

Eine Forschungsagenda für Softwarequalität

Stefan Wagner¹, Manfred Broy¹, Florian Deißböck¹, Michael Kläs², Peter Liggesmeyer², Jürgen Münch², and Jonathan Streit³

¹ Institut für Informatik, Technische Universität München,
Boltzmannstr. 3, 85748 Garching b. München

² Fraunhofer IESE,
Fraunhofer-Platz 1, 67663 Kaiserslautern

³ itestra GmbH,
Ludwigstrasse 35, 86916 Kaufering

Zusammenfassung Softwarequalität ist für die Praxis, gerade in einem Technologiestandort wie Deutschland, eine große Chance aber auch eine enorme Herausforderung. Qualitätsfragen haben einen starken wirtschaftlichen Einfluß, aber durch die Komplexität und Vielschichtigkeit von Qualität sind sie schwer greifbar. Diese Papier stellt das Ergebnis der Diskussionen von Qualitätsexperten aus Forschung und Praxis dar, welche Fragestellungen in der Zukunft am dringlichsten im Bereich der Softwarequalität bearbeitet werden müssen, damit sowohl der Stand der Forschung als auch der Technik entscheidend verbessert werden kann.

1 Einleitung

Die hohe Qualität von Ingenieursprodukten ist ein traditionelles deutsches Markenzeichen, das immer wieder zur Auszeichnung als Exportweltmeister verhilft. Leider ist dieser Ruf stark auf Maschinenbau-lastige Unternehmen beschränkt und überträgt sich auf deutsche Softwareprodukte nicht immer. Interessanterweise wird aber gerade auch im Maschinenbau Software immer wichtiger. Nicht nur in Deutschland, sondern weltweit, steht Software im Ruf häufig Qualitätsprobleme zu haben.

Dem gegenüber steht aber die hohe wirtschaftliche Bedeutung der Qualität von Software [1]. Beispielsweise wird in [2] dargelegt, dass der Erfolg der deutschen Softwareindustrie von ihrer Fähigkeit mit Qualität umzugehen abhängt. Softwarefehler haben immer wieder hohe Kosten durch Rückrufaktionen verursacht und können auch volkswirtschaftlich enorme Schäden anrichten. Auch um im globalen Konkurrenzkampf bestehen zu können wir eine Differenzierung über Qualität immer wichtiger. Es besteht also ein umfangreicher Handlungsbedarf sowohl in der Forschung, als auch in der Praxis.

Jedoch stellt sich immer noch für jeden, der sich mit Softwarequalität beschäftigt zuvorderst die Frage: Was ist Qualität? Es gibt verschiedenste Blickwinkel, aus denen Produktqualität betrachtet werden kann. Garvin legte das in seinem bekannten Papier [3] eingängig dar. Solch grundlegende Fragen müssen weiterhin betrachtet werden. Qualitätsmodelle sind das Mittel der Wahl, um genauer darzustellen, was für Attribute für Softwarequalität entscheidend sind. Dabei gibt es

bestehende Standards, wie die ISO-Norm 9126 [4], firmenspezifische Standards, aber auch neuere akademische Ansätze [5].

Aber auch in diesen Qualitätsmodellen gibt es verschiedenste Fragen, die noch zu beantworten sind. Beispielsweise ist die Aggregation von Teilergebnissen in Qualitätsmodellen zu höherwertigen Aussagen immer noch nicht eindeutig geklärt. Eine andere Frage, die sich durch die Diskussionen zieht, ist der Zusammenhang zu Kosten/Nutzen-Modellen. Müssen Kosten im Zusammenhang mit Qualität berücksichtigt werden?

Dieses Papier hat das Ziel die dringendsten Fragen im Umfeld von Softwarequalität nicht nur, aber insbesondere, in Deutschland zusammenzutragen. Aus diesen Forschungsfragen leiten wir dann eine Forschungsagenda für Softwarequalität ab, die der Community aus Forschung und Praxis eine gemeinsame Zielrichtung geben soll.

Wir haben hierzu im Workshop “Software-Qualitätsmodellierung und -bewertung” im Zusammenhang mit der SE 2008 in ausführlichen Diskussionen drei große Herausforderungen identifiziert. Erstens stellt die Anpassung von Qualitätsmodellen ein noch weitgehend unbearbeitetes, aber praktisch relevantes Thema dar, da Software in verschiedensten Domänen und Kontexten eingesetzt wird. Zweitens ist es wichtig, eine klare Strategie bei der Einführung von Qualitätsmodellen in Unternehmen zu haben, da von der Akzeptanz bei den Entwicklern der Erfolg des Modells abhängt. Schließlich drittens ist das Messen von Qualität, selbst mithilfe von Qualitätsmodellen, noch oft unsystematisch. Hierzu fehlen noch die richtigen Maße und Meßmethoden.

In den Qualitätsmodellen läßt sich meist beobachten, dass nur Defizite beschrieben werden, aber keine Möglichkeiten und Stellschrauben, um diese zu beeinflussen oder zu beheben. Weiterhin werden in verbreiteten Modellen Kosten nicht explizit berücksichtigt, obwohl dies offensichtlich ein wichtiger Aspekt in ihrer Anwendung in kommerziellen Projekten ist. Weiterhin wurde in verschiedenen Projekten die Erfahrung gemacht, dass in der konkreten Anwendung Qualitätsmodelle sehr groß werden können, da sie umfangreiche Informationen zu verschiedensten Qualitätsaspekten abdecken müssen. Trotz dieser Kritikpunkte wurde aber auch die Erfahrung gemacht, dass Qualitätsmodelle sehr wertvolle Informationen liefern, die im Entwicklungsprozess noch besser nutzbar gemacht werden müssen.

In den folgenden Abschnitten 2–4 stellen wir die drei Hauptthemen im Detail dar. Im Abschnitt 5 fassen wir die Forschungsfragen in einer Forschungsagenda zusammen und schließen mit einer Zusammenfassung in Abschnitt 6.

2 Anpassung von Qualitätsmodellen

Software ist in den verschiedensten Bereichen und Domänen im Einsatz, wodurch auch sehr unterschiedliche Anforderungen an die Software-Systeme gestellt werden. Dies trifft auch auf Qualitätsanforderungen zu, die im Zusammenhang zu einem Qualitätsmodell definiert werden müssen. Die normierten Modelle, wie die ISO 9126, bieten heute keine konkreten Vorgaben, wie sie an diese Gegebenheiten

anzupassen sind. Dies ist ein starker Hinderungsgrund gegen die operationalisierte Verwendung von Qualitätsmodellen in Software-Projekten. Wir haben hier bewußt das Wort “Anpassung” gewählt, da die Alternative “Tailoring” oft nur für das Weglassen von Teilen, aber nicht für die Erweiterung steht, was aber in diesem Kontext auch wichtig ist.

2.1 Erfahrungen und Herausforderungen

Es existieren bereits eine Reihe von Qualitätsmodellen aus verschiedensten Quellen. Modelle finden sich in der wissenschaftlichen Literatur [5–9] und in verschiedenen allgemeinen und Domänen spezifischen Standards wie [4, 10]. Darüberhinaus definieren Unternehmen, die sich weitergehend mit Softwareentwicklung beschäftigen typischerweise firmenintern Richtlinien, die auf die konkreten Anforderungen in den jeweiligen Projekten angepaßt sind. Schließlich kann auch der Einsatz von Mess-Werkzeugen ein entsprechendes Qualitätsmodell implizieren.

Grundsätzlich bewegt man sich bei Qualitätsmodellen für die Software-Entwicklung innerhalb des Spektrums zwischen der kompletten Eigendefinition des Modells und der direkten Nutzung eines bestehenden Modells. Ersteres erlaubt das genaue Zuschneiden auf die konkreten Bedürfnisse, bedarf aber auch eines enormen Aufwands und entsprechender Expertise. Letzteres hingegen minimiert die Aufwände, kann aber erhebliche Probleme verursachen, da das eingesetzte Modell eventuell nicht auf die eigentlichen Gegebenheiten im Unternehmen oder Projekt paßt.

Aus diesen Gründen wird ein Qualitätsmodell als sinnvoll erachtet, das ein normiertes Grundmodell definiert, aber auch vorgegebenen Prozess umfasst, der die Anpassung an die eigenen Anforderungen und Gegebenheiten ermöglicht. Zur Entwicklung eines solchen Modells, müssen aber noch einige grundlegende Fragen geklärt werden.

Zuvorderst muss festgelegt werden, was genau unter einem Qualitätsmodell zu verstehen ist. Dazu existieren sehr unterschiedliche Ansichten. Definitionen aus Standards [11] helfen hier aufgrund ihrer sehr allgemeinen Formulierung nur wenig weiter. Für viele sind die Definition und Verfeinerung von “-ilities”, wie *Reliability* oder *Functionality* der ISO 9126, ausreichende Bestandteile eines Qualitätsmodells. Andere Ansätze gehen diesbezüglich wesentlich weiter und beschreiben auf sehr detaillierter Ebene Wirkzusammenhänge [5] oder den Zusammenhang zu automatisierten Qualitätsüberprüfungen [12]. Daher wäre also ein einheitliches Meta-Modell wünschenswert, das notwendige und optionale Bestandteile eines Qualitätsmodells beschreibt.

Die Methoden *Factors-Criteria-Metrics* (FCM) [13] und *Goal/Question/Metric* (GQM) [14] beschreiben stärker diese Meta-Ebene sowie ein generisches Vorgehen zur Entwicklung angepasster Messsysteme. Nur der SQUID-Ansatz [15] beschreibt derzeit ein Vorgehen zur spezifischen Definition von Qualitätsmodellen, bietet aber weniger Details zu deren Operationalisierung. Es fehlt heute eine detaillierte Untersuchung wo überall in Qualitätsmodellen Möglichkeiten zur Anpassung berücksichtigt werden und welche Kriterien diese Anpassungen

notwendig machen. Es ist also eine Herausforderung Einflussfaktoren auf die Anpassung zu finden und ihre Auswirkungen zu untersuchen.

2.2 Einsatz von Qualitätsmodellen

Um eine sinnvolle Anpassung von Qualitätsmodellen zu ermöglichen, muss zuerst noch der Einsatz von Qualitätsmodellen in der Software-Entwicklung diskutiert werden. Generell stellen Qualitätsmodelle eine Abstraktion von Qualitätsmerkmalen für Software dar. Sie verallgemeinern von Qualitätsdefekten und bieten auch eine Grundlage für Messungen. Dafür sollte eine Hierarchie an Nutzungsszenarien entwickelt werden, die jeweils aufeinander aufbauen. Beispielsweise könnte das Modell zuerst verwendet werden, um Qualität zu *verstehen*. Die komplexen und vielschichtigen Aspekte von Software-Qualität sollten in einem Qualitätsmodell dabei so beschrieben sein, dass das Verständnis verbessert wird. Darauf aufbauend kann Qualität besser *definiert* werden. Nur ein strukturiertes Vorgehen, wie es ein Qualitätsmodell bietet, erlaubt überhaupt eine sinnvolle Definition von Qualität. Diese Definitionen können wiederum als Ausgangspunkt für die *Analyse* von Qualität verwendet werden. Das Modell kann hierbei Indikatoren und Metriken definieren und bei der Interpretation der Messergebnisse unterstützen. Weiterhin sollte das Modell auch Wege zur *Verbesserung* der Qualität aufzeigen und im Idealfall helfen Qualität *vorherzusagen*.

Grundsätzlich lässt sich festhalten, dass Qualitätsmodelle kein Selbstzweck sein dürfen, so wie auch das Messen von Qualität kein Selbstzweck sein kann. Vielmehr müssen sie immer als Entscheidungsgrundlage im Software-Projekt dienen. Beispielsweise können sie helfen eine *Make or Buy*-Entscheidung zu treffen oder festzulegen, ob eine Software reif zur Auslieferung ist.

2.3 Empfehlungen

Ein wichtiger Punkt in der Anpassung von Qualitätsmodellen ist die Identifizierung der Anpassungsdimensionen bzw. der Einflussfaktoren. Insbesondere spielt es eine Rolle, was die Ziele des Unternehmens oder Projekts sind. Danach muss untersucht werden, wie sich dies im Qualitätsmodell auswirkt und das Modell entsprechend angepasst werden. Insbesondere ist wichtig diese Anpassungen zu operationalisieren. Schließlich muss eine Validierung stattfinden, also überprüft werden, ob die Anpassungen sinnvoll sind. Dies kann beispielsweise durch Rückkopplung mit den Entwicklern und Qualitätssicherungspersonal geschehen.

Für eine längerfristige Anwendung ist auch die Integration in das verwendete Prozessmodell notwendig. Es muss beispielsweise festgelegt werden, an welchen Entscheidungspunkten (Quality-Gates) das Modell zum Einsatz kommt oder wann Messungen stattfinden. Zuletzt kann noch eine konkrete Anreicherung um Kosten erfolgen, die monetäre Schätzungen und Bewertungen erlaubt.

Es ist unwahrscheinlich, dass zu Beginn eines Projekts entweder die Qualitätsanforderungen oder die Qualitätsmodelle vollständig verfügbar sind. Deshalb scheint ein *Bootstrapping*-Verfahren am sinnvollsten: Anforderungen und

Qualitätsmodelle werden im Projektverlauf iterativ und inkrementell abgeglichen [16, 17]. Dabei werden auch Prioritäten und Gewichtungen der einzelnen Teile des Qualitätsmodells festgelegt, was wiederum das Budget für Qualität bestimmen könnte, wie im *Activity-based Costing*. Bei festgelegtem Budget könnte dieses natürlich wiederum die Gewichtungen bestimmen.

2.4 Forschungsfragen

Aus diesen Herausforderungen und Empfehlungen können nun eine Reihe von noch offenen Forschungsfragen abgeleitet werden, deren Beantwortung als notwendig für eine praktische Anpassung von Qualitätsmodellen gesehen wird. Zuerst muss untersucht werden, welches Spektrum an Qualitätsmodellen zur Zeit existiert. Dies ist notwendig, um zu entscheiden, wie flexibel die Anpassungsmechanismen sein müssen. Ähnlich dazu muss analysiert werden, welche Komponenten ein Qualitätsmodell notwendigerweise und optional umfasst. Dazu gehört einerseits die Struktur eines Qualitätsmodells, andererseits auch welche Inhalte zwingend und welche optional sind. Insbesondere im Bezug auf Kosten stellt sich die Frage, welche Beziehung zwischen bestehenden Kostenrechnungs- und Qualitätsmodellen besteht, da dies ein wichtiger Faktor in der Anpassung sein könnte.

Eine weitere wichtige Frage ist, wie groß die Variabilität in Qualitätsmodellen über Domänen, Projekte, Technologien hinweg überhaupt ist. Aus den dargestellten Erfahrungen ergibt sich eine signifikante Variabilität, dies wurde aber noch nicht wissenschaftlich untersucht. Diese Information ist jedoch wichtig ist um einen brauchbaren Anpassungsmechanismus abzuleiten. Daran anschließend ist noch unklar, wie ein umfassendes und umfangreiches Qualitätsmodell (nach seiner Anpassung) validiert werden kann. Eine vollständige empirische Überprüfung kann aufgrund des dafür nötigen Aufwands sicher ausgeschlossen werden.

3 Einführung von Qualitätsmodellen

Unklar ist bisher, welches die nötigen Schritte zur Einführung von Qualitätsmodellen in Unternehmen sind. Dies ist nicht nur durch die allgemeine Uneinigkeit bzgl. einer geeigneten Operationalisierung von Qualitätsmodellen bedingt, sondern vor allem durch die langfristige Natur der Fragestellungen, die von ihnen adressiert werden. Abstrakt betrachtet, hat die Einführung von Qualitätsmodellen das Ziel, "Qualitätseigenschaften die Freiwilligkeit zu nehmen". Das heißt, ähnlich wie bei funktionalen Anforderungen soll erreicht werden, dass Qualitätsanforderungen selbstverständlich umgesetzt und nicht nach Belieben weggelassen oder verändert werden.

3.1 Erfahrungen und Herausforderungen

Qualitätsmodelle werden verwendet um in einem gegebenen Projekt- oder Unternehmenskontext ein gemeinsames Verständnis von Produkt-Qualität zu eta-

blieren. Ihr Einsatz erfolgt meist im Rahmen von langfristig angelegten Qualitätsinitiativen und nur selten zur Behebung akuter Probleme. Aufgrund dieser Langfristigkeit ergeben sich eine Reihe von Problemen bzgl. der Motivation der beteiligten Personen, die den unmittelbaren Nutzen der Initiativen nicht einfach erkennen können. Verstärkt wird dieses Problem durch den Umstand, dass solche Qualitätsinitiativen von den meisten Beteiligten, zumindest zu Beginn, eine gewisse Mehrarbeit erfordern, von Managementseite jedoch selten die entsprechenden Ressourcen zur Verfügung gestellt werden.

Darüber hinaus wurde mehrfach festgestellt, dass nicht alle Projektbeteiligten über die entsprechende Ausbildung verfügen, die es Ihnen erlauben würde, den langfristigen Nutzen von Qualitätsinitiativen zu erkennen bzw. zu realisieren, was die Nicht-Einhaltung von Qualitätsstandards zur Folge hat.

Ein weiteres Problem bei der Einführung von Qualitätsmodellen können sich durch die organisatorische Trennungen von Mitarbeitern unterschiedlicher Prozessphasen ergeben. So ist es u. U. schwierig Entwickler, die nie mit Wartungsaufgaben betraut werden, von den Vorteilen von lesbarem Code zu überzeugen.

Neben diesen motivatorisch-personellen Problemen stellt die in Abschnitt 2 diskutierte Inflexibilität heutiger Qualitätsmodelle eine Schwierigkeit dar, da sie eine Anpassung auf die projektspezifischen Qualitätsbedürfnisse erschwert und damit weder zur Effektivität noch zur Effizienz des Qualitätsmanagements beiträgt.

3.2 Empfehlungen

Dieser Inflexibilität kann mit den im letzten Abschnitt diskutierten Anpassungsmechanismen entgegnet werden. Aus Sicht der Einführung von Qualitätsmodellen sind hierbei folgende Dimensionen der Anpassung von Bedeutung:

- *Projekt-spezifische Anpassung.* Eine Anpassung des Qualitätsmodells an das Projekt ist wichtig um den Projektbeteiligten zu demonstrieren, dass sich bei dem eingesetzten Qualitätsmodell nicht um einen bürokratischen Selbstzweck sondern um ein Mittel zur langfristigen Steigerung der Projekt-Produktivität handelt.
- *Mitarbeiter-spezifische Sichten.* Neben der tatsächlichen Anpassung sollte das Qualitätsmodell die Möglichkeit bieten für die unterschiedlichen Projektbeteiligten unterschiedliche Sichten zu definieren. Dies ermöglicht es, jedem Projekt-beteiligten nur den für ihn relevanten Anteil zur Verfügung zu stellen und ihn nicht mit unnötiger Information zu belasten.
- *Konfiguration der Überprüfungsverfahren.* Speziell automatische Überprüfungsverfahren für Qualitätskriterien leiden oft unter einer hohen Zahl von *False Positives*, die schnell zu einer Demotivation der Projektbeteiligten führen können. Daher ist es notwendig, Überprüfungswerkzeuge und -methoden so auf den jeweiligen Kontext anzupassen, dass die Anzahl der *False Positives* auf ein erträgliches Maß (<5%) beschränkt wird.

Darüber hinaus wird empfohlen, die Einführung von Qualitätsmodellen möglichst behutsam durchzuführen, um die Beteiligten nicht durch eine Fülle von

Neuerung zu überfordern. Als äußerst hilfreich hat es sich erwiesen, einen unmittelbaren Nutzen für einzelne Beteiligten zu identifizieren, da dieser oft dazu beiträgt, die Bereitschaft der Beteiligten auch für langfristige Neuerungen, ohne unmittelbaren Nutzen, zu erhöhen. Jenseits dieser unmittelbaren Nutzen, muss natürlich dafür gesorgt werden, dass alle Beteiligten ein entsprechendes Verständnis für die Ziele, die mit Hilfe von Qualitätsmodellen erreicht werden sollen, haben und sich insbesondere über die langfristigen Konsequenzen einer Nicht-Berücksichtigung von Qualitätseigenschaften bewusst sind.

3.3 Forschungsfragen

Aus diesen Erkenntnissen und Empfehlungen ergeben sich eine Reihe von Forschungsfragen die bzgl. der Einführung von Qualitätsmodelle relevant sind. Neben den eher technischen Fragen bzgl. der verbesserten Anpassbarkeit und besseren Werkzeugunterstützung von Qualitätsmodellen spielen hierbei vor allem sozio-psychologische Aspekte (*Change Management*) und ein tiefgehendes Verständnis von Kosten-Nutzen-Zusammenhängen eine Rolle.

Hierbei wäre insbesondere eine strukturierte Aufarbeitung der Erkenntnisse aus anderen Disziplinen, z. B. Prozesseinführung, ein vielversprechender Startpunkt um besser zu verstehen, welchen Probleme die Einführung von Qualitätsmodellen auf einer sozio-psychologischen Ebenen begegnet.

Darüber hinaus würde eine systematische Aufarbeitung Kosten-Nutzen-Zusammenhänge im Bereich der Software-Qualität sicherlich die Einführung von Qualitätsmodellen deutlich erleichtern. Durch die Bereitstellung von Kosten-Nutzen-Modellen, die in der Lage sind quantitative Aussagen bzgl. des langfristigen Nutzens von Qualitäts-Aktivitäten zu treffen, könnten viele Aussagen, die heute eher anekdotischer Natur sind, auf ein sicheres Fundament gestellt werden und damit zur sachlichen Diskussion beitragen.

Jenseits dieser konkreten Vorschläge, ist es sinnvoll, die Qualitätsthemen noch stärker in der Ausbildung zukünftiger Fachkräfte zu berücksichtigen und damit langfristig eine Situation zu erreichen, in der Qualitätsmodelle gar nicht eingeführt werden müssen, da sie integraler Bestandteil eines jeden Softwareprojekts sind.

4 Qualitätsmessungen mit Qualitätsmodellen

Die Definition von Qualitätsmodellen und deren Nutzung bei der Qualitätsplanung und -sicherung im Software- und Systems-Engineering erfordert die Quantifizierung von Soll- und Istwerten. Die Auswahl geeigneter Metriken ist dabei von einer Menge von Faktoren (wie z.B. dem Entwicklungskontext) abhängig, ohne deren Beachtung Qualitätsmodelle einen Großteil ihres Nutzens einbüßen. Dies macht die Nutzung von derzeit existierenden Standards für Software-Qualität schwierig, da diese oftmals durch ein hohes Abstraktionsniveau Variabilität vermeiden und ihr praktischer Nutzen somit äußerst fragwürdig ist. Grundvoraussetzung für ein messbasiertes Qualitätsmanagement ist die Schaffung der hierfür

notwendigen Akzeptanz in Unternehmen und Organisationen. Die Einführung von Messprogrammen bzw. die Änderung existierender Messverfahren ist ein aufwändiger Prozess, der sorgfältig geplant und im Vorhinein auf den zu erwartenden Nutzen analysiert werden muss. Im Rahmen des Workshops wurden folgende Themen erörtert und in der angegebenen Reihenfolge bezüglich ihrer Dringlichkeit aus praktischer Sicht priorisiert:

1. Wie lassen sich Qualitätsmessungen in Unternehmen und Organisationen motivieren?
2. Wie lässt sich Qualität definieren und an den Entwicklungskontext anpassen?
3. Wie werden Qualitätsmessungen eingeführt?
4. Welche Messverfahren und Werkzeuge können zur Umsetzung genutzt werden?

4.1 Erfahrungen und Herausforderungen

Die Notwendigkeit von Software-Messungen und Aktivitäten für Softwarequalität ist bis heute in vielen Unternehmen und Organisationen nicht selbstverständlich. Vielfach wird der Aspekt der Software-Qualität ignoriert oder findet erst Beachtung, wenn bereits erhebliche Probleme aufgetreten sind. Häufig finden Qualitätsaspekte erst sehr spät im Entwicklungsprozess Aufmerksamkeit, in der Regel im Zusammenhang mit ersten Testprozessen. Zu diesem Zeitpunkt bestehen allerdings nur noch sehr begrenzte Möglichkeiten auf die Qualität der resultierenden Produkte Einfluss zu nehmen. Die meisten Qualitätsmängel lassen sich zu diesem Zeitpunkt nicht mehr bzw. nur noch mit sehr hohem Aufwand beheben.

Eine Ursache für das fehlende Bewusstsein für Software-Qualität in Entwicklungsorganisationen ist, dass die wirtschaftliche Bedeutung von Software-Qualität Entscheidungsträgern oftmals unklar ist. Der Bezug von Maßnahmen zur Erzielung qualitativ hochwertiger Software zu höheren Zielen einer Organisation, insbesondere zu Geschäftszielen, ist vielfach nicht explizit dargestellt. Die Kosten von Maßnahmen und Messungen sind dabei klar sichtbar, während der Nutzen in Form von zukünftigen Einsparungen und vermiedenen Folgekosten nicht offensichtlich und schwer quantifizierbar ist.

Hinzu kommt der Gegensatz zwischen der vermeintlich langfristigen Amortisierung von Qualitätsmaßnahmen und kurzfristigem Planungshorizonten auf Managementebene. Erfahrungen aus der industriellen Praxis zeigen, dass oftmals sehr hohe ROI-Erwartungen in Bezug auf Maßnahmen zur Verbesserung der Software-Qualität existieren. Im Vordergrund stehen die Kosten von Software-Qualität und Messungen. Dies hängt damit zusammen, dass Software in vielen Unternehmen als umsetzender und nicht als wertschöpfender Faktor in Bezug auf die Unternehmensziele gesehen wird. Diese Tatsache ist bedenklich, da in vielen Branchen Software mittlerweile der bedeutendste Innovationstreiber ist.

Oftmals ist Software-Qualität auch nicht relevant bei der Vergabe von Aufträgen. Bei Ausschreibungen zählt in der Regel das wirtschaftlich günstigere Angebot. Langfristige Kosten (insbesondere durch Qualitätsmängel verursachte

Kosten) werden bei der Auftragsvergabe nicht oder nicht ausreichend berücksichtigt. Dies ist unter anderem dadurch bedingt, dass die Bedeutung nichtfunktionaler Eigenschaften schwierig zu vermitteln ist und nichtfunktionale Anforderungen schwierig in vertragliche Regelwerke zu integrieren sind. Im öffentlichen Bereich gibt es mittlerweile erste Ansätze, um Wirtschaftlichkeitsuntersuchungen bei IT-Investitionen zu berücksichtigen (insbesondere das WiBe-Verfahren).

Eine weitere Erfahrung in Bezug auf Software-Messungen ist, dass viele Unternehmen sich vorzugsweise an Standards orientieren, um entsprechend des Standes der Technik zu handeln und sich so gegen Risiken (insbesondere juristische) abzusichern. Diese Standards berücksichtigen den Einsatzkontext in der Regel jedoch nur völlig ungenügend. Auch ist eine Präferenz für die Anwendung starrer Metriksätze zu erkennen. Das Bewusstsein, dass im Software-Bereich eine wesentlich höhere Anzahl von Variationsparametern als bei Produktions- und Fertigungsprozessen existiert und somit Metriken und Qualitätsmodelle wesentlich größeren Variationen unterliegen, ist oftmals nicht vorhanden.

Herausforderungen in Bezug auf die quantitative Qualitätsdefinition werden unter anderem darin gesehen, dass es für viele Qualitätsaspekte nur ungenaue, implizit definierte Begriffe gibt (z.B. Wartbarkeit, Performanz, Genauigkeit). Selbst vergleichsweise klar erscheinende Begriffe wie Größe, Kosten oder Aufwand können in unterschiedlichen Umgebungen sehr unterschiedliche Definitionen haben und ihre Quantifizierung kann sich als große Herausforderung gestalten. Insgesamt stellt sich die Ableitung geeigneter Metriken und die Minimierung der zu erfassenden Metrikmenge als schwierig da. Hierzu gehört unter anderem die Beantwortung der Fragen, wie man zielorientiert misst, wie man Einflussfaktoren auf Qualität (z.B. Erfahrung, Teamkohärenz) ermittelt und quantifiziert oder wie man weichere Faktoren (z.B. die Güte von Codekommentaren) misst.

Herausforderungen in Bezug auf den Einführungsprozess von Software-Messungen sind unter anderem ungünstige Randbedingungen (z.B. Lieferdruck, fehlende Ressourcen, geringe Organisationsgröße, fehlende Unterstützung des Managements). Der Mangel an nützlichen Standards und Anleitungen führt darüber hinaus dazu, dass es für Unternehmen schwierig ist, die für sie wichtigen Indikatoren zu ermitteln. Es ist oftmals auch nicht klar, wie bestehende Messprogramme genutzt werden können, um neue Messziele zu beantworten. Darüber hinaus fehlen in der Praxis meist Baselines, sodass ein Vergleichswerte für die Beurteilung von Software-Qualität nicht verfügbar sind.

Beispiele für Herausforderungen hinsichtlich der technischen Umsetzung von Software-Messungen sind die kosteneffiziente Erfassung von Daten (ggf. durch automatisierte Verfahren), die Integration von Messwerkzeugen in den Entwicklungsprozess und die Harmonisierung von Messungen im Falle von verteilter Entwicklung über Organisationsgrenzen hinweg.

4.2 Empfehlungen

Im Folgenden werden ausgewählte Empfehlungen und Hinweise zur erfolgreichen Einführung von Software-Messungen in der Praxis aufgeführt:

- Es ist wichtig, Kosten und Nutzen von Software-Qualität und deren Vermessung explizit darzulegen. Hierzu gehört die Darstellung der Bedeutung von Software-Qualität im Kontext und in Bezug auf IT- und Geschäftsziele einer Organisation.
- Metriken müssen aus Zielen abgeleitet werden und für den Kontext passen.
- Die Einsicht zur Berücksichtigung von Software-Qualität lässt sich mit Konsequenzen mangelnder Qualität in Bezug auf Mehrkosten, Risiken, oder Haftung im juristischen Sinne motivieren.
- Existierende Messprogramme sollten soweit möglich genutzt werden, um Messziele zu beantworten.
- Messergebnisse sollten in regelmäßigen Feedback-Sitzungen an die Bereitsteller von Daten kommuniziert werden. Analyseresultate sollten in Aktionen umgesetzt werden.
- Auftragnehmer sollten gegenüber ihren Auftraggebern darauf bestehen, dass auch nichtfunktionale Anforderungen frühzeitig und präzise formuliert werden.
- Wichtig für die Akzeptanz von Qualitätsmessungen in der Software- und Systementwicklung ist, dass diese als ein Mittel zum Zweck gesehen wird und nicht als Selbstzweck. Oftmals ist es hilfreich, Software-Messungen im Rahmen von Prozessverbesserungsprogrammen einzuführen.
- Qualität ist in der Regel nicht das einzige Kriterium für Entwicklungsvorhaben. Qualität steht oftmals im Konflikt mit Kosten- und Terminzielen und ein entsprechender Tradeoff ist erforderlich. Für die Software-Qualität bedeutet dies, dass ein „angemessenes“ Qualitätslevel innerhalb einer zu definierenden Toleranzschwelle erzielt werden muss.
- Software-Messungen sollten nicht mit zu ambitionierten Zielen starten (z.B. Vorhersage). Der Aufbau von Messkompetenz in Unternehmen ist in der Regel ein mehrjähriger Prozess. Als Start wird die Ermittlung von Baselines empfohlen.

4.3 Forschungsfragen

Aus diesen Herausforderungen und Empfehlungen ergeben sich eine Reihe von Forschungsfragen, die bezüglich der Messung von Software-Qualität relevant sind. Hierzu gehört die Definition und Erprobung von Einführungsprozessen für Software-Messprogramme, die Entwicklung einfach zu bedienender und leicht anpassbarer Messwerkzeuge, die Verbindung von Metriken auf der Engineering-Ebene einer Organisation und höheren, geschäftsrelevanten Metriken (beispielsweise über eine Balanced Scorecard definiert).

Für die praktische Umsetzung stellt sich für Unternehmen die Frage, wie sie zu den für ihre Ziele und Randbedingungen wichtigen Metriken kommen. Dies kann als zentrale Herausforderung bei der Normierung von Qualitätsmodellen angesehen werden: Wie lassen sich abstrakte, empirisch belastbare Industriestandards für Qualitätsmodelle entwickeln, die zum einen hinreichend abstrakt sind und zum anderen systematische Anleitungen zum Anpassen an Firmenkontexte ermöglichen.

5 Forschungsagenda

Interessanterweise ist in all diesen Diskussionen ein Punkt unstrittig geblieben: Ein Modell für die Produktqualität von Software ist zentral für den Umgang mit Qualität in der Software-Entwicklung. Das Thema ist zu diffus, um ohne klare Definitionen und Festlegungen auskommen zu können. Aus diesem Grund ist eine zentrale Aufgabe für die Forschung in den nächsten Jahren basierend auf den bestehenden Standards und neueren Ergebnissen ein Qualitätsmodell zu entwickeln, das den aktuellen Herausforderungen gerecht wird. Es müssen Fragen beantwortet werden, wie ein Qualitätsmodell überschneidungsfrei verschiedene Attribute darstellen kann oder wie mit den weitreichenden Informationsmengen umgegangen werden kann, die in einem in der Entwicklung einsetzbaren Modell erfasst werden müssen.

Neben solch eher technischen Fragestellungen, kommt ein nicht zu unterschätzender Teil an sozio-psychologischen Themen hinzu. Ein Qualitätsmodell muss gut im Unternehmen verankert sein, damit es seine voll Wirkung entfalten kann. Hierzu müssen Erfahrungen aus der allgemeinen Prozesseinführung genutzt werden, damit hier eine nachhaltige Verbesserung besteht. Dazu gehört auch die Einführung eines entsprechenden Messprozesses. Auch wenn es sich wahrscheinlich nicht lohnt alle Bereiche eines Qualitätsmodells mit genauen Messungen abzudecken, müssen die wichtigen Teile identifiziert und mit einem entsprechenden Messprogramm unterfüttert werden. Es müssen also Verfahren entwickelt werden, die einerseits Qualitätsmodelle mit Geschäftszielen verknüpft und andererseits systematisch aus Qualitätsmodellen ableiten lässt.

Zu den obigen Herausforderungen gehören noch zwei weitere Gesichtspunkte, die dort aufgetaucht sind. Für die Konzeption und Einführung von Qualitätsmodellen genauso wie für das Messen im Rahmen von Qualitätsmodellen ist es wichtig auch die Kosten/Nutzen-Zusammenhänge mit zu betrachten. Dies hilft einerseits direkt die richtigen Qualitätsaspekte zu analysieren, aber andererseits auch Argumentationshilfen für die Einführung zu liefern. Weiterhin ist eine umfassende und integrierte Werkzeugunterstützung unbedingt notwendig. Es müssen sowohl bei der Erstellung von Modellen, als auch beim Messen von Qualität mit großen Mengen an Daten und Informationen umgegangen werden. Außerdem ist es natürlich wünschenswert, die offensichtlichen Verbindungen zwischen Qualitätsanforderungen, Qualitätsmodellen und Qualitätsmessungen auch in den Werkzeugen abzubilden.

Ein weiterer großer Punkt ist die Anpassung von Qualitätsmodellen auf unternehmens- und projektspezifische Bedürfnisse. Hierzu muss die Variabilität in existierenden und zukünftigen Modellen analysiert und daraus Schlüsse auf die Einflussfaktoren gezogen werden. Insbesondere wäre es wünschenswert im Hinblick auf Wiederverwendung, aber auch Normierung, eine gemeinsame Basis in den Qualitätsmodell-Varianten zu identifizieren.

6 Zusammenfassung

Software-Qualität ist ein entscheidender Faktor für die Konkurrenzfähigkeit von Unternehmen und Wirtschaftsstandorten. Leider konnte sich der gute Ruf von *Made in Germany* bisher nur bedingt auf Softwareprodukte übertragen lassen. Dies ist ein Problem, dem mit verbesserten Methoden zur Software-Qualität entgegen werden muss.

Qualitätsmodelle sind offensichtlich ein akzeptiertes Mittel zum Umgang mit Softwarequalität. Jedoch gibt es sehr unterschiedliche Arten der Verwendung, wie auch sehr unterschiedliche Ausprägungen in den verschiedenen Domänen und Unternehmen. Für alle diese Ausprägungen stellen sich aber eine Reihe von Forschungsfragen, die beantwortet werden müssen, um auch Softwarequalität in der Praxis handhabbar zu machen.

Dieses Papier liefert hierzu eine Forschungsagenda mit den wichtigsten Punkten, die zukünftig adressiert werden müssen. Die Agenda basiert auf einer ausführlichen Diskussionen mit akademischen und industriellen Experten und spiegelt damit ein umfangreiches Erfahrungsspektrum wider. Ein Eingehen auf die gemachten Vorschläge kann die Chance eröffnen, auch für Software die deutschen Ingenieursleistungen in der Welt beliebt zu machen.

Danksagung

Wir danken allen Teilnehmern des Workshops SQMB '08, insbesondere denjenigen, die in den Diskussionsgruppen teilgenommen haben und durch ihre wertvollen Beiträge zur Entwicklung dieser Forschungsagenda beigetragen haben.

Literatur

1. Broy, M., Jarke, M., Nagl, M., Rombach, D.: Manifest: Strategische Bedeutung des Software Engineering in Deutschland. *Informatik-Spektrum* **29**(3) (2006)
2. : Deutschland schlampt bei Softwarequalität. *Computer Zeitung* **35**(21) (2005)
3. Garvin, D.A.: What does product quality really mean? *MIT Sloan Manage. Rev.* **26**(1) (1984) 25–43
4. ISO 9126-1: Software engineering – Product quality – Part 1: Quality model (2001)
5. Deissenboeck, F., Wagner, S., Pizka, M., Teuchert, S., Girard, J.F.: An activity-based quality model for maintainability. In: *Proc. 23rd International Conference on Software Maintenance (ICSM '07)*, IEEE Computer Society Press (2007)
6. Boehm, B.W., Brown, J.R., Kaspar, H., Lipow, M., Macleod, G.J., Merrit, M.J.: *Characteristics of Software Quality*. North-Holland (1978)
7. Heitlager, I., Kuipers, T., Visser, J.: A practical model for measuring maintainability. In: *Proc. Sixth International Conference on the Quality of Information and Communications Technology*, IEEE Computer Society Press (2007) 30–39
8. Dromey, R.G.: A model for software product quality. *IEEE Transactions on Software Engineering* **21**(2) (1995) 146–162
9. Seffah, A., Donyaee, M., Kline, R.B., Padda, H.K.: Usability measurement and metrics: A consolidated model. *Software Quality Control* **14**(2) (2006) 159–178

10. ISO 9241-11: Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability (1998)
11. ISO 14598: Information technology – software product evaluation (2000)
12. Plösch, R., Gruber, H., Hentschel, A., Körner, C., Pomberger, G., Schiffer, S., Saft, M., Storck, S.: The EMISQ method – expert based evaluation of internal software quality. In: Proc. 31st Annual IEEE/NASA Software Engineering Workshop (SEW-31). (2007) 99–108
13. McCall, J., Walters, G.: Factors in Software Quality. The National Technical Information Service, Springfield, VA, USA (1977)
14. Basili, V.R.: Software modeling and measurement: the Goal/Question/Metric paradigm. Technical Report UMIACS TR-92-96, University of Maryland at College Park (1992)
15. Kitchenham, B., Linkman, S., Pasquini, A., Nanni, V.: The SQUID approach to defining a quality model. *Software Quality Control* **6**(3) (1997) 211–233
16. Wagner, S., Deissenboeck, F., Winter, S.: Erfassung, Strukturierung und Überprüfung von Qualitätsanforderungen durch aktivitätsbasierte Qualitätsmodelle. In: Workshop Erhebung, Spezifikation und Analyse nichtfunktionaler Anforderungen in der Systementwicklung. (2008)
17. Wagner, S., Deissenboeck, F., Winter, S.: Managing quality requirements using activity-based quality models. In: Proc. 5th Workshop on Software Quality (6-WoSQ), ACM Press (2008) Zur Publikation angenommen.