

Ausgabe 2012 / 4

Erscheinungsart: ca. 4 x jährlich in elektronischer Form

Die Welt der Software-Qualität

In dieser Ausgabe:

- ⇒ Ausrichtung auf Software-Qualität
- ⇒ Wichtige Themen im Bereich der Software-Qualität
- ⇒ Literatur, Zitate

Die Ausrichtung auf Software-Qualität bedeutet Veränderung

Viele Entwicklungsorganisationen verharren in Selbstzufriedenheit mit folgendem Denken und Handeln:

- Wir sind doch der Marktführer/ die Besten
- Dazu haben wir jetzt keine Zeit
- Das ist nur eine Konjunkturkrise
- Bei uns ist das alles ganz anders
- Wir machen doch Gewinne, unser Geschäft läuft doch
- Den anderen geht es auch nicht besser
- Wir schwärmen von Erfolgen der Vergangenheit
- Wir fokussieren uns auf das Tagesgeschäft
- Mit unseren Mitarbeitern geht das nicht
- Etc.

Organisationen, die so oder ähnlich denken, werden sich aus dem Sumpf der Verweigerung, Verwirrung und Selbstzufriedenheit nicht lösen und typischerweise von anderen Organisationen überholt und in den Schatten gestellt.

Leider gilt auch oft das Sprichwort „Nachdem wir das Ziel aus den Augen verloren hatten, verdoppelten wir unsere Anstrengungen!“

Was können Sie nun tun, um sich mit Ihrer Organisation weiter zu entwickeln?

- Stellen Sie die Denkgewohnheiten in Frage und schauen Sie gemeinsam über den Tellerrand hinaus.
- Erzeugen Sie Aufbruchsstimmung und geben Sie dem Denken eine neue Richtung z.B. durch eine mit dem Management entwickelte gemeinsame Vision.
- Brechen Sie eingefahrene Gruppenstrukturen auf und bewegen Sie die Mitarbeiter zu Veränderungen - ev. auch physisch, z.B. Job-Rotation zwischen Entwicklern und Testern oder gemischte Teams wie in der agilen Entwicklung.
- Leben Sie Veränderungen selbst vor, auch wenn es vorerst nur in einem eingeschränkten Organisations- oder Themenbereich ist.
- Wichtig ist, dass alle beteiligten Personen inkl. dem verantwortlichen Management anfangen, über Veränderung nachzudenken und begreifen, dass Veränderungen notwendig sind!

In diesem Sinn kann der vorliegende Knowledge-Letter einige Denkanstöße liefern, welche Themen zur Verbesserung der Software Qualität in der Organisation verändert und optimiert werden können.



Wie die Erkenntnis wächst!

Es gibt drei Arten von Organisationen:

- Die einen bewirken, dass etwas geschieht
- die anderen beobachten, dass etwas geschieht und
- wieder andere fragen sich, was geschehen ist!

In der Welt der Software-Qualität ist die erstgenannte Gruppe leider noch in der Minderheit. Jedoch hat es in den letzten 10 Jahren eine deutliche Verschiebung hin zu den engagierten Organisationen gegeben.

Immer mehr Ausbildungsinstitutionen (UNIs, FHs, etc.) nehmen Software-Qualitätsthemen in das Ausbildungsprogramm und die Absolventen tragen dieses Wissen hinaus in die Unternehmen.

Wo früher in den Software-Organisationen nur Entwickler gearbeitet haben, finden wir heute auch schon einige Tester.

Wo früher die Tester diejenigen waren, die ins Testen „abgeschoben“ wurden, weil sie nicht gut genug programmieren konnten, sind die Tester heute oft höher qualifiziert als die Entwickler und in einer Schlüsselrolle zwischen Produktmanagement, Kunden und Entwicklung.

Viele Software-Organisationen machen sich auch Gedanken über die Prozesse und (weniger oft) werden diese auch systematisch definiert und weiterentwickelt.

Der Trend geht in die richtige Richtung!

Dipl.-Ing. Johannes Bergsmann

Staatl. befugter und beedeter Ingenieurkonsultent für Informatik

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 und andere Fachbeiträge aus unseren Themenbereichen.

Aktuelle Fach- und Forschungsbeiträge sind willkommen. Einsendungen an info@software-quality-lab.com.

Weitere Infos zu diesem und anderen Themen finden Sie auf www.software-quality-lab.com.

Die Welt der Software Qualität

DI Johannes Bergsmann - Software Quality Lab

Software Qualität ist nicht nur Testen und Qualitätssicherung, sondern viele Themen sind in diesem Zusammenhang wichtig wie z.B. Requirements-Engineering, Change-Management, Code-Qualität, Risiko-Einschätzungen, angemessene Prozesse, Automatisierung, Vorgehensweisen und Prozesse, Tools und Tool-Integration, etc.

In diesem Beitrag erhalten Sie einen Überblick über wichtige Themen aus Sicht der Software-Qualität.

Einleitung & Abgrenzung

Dieser Beitrag soll verschiedene Kernthemen, die für den Bereich Software Qualität wichtig sind, ansprechen und vor allem als Gedankenanstoß dienen, welche Fragen man sich selbst in diesen Themen stellen sollte und worauf man in der eigenen Organisation achten sollte.

Dieser Beitrag soll NICHT eine vollständige Abhandlung zum Thema Software Qualität sein und es darf aufgrund des geringen Umfangs auch nicht erwartet werden, dass umfangreiche Analysen und Lösungsansätze vorgestellt werden.

Requirements

Das Thema Requirements ist ein Dauerbrenner in fast jeder Entwicklungsorganisation und das schon seit Jahrzehnten. Die Langzeit-Beobachtung lässt fast befürchten, dass viele Organisationen dieses Thema auch in den nächsten Jahrzehnten noch nicht nachhaltig gelöst haben.



Viele Betroffene tapen hier noch ziemlich im Dunkeln - nicht nur, was die Inhalte der Requirements an sich betrifft, sondern es sind hier einfach in den meisten Fällen noch viele Fragestellungen offen wie z.B.

- **Umfang, Detailliertheit und Qualität der Requirements**

Einerseits ist dies unter dem Aspekt zu sehen, dass möglichst überflüssiger Ballast und administrativer Aufwand vermieden wird. Andererseits jedoch soll auch der Wissenstransfer durch den Prozess sichergestellt werden und die Requirements sollen als gute Basis für die weiteren Prozess-Schritte und die Dokumentation des Systems auch eine entsprechend gute Qualität aufweisen.

- **Unterscheidung WAS – WIE**

Oft werden Lösungsausarbeitungen (das WIE es umgesetzt wird) und die Anforderungen aus der Kundensicht (das WAS der Kunde gerne haben möchte) vermischt. In den heute üblichen iterativen Vorgehensweisen ist es auch notwendig zwischen dem WAS und dem WIE einen ständigen Abgleich und Konsens zu schaffen. Jedoch sollte schon aus dem Aspekt der unterschiedlichen Verantwortlichkeit und Kompetenz für verschiedene Themen darauf geachtet werden, dass diese beiden Sichten

nicht vermischt werden.

Aus einer Anforderung sollte klar hervorgehen, ob sie in der Kundenverantwortung liegt oder ob sie ein Vorschlag des Umsetzers ist.

- **Aufbau und Struktur**

Die Strukturierung der Requirements ist auch oft ein Thema. Viel zu schnell begibt man sich dabei in die Tiefen des zu spezifizierenden Systems und vergeudet wertvolle Zeit mit Details, die in der jeweiligen Phase noch gar nicht relevant sind. Eine angemessene Top-Down-Vorgehensweise von der Idee über Projektziele, Kontextdarstellungen, Business-Prozesse, Use-Cases, Features bis hin zur detaillierten Masken- und Schnittstellenbeschreibung ist hier meist zielführend, um den Überblick nicht zu verlieren.

- **Genügend Zeit? Der Chef bzw. Auftraggeber drängt?**

In vielen Unternehmen ist es ein Bewusstseins-Thema nach dem Motto „Wieso wird denn hier so viel geschrieben? Wäre es nicht besser, wenn endlich gearbeitet wird?“. In solchen Organisationen erübrigt sich das Thema Requirements-Spezifikation meist von selbst. In vielen Fällen ist es aber auch der Zeitdruck, der meist aus einer falschen Aufwandsschätzung und Planung resultiert. Es wird dann einfach zu wenig Zeit für eine gute Spezifikation eingeplant oder auch bei guter Planung kann es dem Auftraggeber oder Chef oft nicht schnell genug gehen, bis er endlich etwas Vorzeigbares sieht. Meist passiert dies dann auf Kosten der Qualität im Prozess.

- **Requirements-Engineering in agilen Projekten**

Aufgrund der Verbreitung der agilen Vorgehensweisen ist dies immer mehr ein wichtiges Thema, da die meisten agilen Vorgehen darauf leider nicht genau eingehen. User-Stories und andere agile Requirements-Artefakte werden zwar erwähnt, aber leider nur sehr oberflächlich beschrieben. Daher sollte sich jede Organisation, die agile Vorgehensweisen anwendet, auch Gedanken darüber machen, wie mit den Requirements umgegangen wird.

Sehr oft beschränkt man sich leider nur auf 1-2 Ebenen wie z.B. User-Stories und Use-Case-Descriptions. Themen wie Business-Prozess-Sicht, Architekturdokumentation, nicht-funktionale Anforderungen und andere

Vollständiger Knowledge Letter Zugang

Wir freuen uns, dass Sie an diesem Thema Interesse haben und den Knowledge Letter von Software Quality Lab bis hierher gelesen haben.



Dieser Knowledge Letter ist eine Vorschau (gekürzte Version des gesamten Artikels).

Wenn Sie den ungekürzten Knowledge Letter lesen möchten, registrieren Sie sich bitte unter <http://www.software-quality-lab.com/download/knowledge-letter/anfrage-knowledge-letter/>

Sie erhalten nach der Registrierung vollen Zugang zu allen bisherigen Knowledge Letters von Software Quality Lab und erhalten automatisch künftige Knowledge Letter per E-Mail.

Software Quality Days — Die größte Konferenz zum Thema „Software Qualität“ in Europa!



Besuchen Sie die Top-Konferenz mit allen Infos rund um Software Qualität.

Beste Qualität der Vorträge und Tutorials sowie eine Mischung aus praktischen und wissenschaftlichen Beiträgen machen die Software Quality Days zum Top-Event.

In den 3 praktischen Tracks werden anwendungsorientierte Vorträge präsentiert. Der wissenschaftliche Track zeigt Beiträge mit hohem Innovationsgrad und praktischer Anwendbarkeit, basierend auf Forschungsergebnissen. Im Solution Provider Forum präsentieren Aussteller ihre neuesten Tools mit Praxis-Beispielen.

Nähere Infos unter

www.software-quality-days.com

