→ 2., überarbeitete und erweiterte Auflage

Kurt Schneider

Abenteuer Software Qualität

Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement

dpunkt.verlag

Abenteuer Softwarequalität

Prof. Dr. Kurt Schneider leitet das Fachgebiet Software Engineering an der Leibniz Universität Hannover. Er hat in Erlangen Informatik studiert und anschließend an der Universität Stuttgart promoviert. Bei einem Forschungsaufenthalt an der University of Colorado at Boulder beschäftigte er sich mit Techniken zum systematischen Lernen aus Erfahrung im Software Engineering. Er war sieben Jahre bei der DaimlerChrysler AG am Forschungszentrum Ulm tätig. In Projekten mit verschiedenen Unternehmensbereichen spielten Softwarequalität, Prozessgestaltung und wiederum die Erfahrungsnutzung eine wichtige Rolle. Zu seinen Forschungsthemen gehören daneben Softwareanforderungen und agile Methoden, um Informationsflüsse und die Dokumentation in Softwareprojekten zu optimieren. Kurt Schneider legt viel Wert darauf, diese Themen praxisnah zu bearbeiten und zu vermitteln.

Kurt Schneider

Abenteuer Softwarequalität

Grundlagen und Verfahren für Qualitätssicherung und Qualitätsmanagement

2., überarbeitete und erweiterte Auflage

dpunkt.verlag

Prof. Dr. Kurt Schneider
Kurt.Schneider@inf.uni-hannover.de

Lektorat: Christa Preisendanz
Copy-Editing: Ursula Zimpfer, Herrenberg
Herstellung: Birgit Bäuerlein
Umschlaggestaltung: Helmut Kraus, www.exclam.de
Druck und Bindung: M.P. Media-Print Informationstechnologie GmbH, 33100 Paderborn

Fachliche Beratung und Herausgabe von dpunkt.büchern im Bereich Wirtschaftsinformatik:
Prof. Dr. Heidi Heilmann · heidi.heilmann@augustinum.net

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über http://dnb.d-nb.de abrufbar.

ISBN:
Buch 978-3-89864-784-7
PDF 978-3-86491-108-8
ePub 978-3-86491-109-5

2., überarbeitete und erweiterte Auflage 2012
Copyright © 2012 dpunkt.verlag GmbH
Ringstraße 19 B
69115 Heidelberg

Die vorliegende Publikation ist urheberrechtlich geschützt. Alle Rechte vorbehalten. Die Verwendung der Texte und Abbildungen, auch auszugsweise, ist ohne die schriftliche Zustimmung des Verlags urheberrechtswidrig und daher strafbar. Dies gilt insbesondere für die Vervielfältigung, Übersetzung oder die Verwendung in elektronischen Systemen.
Es wird darauf hingewiesen, dass die im Buch verwendeten Soft- und Hardware-Bezeichnungen sowie Markennamen und Produktbezeichnungen der jeweiligen Firmen im Allgemeinen warenzeichen-, marken- oder patentrechtlichem Schutz unterliegen.
Alle Angaben und Programme in diesem Buch wurden mit größter Sorgfalt kontrolliert. Weder Autor noch Verlag können jedoch für Schäden haftbar gemacht werden, die in Zusammenhang mit der Verwendung dieses Buches stehen.
5 4 3 2 1 0

Barbara, Stefan und Angelika
gewidmet

Vorwort

Softwarequalität bleibt spannend

Die Informatik ist eine schnelle Disziplin: Was heute neu ist, kann morgen schon veraltet sein. Internetanwendungen, Smartphone-Apps und zahllose neue Techniken kommen und gehen in einem atemberaubenden Tempo.

Aber auch in der Informatik hat manches Bestand. Die Grundlagen der Softwarequalität gehören dazu. Als 2007 die erste Auflage dieses Buches erschien, wollte ich neben den Techniken zum Testen und Prüfen unbedingt auch vermitteln, wie es sich anfühlt, wenn man für die Qualitätssicherung zuständig ist. Um das ein bisschen lebendiger zu gestalten, habe ich die Figur »Q« eingeführt. Der Leser folgt Q durch verschiedene Situationen in einem Unternehmen. Dabei kommen die Randbedingungen, die Erwartungen und Befürchtungen zum Vorschein, die wesentlich mitentscheiden, was in der Praxis möglich und sinnvoll ist. Die Menschen ändern sich nicht so schnell.

In diesem Buch werden die Grundideen und wichtigsten Methoden der Softwarequalität vorgestellt. Mein Ziel ist immer, dass die Leser nicht nur verstehen, wie ein Ansatz funktioniert, sondern auch, wozu er dient und was man bei der Anwendung bedenken muss. Diese Art von Wissen hat eine »längere Halbwertszeit« als technische Spezifikationen oder Modethemen. An einer Universität sollen die Studierenden Prinzipien kennenlernen, die ihnen auch nach zehn Jahren noch nützlich sind. Dieses Buch ist aus einer universitären Vorlesung entstanden und soll für solche Themen ein solides Verständnis vermitteln.

In der zweiten Auflage hat sich einiges geändert und ist ergänzt oder aktualisiert worden. Neu ist ein Kapitel über die agilen Methoden. Wer sich ernsthaft für Softwarequalität interessiert, muss heute wissen, worum es dabei geht und welche Konsequenzen agile Methoden wie Extreme Programming, Scrum, Lean Software Development oder Kanban auf die Softwarequalität haben. Natürlich können nicht alle diese Methoden im Detail diskutiert werden; für jede einzelne gibt es dicke Bücher. Aber auch hier ist es mein Anliegen, auf wenigen Seiten eine vernünftige Orientierung zu geben.

Ich danke meinen Mitarbeitern am Lehrstuhl Software Engineering für spannende Diskussionen in vielen Softwareprojekten. Raphael Pham nimmt sich zurzeit mit besonderem Engagement der Übungen zur Vorlesung an. Freundlich und kompetent haben mich Heidi Heilmann und Christa Preisendanz vom dpunkt.verlag auch bei der zweiten Auflage unterstützt. Vielen Dank!

Kurt Schneider
Hannover, Februar 2012

Vorwort zur 1. Auflage

Softwarequalität und Abenteuer

Softwarequalität kann eine spannende Sache sein. Das sieht man aber nicht sofort.

Als Student habe ich noch nicht viel von Softwarequalität gehört. Natürlich, wir haben auch damals schon getestet. Aber in den Lehrbüchern hörte sich das alles doch eher bürokratisch und langweilig an. Eher etwas für große Firmen als für spannende Projekte.

Im Laufe der Zeit hatte ich dann an verschiedenen Stellen die Gelegenheit, Softwarequalität in der Praxis mitzuerleben. Da kam es plötzlich neben den Techniken und Verfahren auch auf ganz andere Dinge an – auf Einfühlungsvermögen und Durchsetzungsfähigkeit, auf Aufwand, Kosten und auf guten Überblick. Das war viel aufregender und herausfordernder.

Auch in meiner Vorlesung *Software-Qualität* bin ich zuerst streng »nach Lehrbuch« vorgegangen. Aber wieder hatte ich das Gefühl, dass da etwas fehlte, und zwar etwas ganz Wichtiges. Dann hatte ich eine Idee. Ich ließ eine fiktive Person immer wieder in der Vorlesung auftauchen. Durch diese Person konnte ich auch Zweifel und Hoffnungen ausdrücken, ohne sie zu sehr mit dem Lehrstoff zu durchmischen. Das hat den meisten Studierenden laut einer Evaluation gut gefallen.

Diesem Buch liegt die gleiche Idee zugrunde. Es behandelt die üblichen Themen, ergänzt sie aber um einige Aspekte, die den Zusammenhang herstellen und das Bild abrunden. Die fachliche Sicht wird ergänzt durch die fiktive Person – und ihre Abenteuer in der Softwarequalität. Das ist für ein Fachbuch vielleicht ungewöhnlich. Ich hoffe aber, dass der Stoff so leichter zu lesen und zu verstehen ist.

Danke

Ich möchte mich bei vielen bedanken, die direkt oder indirekt zu diesem Buch beigetragen haben. Das beginnt bei meinen Kollegen an der Universität Stuttgart, Horst Lichter, Marcus Deininger, Jürgen Schwille und Anke Drappa. Mein Doktorvater Jochen Ludewig hat mir viel beigebracht; seine prägnanten Bücher haben mir imponiert, und viele Grundüberzeugungen habe ich von ihm. Marcus Deininger verdanke ich besonders spannende Diskussionen um die »Softwarequanten«, mit denen wir Softwareprojekte und Softwarequalität simuliert haben.

Bei Gerhard Fischer an der University of Colorado at Boulder wurde mir zum ersten Mal das Spannungsverhältnis zwischen Nützlichkeit und Bedienbarkeit von Software bewusst. In Boulder habe ich auch gesehen, wie kreativ man damit umgehen kann.

Auch meinen Kollegen bei DaimlerChrysler möchte ich herzlich danken – es sind zu viele, um sie hier alle zu nennen. Michael Offergeld hat meine Sicht auf Usability geprägt und weiterentwickelt, die ich von Boulder mitgebracht hatte. Viele Beobachtungen verdanke ich der gemeinsamen Arbeit mit Thilo Schwinn, dem Experten für Reviews. In unseren Arbeiten zu erfahrungsbasierter Prozessverbesserung waren viele Kollegen beteiligt, darunter Frank Sazama, Stefanie Lindstaedt und Frank Houdek. Frank Houdek ist jetzt seit langem der ausgewiesene Experte für Anforderungen bei DaimlerChrysler. Die Projekte mit Dieter Rombach und Vic Basili waren eine große Bereicherung für mich und haben meine Erfahrung in der Softwarequalität geprägt. Thomas Gantner war ein Vorbild als ein Projektleiter, dem die Qualität ein echtes Anliegen ist.

An der Leibniz Universität Hannover haben Thomas Flohr, Daniel Lübke und Eric Knauss regen Anteil daran, dass sich meine Vorlesung zur Softwarequalität weiterentwickelt. Etliche Anregungen dazu verdanke ich Martin Glinz von der Universität Zürich. Dieses Buch ist der Versuch, die vielen Einflüsse der letzten Jahre zu sortieren und eine Auswahl davon mit einer durchgängigen Geschichte zu verbinden. Das hat Spaß gemacht.

Ich danke dem dpunkt.verlag für die Betreuung und Heidi Heilmann für die engagierte fachliche Durchsicht; Christa Preisendanz hat sich besonders für das Buch eingesetzt.

Vielen Dank für die inspirierenden Gespräche und die Zusammenarbeit!

Kurt Schneider
Hannover, April 2007

Inhalt

1 Einleitung

1.1 Softwarequalität betrifft viele

Softwarequalität ist ein Thema, das heute jeden direkt oder indirekt betrifft.

Fast jeder geht privat oder im Beruf ständig mit Programmen und softwaregesteuerten Geräten um: Handys und Autos, Internet oder Waschmaschine. Wir verlassen uns darauf, dass die Programme funktionieren – und zwar so, wie wir es erwarten. Aber dies ist leider nicht immer der Fall.

Nicht nur Fachzeitschriften, sondern auch normale Tageszeitungen berichten immer wieder von großen Softwarepannen. Durch fehlerhaft oder schlecht bedienbare Software geht viel Zeit verloren; Fehlbuchungen kosten Geld und Nerven. Viele Unternehmen verbinden mit Softwarequalität zunächst vor allem Kosten. Sie denken an unzufriedene Kunden oder Rückrufaktionen.

Natürlich gibt es auch Programme von guter Qualität, die nicht abstürzen und die leicht bedienbar sind. Sie sind von Fachleuten entwickelt, denen Qualität ein Anliegen ist. Softwarequalität geht nicht nur Informatiker oder Studierende etwas an: Auch Projektleiter, Manager und sogar Auftraggeber brauchen heute solide Grundkenntnisse, um ihren Anteil zu einer guten Softwarequalität beizutragen.

Was bedeutet aber »gute Qualität« überhaupt? Software und ihre Qualität sind nicht greifbar und daher auch schwer zu messen. Umso wichtiger ist es, dabei systematisch vorzugehen.

1.2 Für wen dieses Buch gemacht ist

Natürlich müssen sich **Qualitätsbeauftragte, Entwickler und Projektleiter** von Softwareprojekten mit Softwarequalität beschäftigen. Nicht jeder und jede von ihnen muss alle Details kennen, aber einen breiten Überblick und ein gutes Verständnis für die wichtigsten Zusammenhänge kann man erwarten. Dieses Buch soll genau dieses Verständnis fördern.

Damit lässt sich auch gut **altes Wissen auffrischen**. Wer also schon vieles weiß, der wird sich mit diesem Buch schnell an einiges erinnern, das fast vergessen war. Denn das Buch enthält auch viele konkrete Begriffsklärungen, Beispiele und Tipps. Man kann sich gezielt ein Thema vornehmen und erfährt, wie die Verfahren funktionieren, was also dahintersteckt. Wer zu einem speziellen Thema noch mehr wissen will, wird in der angegebenen Spezialliteratur fündig werden.

Studierende der Informatik und angrenzender Gebiete finden hier Orientierung: Sie können sich mit den Grundideen zum Testen, Messen und von Reviews vertraut machen und sehen Zusammenhänge zu aktuellen Themen wie Bedienbarkeit oder agilen Methoden. Manche Themen sind aus der Praxis motiviert: Quality Gates sind ein Beispiel, das überall verbreitet ist, aber in der Lehre noch selten vorkommt.

Das Buch ist absichtlich schlank und lesbar gehalten, damit man es auch fast wie einen Roman lesen kann. Es ist durchzogen von einer durchgängigen Geschichte: dem Abenteuer von Q in der Softwarequalität. So bekommt man nebenbei einiges darüber mit, was im Kopf von **Qualitätsbeauftragten** vorgeht. Das können alle brauchen, die einmal in Projekten arbeiten werden. Egal, ob sie mit Qualitätsbeauftragten zusammenarbeiten – oder selbst einer werden wollen.

Viele **Manager und Auftraggeber** von Softwareprojekten haben erkannt, dass sie sich selbst einen Gefallen tun, wenn sie sich ein Grundverständnis für die eingesetzten Verfahren, für die Stärken und Schwächen, und auch für den Aufwand von Qualitätsmaßnahmen aneignen. Für diesen Zweck kann man einige Abschnitte überspringen, wird aber vom Überblick und dem Zusammenhang profitieren.

Und wenn Sie zu keiner dieser Gruppen gehören, sich aber einfach schon lange **für Softwarequalität interessieren**? Dann ist das Buch natürlich auch für Sie gemacht.

1.3 Was Sie von diesem Buch erwarten können

In diesem Buch geht es darum, allen Interessierten einen fundierten Überblick über das Thema Softwarequalität zu geben. Es gibt dicke Bücher über Software Engineering, in denen *auch* Softwarequalität eine gewisse Rolle spielt. Diese Einführung konzentriert sich dagegen ganz auf Softwarequalität. Die bekanntesten Begriffe werden geklärt und die wichtigsten Verfahren erläutert. Damit soll der Einstieg in das Thema gelingen.

Jeder, der programmieren kann, hat schon einmal etwas vom Testen gehört. Aber *systematisches* Testen ist für viele Studierende und Praktiker graue Theorie geblieben. In diesem Buch geht es darum – beim Testen wie bei den anderen Themen –, kurz und verständlich herauszuarbeiten, worauf man achten muss, damit am Ende Software der gewünschten Qualität herauskommt. Auch wenn man

nicht das letzte Detail jedes Verfahrens durchdringt, soll man sich nach der Lektüre doch ein gutes Urteil bilden können. Und das ist das Wichtigste.

Softwarequalität ist ein technisches Thema, aber nicht nur ein technisches: Psychologie und Einfühlungsvermögen sind für die Softwarequalität nicht nur wichtig, sondern unentbehrlich. Projekte stehen heute meist unter hohem wirtschaftlichem Druck und sind ständig in Eile. Man muss daher Aufwand und Nutzen von Verfahren einschätzen können, um realistische Qualitätssicherungsmaßnahmen zu planen und erfolgreich durchzuführen. Der ständige Druck wirkt sich auf die Arbeitsweise im Projekt aus und auf die Gefühlslage der Beteiligten – ein wichtiger Aspekt für erfolgreiche Projekte. Wie vermittelt man das in einem Buch?

1.4 Das Abenteuer von Q

Ich versuche es mit einer fiktiven Person, Herrn oder Frau »Q«. Neben den Sachinformationen wird in diesem Buch die Geschichte von Q erzählt: das alltägliche Abenteuer, als Qualitätsbeauftragter in einer Softwarefirma zu arbeiten. Diese Geschichte soll die Gefühlsperspektive betonen, Fragen und Bedenken aufgreifen, die auch manchen Leser beschäftigen werden. Herr oder Frau Q hat Informatik studiert und bewirbt sich auf eine Stelle in der Softwarequalität. Indem wir Q durch die fiktive Firma *FunGate* folgen, lassen sich Zweifel und Hoffnungen besser nachvollziehen, die man in einem traditionellen Lehrbuch unter den Teppich kehren würde, um sich »auf die Sache zu konzentrieren«. Sie gehören aber »zur Sache«! Zweifel und Hoffnungen sind in der Softwarequalität wichtige Voraussetzungen für eine pragmatische Einschätzung.

Die Geschichte von Q ist kursiv gesetzt und hebt sich vom restlichen Text ab; man kann sie also auch leicht überblättern, wenn man möchte. Q soll einen roten Faden durch die Themen der Softwarequalität ziehen und damit ein wenig »Erfahrungsqualität« vermitteln.

Wie wir sehen werden, ist Erfahrung ein Schlüsselbegriff in der Softwarequalität.

1.5 Themen und Anspruch

Softwarequalität ist ein weites Feld. Wenn man die ökonomischen und menschlichen Seiten des Gebiets mit einschließt, umso mehr. Daher muss man für eine kompakte Übersicht Themen auswählen und andere weglassen. Die hier vorgestellten Themen gehören zum Grundwissen für alle Projektbeteiligten, auch für Kunden von Softwareprojekten.

Damit ist das Thema Softwarequalität nicht erschöpfend behandelt, aber auf eine vernünftige Grundlage gestellt. Je nach Interessenlage kann man sich in verschiedene Richtungen vertiefen.

1.5.1 Themenauswahl und Gewichtung

Welches Niveau von Kenntnissen soll das vorliegende Buch vermitteln? Zumindest muss man die einschlägigen Begriffe *»kennen«*, sie einordnen können und verstehen, wenn jemand anders sie gebraucht. Das ist die erste Stufe. Sie soll für alle vorgestellten Themen erreicht werden.

In manchen Fällen sollte man zusätzlich in der Lage sein, etwas selbstständig *»anzuwenden«*, also in einem einfachen Fall durchzuführen. Für diese Stufe sind ausführlichere Informationen nötig, damit der Leser das Verfahren ausprobieren kann. Manche Techniken, wie das Testen, sollte man im Projekt sogar regelrecht *»beherrschen«*, also auch unter etwas ungewöhnlichen Umständen sicher einsetzen können. Entsprechend werden diese Themen besonders intensiv behandelt.

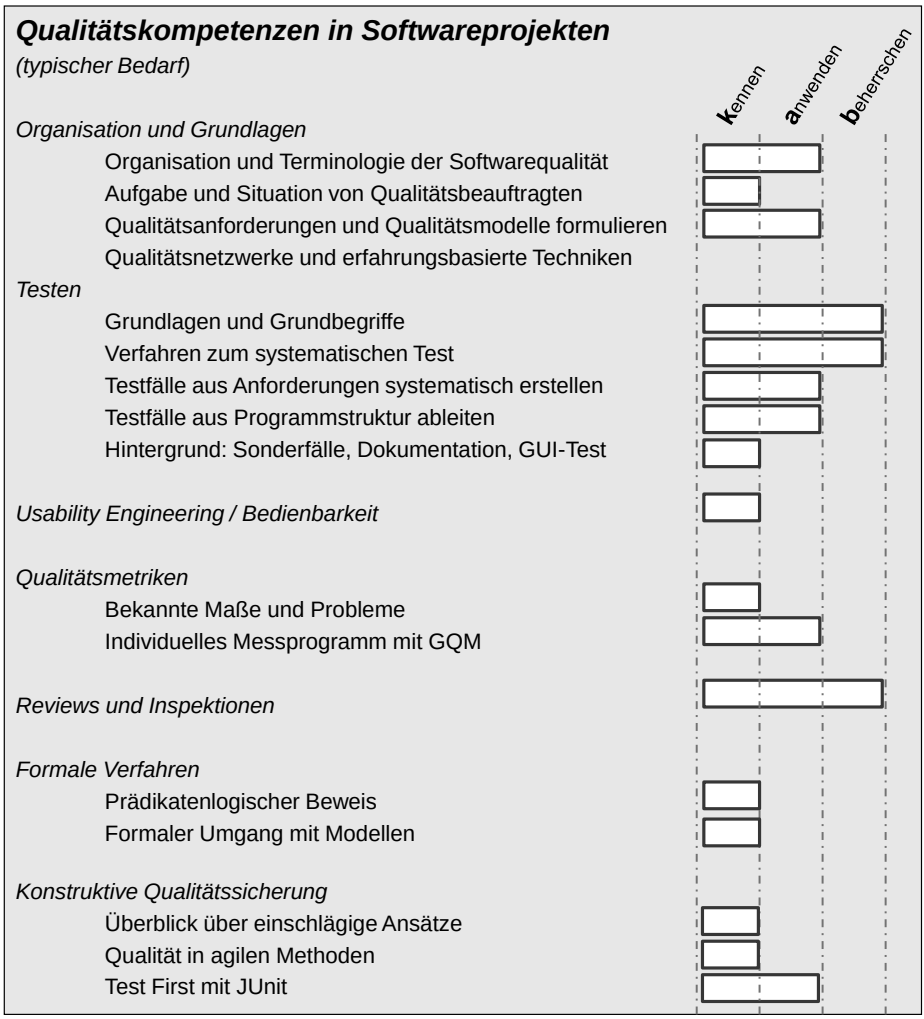
Um eine Technik zu beherrschen, muss man sie mehrfach selbst anwenden. Hier sind die Grenzen einer Einführung erreicht; dieses Buch soll eine Ermunterung an den Leser sein, die Verfahren auch einzusetzen. Für manche Themen gibt es typischerweise mehr Bedarf als für andere.

Tabelle 1–1 zeigt, von welchem Bedarf man bei normalen Projekten ausgehen kann. Dabei ist natürlich jedes Projekt etwas anders gelagert, die Tabelle kann nur einen ersten Überblick geben. Wer in seinem Projekt beispielsweise modellbasiert arbeitet, wird sich natürlich auf diesem Gebiet vertiefen müssen, während das in anderen Projekten unnötig ist. Aber jeder Projektteilnehmer, ob Entwickler, Projektleiter oder Qualitätsbeauftragter, sollte mit den Themen aus Tabelle 1–1 einigermaßen vertraut sein. Idealerweise gilt das auch für Manager und qualitätsbewusste Kunden von Softwareprojekten.

Die Auswahl und Gewichtung der Themen orientiert sich an deren praktischer Bedeutung, also an Tabelle 1–1. Auch für Themen, die man nicht ständig braucht, wird ein Überblick geboten, damit man die Grundlagen kennt und sie bei Bedarf einordnen kann. Bei Themen, die in Projekten häufiger verlangt werden, ist auch die Darstellung detaillierter.

Sicher wäre es ideal, alle Themen zu *beherrschen*. Aber der Weg dorthin ist oft weit und steinig. Nach vielen Jahren ist er mit guten und schlechten Erfahrungen gepflastert. Anfangs darf es auch etwas weniger sein: Auf Spezialgebieten wie dem Usability Engineering ist schon viel erreicht, wenn man sich so weit auskennt, dass man mit Spezialisten vernünftig zusammenarbeiten kann.

Darum geht es hier. Dieses Buch soll einen guten Start geben und Orientierungshilfe für diesen Weg liefern.



Tab. 1–1 *Wie gut man die Themen für den Projektalltag kennen sollte*

1.5.2 Die Reihenfolge der Themen

Die Qualitätssicherung wird oft systematisch in drei Bereiche eingeteilt: analytische, konstruktive und organisatorische Maßnahmen. Analytische Maßnahmen suchen Fehler, damit man sie beheben kann. Konstruktive versuchen schon bei der Entwicklung, Probleme gar nicht erst entstehen zu lassen. Und organisatorische Maßnahmen bilden den Rahmen um die ersten beiden: Sie schaffen die Voraussetzungen, damit diese wirken können.

Diese Einteilung taucht natürlich auch in diesem Buch auf, die Themen sind aber nach einem anderen Kriterium angeordnet: Wer die Kapitel der Reihe nach

liest, wird von ihnen nach und nach an die Softwarequalität herangeführt. Nicht Techniken und Werkzeuge sollen den ersten Eindruck bestimmen, sondern der Zusammenhang, die menschliche Komponente – und was Projektbeteiligte zuerst einmal bewegt: Bin ich hier überhaupt richtig?

Auf das Testen werfen wir zuerst nur einen kurzen Blick. Die Herangehensweisen strahlen auch in andere Bereiche der Softwarequalität aus, aber die Details des Testens müssen noch etwas warten. Am Schluss kommen dann speziellere Themen zu ihrem Recht: Usability oder Bedienbarkeit und agile Methoden, immer in Bezug auf Softwarequalität.

Wer aber nicht der Reihe nach lesen will oder wer schnell etwas sucht, kann direkt in die jeweiligen Kapitel einsteigen.

1.6 Bedeutung von Softwarequalität

An Softwarequalität denkt man im normalen Leben eigentlich nur, wenn etwas damit nicht stimmt:

Man hat eine Onlineüberweisung abgeschickt, da bleibt das Programm plötzlich stecken und reagiert nicht mehr. Ist das Geld jetzt überwiesen? Wenn nicht, steht eine Mahnung ins Haus. Also lieber einfach noch einmal überweisen. Aber halt! Wenn die Überweisung doch schon erfasst wurde, würde das zu einer Doppelüberweisung führen – mit überzogenem Konto beim Absender und viel Arbeit, bis man alles rückgängig gemacht hat.

Es kann natürlich immer einmal Verbindungsprobleme geben, wie in diesem Beispiel. Aber es hat mit Softwarequalität zu tun, wie aufwendig es ist festzustellen, ob die Überweisung gerade noch getätigt wurde oder gerade nicht mehr. Definitiv ein Qualitätsproblem liegt vor, wenn das Geld zwar beim Absender abgebucht, aber nicht beim Adressaten gutgeschrieben wurde (unvollständige Transaktion). Wenn eines dieser Probleme mit einer größeren Summe passiert, kann der Ärger entsprechend groß werden. Dann wechselt ein Kunde schon einmal die Bank.

Noch ein Beispiel: Im Internet können auch scheinbar kleine Ungewöhnlichkeiten zu Verlusten führen: Beispielsweise gibt es Onlineshops, die keinen Einkaufswagen anbieten oder nicht wie gewohnt damit umgehen. In einem bestimmten Online-Bekleidungsgeschäft sucht man beispielsweise vergeblich nach einem Einkaufswagen. Mancher Kunde und manche Kundin geben nach einigem Suchen entnervt auf und kaufen eben anderswo ein. Dabei heißt dieser Mechanismus hier einfach nicht Einkaufswagen, sondern »Tasche«. Nun kann man sich streiten, ob man in einem realen Bekleidungsgeschäft eher mit einem Einkaufswagen oder einer Tasche einkauft. Wenn der Laden wegen einer schöneren Formulierung aber Kunden verliert, ist der Streit rasch entschieden. Außerdem würde man ja im Laden hoffentlich auch nicht alles in seine Tasche stecken,

solange man noch nicht bezahlt hat. Eine scheinbare Kleinigkeit, die also mehr Verwirrung als Freude stiftet.

Diese Fälle sind real und vor kurzem passiert. Nicht jeder würde sie spontan auf schlechte Softwarequalität zurückführen, aber diese steckt auch dahinter, wenn hier und in tausenden von ähnlichen Fällen Kunden abwandern, Vertrauen verspielt wird und Geld verloren geht. Jeder kennt ähnliche Fälle.

Natürlich liest man in der Zeitung auch von den spektakulären Pannen und Katastrophen: das Computersystem einer Bank, das abstürzt und einen Tag lang nicht verfügbar ist. Noch ein oder zwei Tage länger, und die Bank wäre Bankrott gegangen. Die Handyfirma, die hunderte von Millionen Euro Verlust macht, weil der Warnton einer Baureihe unter gewissen Umständen so laut ertönt, dass er das Gehör schädigen kann – ein Softwareproblem. In einem anderen Fall werden Krebspatienten falsch bestrahlt, weil ein Gerät einen Softwarefehler hat. Auch das System rund um die Gesundheitskarte muss zeigen, dass die Daten darauf sicher gespeichert sind und nicht von Unbefugten angezapft werden können. Diese Liste ließe sich endlos verlängern; überall sind finanzielle und manchmal gar gesundheitliche Schäden die Folge.

Man sollte dabei aber nicht vergessen, dass auch die **positive Liste** eindrucksvoll und es durchaus möglich ist, mit guter Softwarequalität Kunden zu binden und sogar anzulocken. Niemand würde Bücher bei Internethändlern kaufen, wenn ständig Bestellungen verloren gingen oder verfälscht würden – oder auch nur, wenn die entsprechenden Websites nicht so einfach zu bedienen wären. Das ist nicht zufällig so gekommen, sondern es wurde mit intensivem Ringen um gute Bedienbarkeit, Zuverlässigkeit und Korrektheit erreicht. Ein Triumph der Softwarequalität, wenn auch im Stillen.

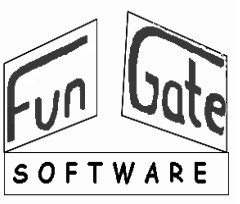
Wer sich heute in ein modernes Auto der Oberklasse setzt, bemerkt gar nicht, wie viel Software ihn umgibt. Vom Navigationssystem ganz abgesehen, basieren auch Bremse und Motorsteuerung, dynamische Servolenkung und Airbag, Stabilitätspaket und Klimaanlage auf Software. Sie funktioniert zuverlässig, sicher und unauffällig in Millionen von Fahrzeugen. Eine einzige Panne mit dem Bremsassistenten hätte so gravierende Folgen, dass Softwarequalität ernst genommen werden muss. Zur Freude der Fahrer.

Auch diese Liste ist lang; nur muss man etwas nachdenken, bevor man im eigenen Umfeld Beispiele findet, in denen gute Softwarequalität großen Nutzen gestiftet hat. Man hält es wohl auch bei so komplexen Systemen für »selbstverständlich«, dass sie »normal funktionieren«.

Nichts könnte aber *weniger* selbstverständlich sein. Softwarequalität ist ein herausforderndes und deswegen auch spannendes Gebiet. Die Auswirkungen auf Wirtschaft, Gesundheit und privates Wohlergehen kann man kaum überschätzen. Grund genug, sein Bestes zu geben.

1.7 Wie Q zur Softwarequalität kam

Q hat Informatik studiert und ist vor kurzem fertig geworden. Wie schon oft ist Q am Frühstückstisch in den Stellenteil der Zeitung vertieft. Bisher waren nicht so viele interessante Sachen dabei. Eine Bewerbung hat Q auch schon geschrieben, aber eher halbherzig. Ganz so eilig ist es nicht mit der ersten richtigen Stelle, schließlich hat Q noch einen Programmierjob, der seit vier Semestern für ein wenig finanziellen Spielraum sorgt. Aber eine richtige Dauerstelle ist das nicht. Eine Stellenanzeige spricht Q an, weil sie nach einer anspruchsvollen Tätigkeit klingt, in der man nicht ständig nur programmieren muss:



FunGate Software
ist ein mittelständisches Software-Unternehmen.

Wir erstellen Individualsoftware für Global Player und für Spezialisten mit hohen Anforderungen. Qualität wird bei uns nicht nur groß geschrieben, sondern in den Projekten täglich gelebt. Zum Ausbau unserer Softwarequalitäts-Mannschaft suchen wir eine/n

Software-Qualitätsbeauftragten (m/w)
am Standort Hannover

zum Einsatz in den innovativen Projekten unseres Hauses. Sie planen und überwachen selbstständig Qualitätsmaßnahmen in der Entwicklung eingebetteter und administrativer Software. Sie stimmen sich mit dem Projektleiter eng ab und berichten direkt dem Qualitätsverantwortlichen des Bereichs.

Wir erwarten einen überdurchschnittlichen Universitätsabschluss in Informatik und das Fingerspitzengefühl, das den erfolgreichen Qualitätsexperten auszeichnet. Für außergewöhnliches Engagement bieten wir außergewöhnlich gutes Betriebsklima und eine angemessene Bezahlung.

FunGate - Hannover

Bewerben Sie sich mit den üblichen Unterlagen …

Abb. 1–1 *Fiktive, aber typische Stellenanzeige für Qualitätsbeauftragte*

Im Studium hat Q diverse Vorlesungen über Software Engineering gehört. Auch das Thema Softwarequalität findet Q interessant; in einem studentischen Projekt war Q für die Qualitätssicherung zuständig und weiß daher, wie schwierig das ist. Außerdem klingt es vielversprechend, gleich mit dem Projektleiter auf Augenhöhe verhandeln zu können – und notfalls sogar noch jemanden in der Bereichsleitung zu kennen. Vor allem glaubt Q, über ziemlich gute soziale Fähigkeiten zu verfügen; nicht erst seit dem Softwareprojekt im vorletzten Semester.

Die Anzeige aus Abbildung 1–1 ist genauso fiktiv wie die Firma FunGate, aber die Anforderungen an Qualitätsbeauftragte stammen aus realen Unternehmen. Der Anzeigentext verrät schon einiges über die Erwartungen und Grundkonzepte, die in der Firma herrschen. Nicht überraschend ist wohl, dass ein Informatik-Hintergrund sehr wichtig ist. Eher verblüffend könnte dagegen sein, dass darüber hinaus von Fingerspitzengefühl die Rede ist, also von »*soft skills*«. Die geforderten »weichen Fähigkeiten« zu haben ist hier eine harte Anforderung! Wer sie nicht mitbringt, wird große Schwierigkeiten haben – und sollte daher die Stelle nicht bekommen. Bewerber sollten die Formulierung in Stellenanzeigen unbedingt sehr ernst nehmen, denn das Wenige, das dort steht, ist meist sehr gut überlegt.

Offenbar ist FunGate keine ganz kleine Firma. Eine sehr kleine Firma hätte kaum explizit einen Qualitätsbeauftragten gesucht; dort wird diese Aufgabe meistens nebenher von Entwicklern übernommen. Q erinnert sich an die Anmeldung bei einem Telefonanbieter. Man musste ein langes Formular auf dem Bildschirm ausfüllen. Links konnte man sehen, welche Schritte noch bevorstanden. Insgesamt waren es 17 Schritte. Im vorletzten Schritt, vor der endgültigen Zustimmung zum Anbieterwechsel, wollte Q noch einmal schnell seine Antwort von Schritt 2 nachsehen und klickte auf diesen Schritt. Wie erwartet wurden die dortigen Angaben angezeigt. Leider konnte Q dann aber nicht mehr zu Schritt 16 zurückspringen und musste alles noch einmal eingeben. So etwas müsste man doch anständiger hinbekommen, denkt Q. Das wäre schon eine lohnende Herausforderung für überzeugte Qualitätsleute, da könnte man etwas für die Kunden bewirken!