibt es in Phasen vor und nach der Programmierung ebenfalls die Notwendigkeit für analytische Qualitätsmanagement-Maßnahmen, die jedoch hier aus Platzgründen nicht alle dargestellt werden können.

Folgende Testphasen und -aktivitäten sind noch zu berücksichtigen:

- Entwickeln einer Testorganisation und eines Testplans
- Testen in der Anforderungs- und Designphase
- Installations- und Abnahmetest
- Testen in der Betriebs- und Wartungsphase
- Testberichte und -aufzeichnungen
- Evaluieren der Testeffektivität

➔ **Testspezifikation und -dokumentation**

Da die Testspezifikation und -dokumentation als Nachweis für eine sorgfältige Testdurchführung und auch als Basis für wiederholte Tests eine sehr wichtige Grundlage ist, soll dieses Thema nachfolgend noch dargestellt werden:

Bestandteile einer Testdokumentation:

Es gibt in diesem Zusammenhang keine verbindlichen Namensgebungen, so dass die Bezeichnungen von Unternehmen zu Unternehmen unterschiedlich sein können. Mit Hilfe der nachfolgenden Aufstellung sollte es möglich sein, die einzelnen Komponenten zuordnen zu können.

Üblicherweise können folgende Arten von Dokumenten und Berichten unterschieden werden:

Organisationsdokumente:

- Testorganisation

Testplanungsdokumente:

- Testspezifikation
- je nach Umfang und Größe des Projekts ev. zusätzlich noch gegliedert in:
 - Testdrehbuch
 - Testfälle

Ergebnis-/Berichtsdokumente:

- Testprotokoll
- Problemmeldung
- Abschlussbericht
- Software-Übergabe/Abnahmeprotokoll

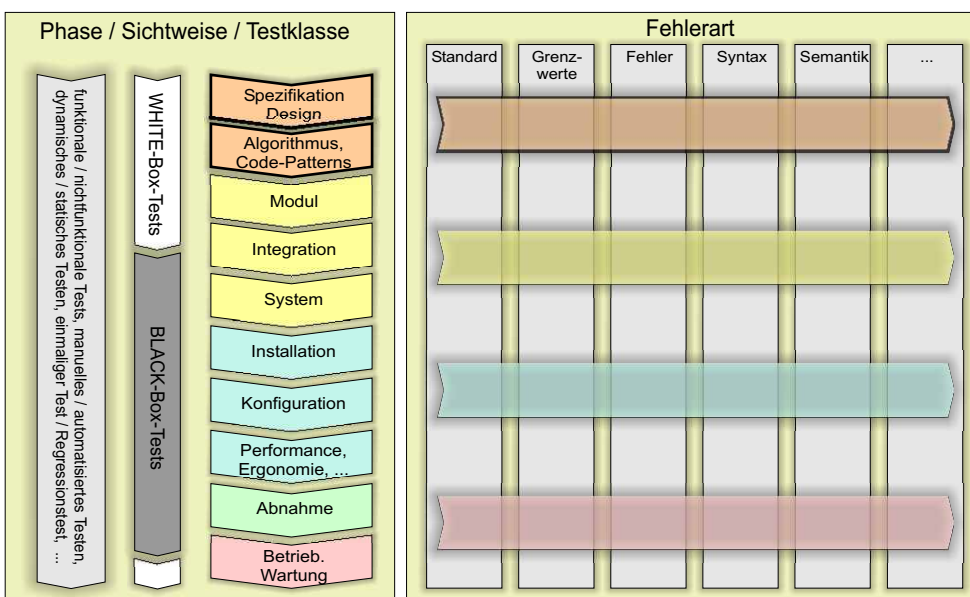
➔ **Testarten und -phasen**

Im Rahmen der Testspezifikation werden die durchzuführenden Testschritte und Testfälle beschrieben.

Es gibt eine Vielzahl von Testarten, die durchgeführt werden können.

Eine Übersicht und strukturierte Gliederung von einigen Testarten ist in der nebenstehenden Grafik dargestellt.

Testen findet in jeder Phase der Software-Entwicklung (beginnend bei der ersten Ideenstrukturierung bis hin zum laufenden Betrieb und Wartung) statt und sollte daher nicht nur auf den Zeitpunkt nach der Programmierung bzw. auf die Abnahmephase beschränkt und umgesetzt werden.



Darstellung verschiedener Testarten und -Phasen
Quelle: Bergsmann

Fortsetzung - Software-Testen >>>

➔ Testen von Standard-Software

Abschließend werden beispielhaft noch Hinweise für das Testen von Standard-Software angeführt, die vor allem für den Kunden / Auftraggeber wichtig sind:

Neben der reinen funktionalen Prüfung ist es erforderlich, noch weitere kritische Erfolgsfaktoren im Rahmen der Qualitätsprüfung zu berücksichtigen:

- Benutzerfreundlichkeit
- Erweiterungsfähigkeit
- Wartbarkeit
- Kosten-Nutzen-Effektivität
- Portabilität
- Zuverlässigkeit
- Sicherheit
- ...

Die Besonderheit bei Auftraggeber-Tests ist, dass hier der Tester meist nicht über den Programmcode oder Details zur Systemarchitektur verfügt.

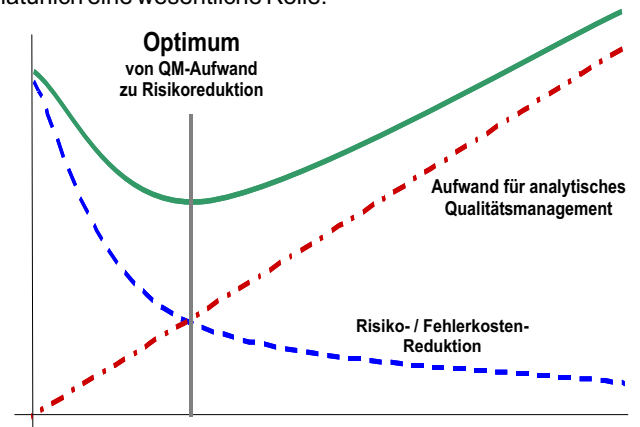
Daher sind in diesem Fall auch nur die Black-Box-Test-Methoden anwendbar.

Für den Auftraggeber empfiehlt es sich, insbesondere folgende Richtlinien zu beachten (nach Perry):

- Wenn zumindest ein Tag (bei größeren Systemen auch mehrere) investiert wird, um die Software zu erlernen und zu überprüfen, wird dies mit einer problemloseren Verwendung der Software entlohnt.
- Die Software sollte erst erworben werden, wenn der Bedarf für diese Software festgestellt wurde und demonstriert werden kann, wie diese Software im täglichen Gebrauch verwendet wird.
- Es sollte bei der Prüfung der Software auch das Bauchgefühl bezüglich der Eignung einfließen.
- Das Testen wird nicht vorgenommen, um das Leben zu verkomplizieren, sondern vielmehr, um es zu erleichtern. Nachdem die Software getestet wurde, kann sie sicherer bedient werden und es sind weniger unbekannte Größen, was funktioniert und was nicht.
- Die Kosten und die Frustration bei der Ersetzung einer ungeeigneten Software sind meist erheblich niedriger als die Kosten eines weiteren Einsatzes einer ungeeigneten Software.
- Personen, die das größte Interesse an der korrekten Funktionsweise der Software besitzen, führen auch die besten Tests durch. Diese Mitarbeiter sollten daher entsprechend Zeit und Möglichkeiten erhalten, um den Test durchzuführen und auch die Ergebnisse zu evaluieren.
- Wenn Benutzer den Abnahmetest nach einer Schulung erfolgreich durchführen können, dann werden die auch in der Lage sein, die Software im Betrieb zu bedienen.

In welcher Tiefe und Umfang Tests durch den Hersteller und auch durch den Kunden / Anwender durchgeführt werden, hängt letztlich von der Größe des Software-Systems und von der Art des Einsatzes bzw. des Umfelds und des damit verbundenen Risikopotentials zusammen.

Der wirtschaftliche Faktor (Was kostet die Testdurchführung im Verhältnis zur erreichten Risikoreduktion) spielt dabei natürlich eine wesentliche Rolle.



Es wird jedoch nicht vernünftig und zulässig sein (ausgenommen bei einer expliziten beidseitigen Vereinbarung), auf einen dem Anwendungsrisiko und der Software-Art angepassten und entsprechenden Test komplett zu verzichten!

Links, Literatur ...

- ⇒ Pol, Koomen, Spillner, *Management und Optimierung des Testprozesses*, dpunkt.verlag, 2000
- ⇒ Perry, *Software testen*, mitp Verlag, 2003
- ⇒ Thaller, *Software-Test*, Heise Verlag, 2000
- ⇒ <http://www.softwareqatest.com/>, interessante Web-Site zu Qualitätsmanagement und Testen mit vielen Inhalten und Links zu anderen Seiten.

Zitate ...

- ⇒ Testen reduziert das Maß an Unsicherheit in Bezug auf die Qualität eines IT-Systems!
Pol, Koomen, Spillner; Management und Optimierung des Testprozesses
- ⇒ Prüfe die Brücke, die Dich tragen soll!
altes Sprichwort
- ⇒ Das Ziel ist nicht die Entwicklung einer Software mit der besten, sondern mit der richtigen Qualität.
Liggismeyer, Software-Qualität
- ⇒ Nur wenige Firmen verfügen über einen Maßstab zur Bewertung der Effektivität des Testens. Dies erschwert das Feststellen der Kosteneffektivität im Testprozess.
Perry, Software testen

Impressum:

Herausgeber und für den Inhalt verantwortlich:

Die Beiträge wurden sorgfältig ausgewählt und bearbeitet. Für Druckfehler und Irrtümer wird nicht gehaftet. Die Rechte von zitierten Beiträgen und Gastbeiträgen liegen bei den jeweiligen Autoren.

Software Quality Lab

Fliederstrasse 8
A-4222 Langenstein

www.software-quality-lab.at

info@software-quality-lab.at

Tel.: +43-(0)664-16 20 220, Fax: +43-(0)7237-4941