

Softwarequalitätsmodelle

Praxisempfehlungen und Forschungsagenda

Stefan Wagner, Manfred Broy, Florian Deißeböck, Michael Kläs, Peter Liggesmeyer, Jürgen Münch, Jonathan Streit

Zusammenfassung. Softwarequalität birgt für einen Technologiestandort wie Deutschland ein großes Potenzial. Sie ist aber gleichzeitig auch eine enorme Herausforderung. Qualitätsmodelle tragen zur Konsolidierung und Konkretisierung der vielschichtigen Qualitätsfragestellungen bei. Dieses Papier stellt als Ergebnis der Diskussionen von Qualitätsexperten aus Forschung und Praxis dar, welche Fragestellungen in der Zukunft am dringlichsten im Bereich der Softwarequalitätsmodelle bearbeitet werden müssen, um den Stand der Forschung und auch die Praxis entscheidend zu verbessern.

Abstract. Software quality holds a high potential for a technology location such as Germany. At the same time it is also a huge challenge. Quality models contribute to the consolidation and concretion of the multi-faceted questions in quality. This paper presents the results of discussions of quality experts from research and practice. To decisively improve the state of research and practice, the formulated questions need to be addressed in the future most urgently.

Softwarequalitätsmodelle sind die Grundlage für jegliche qualitätsrelevanten Aktivitäten. Die Verwendung solcher Modelle birgt ungenutzte Potenziale, aber auch Probleme, denen sich Praxis und Forschung stellen müssen. Hohe Qualität von Ingenieurprodukten ist ein traditionelles deutsches Markenzeichen. Bisher ist diese Assoziation stark auf maschinenbaulastige Produkte beschränkt. Nicht nur in Deutschland, sondern weltweit haftet Software der Ruf unvermeidlicher Qualitätsprobleme an. Dem gegenüber steht die hohe wirtschaftliche Bedeutung der Qualität von Software [3]. Beispielsweise wird in [5] dargelegt, dass der Erfolg der deutschen Softwareindustrie von ihren Fähigkeiten im Umgang mit Qualität abhängt. Softwarefehler verursachen hohe Kosten durch Rückrufaktionen und fehlende Wartbarkeit richtet enorme volkswirtschaftliche Schäden an. Um im globalen Wettbewerb bestehen zu können, wird eine Differenzierung über Qualität immer wichtiger – es besteht also ein dringender Handlungsbedarf.

Die erste Frage, die sich dabei stellt, ist: Was ist Qualität? Produktqualität kann aus unterschiedlichen Blickwinkeln betrachtet werden, wie Garvin in seinem bekannten Papier [7] eingängig darlegt. Beispielsweise macht es einen großen Unterschied, ob ein transzendenter („I know it when I see it“) oder ein produktbasierter (Messung von Produkteigenschaften) Blickwinkel eingenommen wird. Produkt-Qualitätsmodelle sind ein Mittel, um genauer zu definieren, was Softwarequalität in einem bestimmten Kontext bedeutet.

Diverse Standards für Qualitätsmodelle, wie die ISO-Norm 9126 [11], firmenspezifische Standards, aber auch neuere akademische Ansätze [4, 9, 16], existieren bereits, lassen aber Fragen offen. Beispielsweise ist eine Aggregation von einzelnen Qualitätsmessungen zu höherwertigen Aussagen immer noch nicht eindeutig geklärt. Ein weiteres Thema ist die geeignete Modellierung der Zusammenhänge von Qualität zu Kosten und Nutzen.

Dieses Papier hat das Ziel, die dringendsten Fragen im Umfeld von Softwarequalitätsmodellen zusammenzutragen. Aus diesen Fragen leiten wir eine Forschungsagenda für Softwarequalitätsmodelle ab, die der Community aus Forschung und Praxis eine gemeinsame Zielrichtung geben kann. Wir haben hierzu im Workshop „Software-Qualitätsmodellierung und -bewertung“ im Rahmen der SE 2008 in ausführlichen Diskussionen mehrere große Herausforderungen identifiziert. Erstens ist das Messen und Bewerten von Qualität, selbst mithilfe von Qualitätsmodellen, oft noch zu unsystematisch oder zeigt nur Defizite auf, aber keine Möglichkeiten und Stellschrauben, um diese zu beheben. Zweitens wird der Zusammenhang von Qualität und Kosten/Nutzen nur unzureichend behandelt. Drittens stellt die Anpassung von Qualitätsmodellen ein noch weitgehend unbearbeitetes, aber praktisch relevantes Thema dar, da Software in verschiedensten Domänen und Kontexten eingesetzt wird. Viertens ist es wichtig, eine klare Strategie bei der Einführung von Qualitätsmodellen in Unternehmen zu haben, da hiervon der Erfolg des Modells

abhängt. Trotz dieser Kritikpunkte zeigen die Erfahrungen aber auch, dass Qualitätsmodelle wertvolle Informationen liefern.

In den folgenden Abschnitten stellen wir die Hauptthemen im Detail dar und geben jeweils Praxisempfehlungen und relevante Forschungsfragen an.

Qualitätsmessungen mit Qualitätsmodellen

Qualitätsmodelle stellen eine systematische Abstraktion von Qualitätsmerkmalen für Software dar. Sie verallgemeinern von individuellen Qualitätsdefekten und bieten eine Grundlage für Messungen. Für viele sind die Definition und Verfeinerung von „-ilities“, wie Reliability oder Functionality der ISO 9126, ausreichend für ein Qualitätsmodell. Andere Ansätze gehen wesentlich weiter und beschreiben auf sehr detaillierter Ebene Wirkzusammenhänge [4] oder den Zusammenhang mit automatisierten Qualitätsüberprüfungen [16].

Beim Einsatz von Qualitätsmodellen in der Softwareentwicklung gibt es verschiedene, aufeinander aufbauende Nutzungsszenarien [14]. Beispielsweise kann ein Modell zuerst verwendet werden, um Qualität zu verstehen. Dafür sollten die komplexen und vielschichtigen Aspekte von Softwarequalität in einem Qualitätsmodell so beschrieben sein, dass das Verständnis verbessert wird. Nur ein strukturiertes Vorgehen, wie es ein Qualitätsmodell bietet, erlaubt überhaupt eine sinnvolle Definition von Qualität. Diese Definition kann wiederum als Ausgangspunkt für die Qualitätsanalyse verwendet werden. Das Modell kann hierfür Indikatoren und Metriken definieren und bei der Interpretation der Messergebnisse unterstützen. Weiterhin sollte das Modell auch Wege zur Verbesserung der Qualität aufzeigen, indem es hilft, die maßgeblichen Einflussfaktoren zu identifizieren und im Idealfall Qualität frühzeitig vorherzusagen.

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

Erfahrungen und Herausforderungen

Qualitätsplanung und -sicherung im Software- und Systems-Engineering erfordert die Quantifizierung von Soll und Ist. Dafür bedarf es einer Operationalisierung der Qualitätsaspekte in Form konkreter Metriken. Die Auswahl geeigneter Metriken ist dabei von einer Menge von Faktoren (wie z.B. dem Entwicklungskontext) abhängig, ohne deren Beachtung ein Qualitätsmodell einen Großteil seines Nutzens einbüßt. Da derzeit existierende Standards wie [10] aufgrund der angestrebten Breite der Anwendung nur ein hohes Abstraktionsniveau umfassen, können sie diese Operationalisierung nicht bieten und ihr praktischer Nutzen bleibt äußerst fragwürdig. Daher wäre ein einheitliches Meta-Modell wünschenswert, das die Beschreibung sowohl notwendiger als auch optionaler Bestandteile eines Qualitätsmodells erlaubt. Auf einer solchen Meta-Ebene bewegen sich die Methoden Factors-Criteria-Metrics (FCM) [15] und Goal/Question/Metric (GQM) [1]. FCM beschreibt dabei ein einfaches Meta-Modell für Qualitätsmodelle, GQM ein Vorgehen zur Entwicklung angepasster Messsysteme.

Weiterhin gibt es für viele Qualitätsaspekte nur ungenaue, implizit definierte Begriffe (z.B. Wartbarkeit, Performanz, Genauigkeit). Selbst vergleichsweise klar erscheinende Begriffe wie Größe, Kosten oder Aufwand können in unterschiedlichen Umgebungen sehr unterschiedliche Definitionen haben. Insgesamt stellen sich die zielorientierte Ableitung geeigneter Metriken und die Minimierung der zu erfassenden Metrikmenge als schwierig dar, insbesondere bei „weichen“ Einflussfaktoren, wie Erfahrung, Teamkohärenz oder der Güte von Codekommentaren.

Der Mangel an passenden Standards und Anleitungen führt dazu, dass Unternehmen Schwierigkeiten haben, die für sie wichtigen Indikatoren zu ermitteln. Oftmals ist nicht klar, wie bestehende Messprogramme genutzt werden können, um neue Messziele zu beantworten. Darüber hinaus fehlen in der Praxis meist Baselines, sodass eine vergleichende Beurteilung von Softwarequalität nicht möglich ist.

Eine weitere Erfahrung in Bezug auf Softwaremessungen 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. In der Regel berücksichtigen diese Standards jedoch ungenügend den Einsatzkontext. Die Präferenz für die Anwendung starrer Metriksätze dürfte auch daher rühren, dass das Bewusstsein, dass im Softwarebereich eine wesentlich höhere Anzahl von Variationsparametern als bei

Produktions- und Fertigungsprozessen existiert und somit Metriken und Qualitätsmodelle wesentlich größeren Variationen unterliegen, oftmals nicht vorhanden ist.

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

Empfehlungen

Aus den Herausforderungen ergeben sich drei Kernempfehlungen zu Qualitätsmessungen in der Praxis:

- Metriken müssen aus Zielen abgeleitet werden und für den Kontext passen.
- Existierende Messprogramme sollten soweit möglich genutzt werden, um Messziele zu beantworten. Oftmals ist es hilfreich, Softwaremessungen im Rahmen von anderen Prozessverbesserungsprogrammen einzuführen, um Synergieeffekte zu nutzen.
- Softwaremessungen 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.

Forschungsfragen

Aus diesen Herausforderungen und Empfehlungen ergeben sich Forschungsfragen im Hinblick auf die Messung von Softwarequalität. In der praktischen Umsetzung stellt sich für Unternehmen die Frage, wie sie zu den für ihre Ziele und Randbedingungen wichtigen Metriken kommen. Die Ableitung und Definition solcher Metriken ist weiterhin eine Frage für die Forschung. Die Aufgabe, empirisch belastbare Industriestandards für Qualitätsmodelle zu entwickeln, die zum einen hinreichend abstrakt sind und zum anderen systematische Anleitungen zur Anpassung an Firmenkontexte ermöglichen, kann als zentrale Herausforderung bei Qualitätsmodellen angesehen werden. Weiterhin ist die Entwicklung einfacher zu bedienender Messwerkzeuge, die sich leicht an diese Kontexte anpassen lassen, zu nennen.

Kosten und Nutzen von Qualität

Die Zusammenhänge von Kosten und Qualität, seit vielen Jahren Thema im produzierenden Gewerbe, haben auch Eingang in das Software Engineering gefunden. In einem gewinnorientierten Unternehmen kann Qualität nie losgelöst von anderen Faktoren gesehen werden, sondern muss einer Kosten/Nutzen-Betrachtung unterzogen werden.

Erfahrungen und Herausforderungen

Messungen und Aktivitäten zur Softwarequalität sind bis heute in vielen Unternehmen und Organisationen nicht selbstverständlich. Vielfach wird der Aspekt der Softwarequalität ignoriert oder findet erst Beachtung, wenn bereits erhebliche Probleme aufgetreten sind. Betrachtungen von Qualitätsaspekten finden häufig erst sehr spät im Entwicklungsprozess statt, in der Regel im Zusammenhang mit ersten Testprozessen. Die meisten Qualitätsmängel lassen sich zu diesem Zeitpunkt nicht mehr bzw. nur noch mit hohem Aufwand beheben.

Eine Ursache für das fehlende Bewusstsein für Softwarequalität in Entwicklungsorganisationen ist, dass die wirtschaftliche Bedeutung von Softwarequalität Entscheidungsträgern oftmals unklar ist. Der Bezug von Maßnahmen zur Qualitätssteigerung zu höheren Zielen einer Organisation, insbesondere zu Geschäftszielen, ist vielfach nicht explizit dargestellt. Die Kosten von Maßnahmen und Messungen sind klar sichtbar, während der Nutzen in Form von zukünftigen Einsparungen und vermiedenen Folgekosten nicht offensichtlich und schwer quantifizierbar ist [8]. Dies hängt damit zusammen, dass Software in vielen Unternehmen als umsetzender und nicht als wertschöpfender Faktor in Bezug auf die Unternehmensziele gesehen wird, obwohl sie in vielen Branchen mittlerweile der bedeutendste Innovationstreiber ist. Hinzu kommt der Gegensatz zwischen der vermeintlich langfristigen Amortisierung von Qualitätsmaßnahmen und kurzfristigen Planungshorizonten auf Managementebene. Erfahrungen aus der industriellen Praxis zeigen, dass oftmals hohe ROI-Erwartungen in Bezug auf Maßnahmen zur Verbesserung der

Softwarequalität existieren.

Häufig ist Softwarequalität auch bei der Vergabe von Aufträgen nicht relevant. 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 schwerer 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).

Empfehlungen

Hierzu können folgende Empfehlungen zur erfolgreichen Einführung von Kosten/Nutzen-Betrachtungen von Qualität in der Praxis gegeben werden:

- Es ist wichtig, Kosten und Nutzen von Softwarequalität und deren Vermessung explizit darzulegen. Hierzu gehört die Darstellung der Bedeutung von Softwarequalität im Kontext und in Bezug auf IT- und Geschäftsziele einer Organisation. Teilweise können Konsequenzen mangelnder Qualität in Form von Mehrkosten, Risiken und Haftung im juristischen Sinne dargestellt werden.
- Qualität steht mitunter im Konflikt mit (kurzfristigen) Kosten- und Terminzielen oder wird zumindest so wahrgenommen. Hier ist ein entsprechender Tradeoff einzubeziehen, sodass ein „angemessenes“ Qualitätslevel innerhalb einer zu definierenden Toleranzschwelle erzielt wird. Der mittel- und langfristige positive Einfluss von Qualität auf Kosten und Termine muss explizit dargestellt werden.
- Auftragnehmer sollten gegenüber ihren Auftraggebern darauf bestehen, dass auch nichtfunktionale Anforderungen frühzeitig und präzise formuliert werden.

Forschungsfragen

Eine systematische Aufarbeitung der Kosten-Nutzen-Zusammenhänge im Bereich der Softwarequalität würde den Einsatz von Qualitätsmodellen deutlich erleichtern. Durch die Bereitstellung von Kosten-Nutzen-Modellen, die quantitative Aussagen bzgl. des langfristigen Nutzens von Qualitätsaktivitäten treffen können, könnten viele Aussagen, die heute eher anekdotischer Natur sind, auf ein sicheres Fundament gestellt werden und damit zur sachlichen Diskussion beitragen. Metriken auf der Engineering-Ebene einer Organisation müssen mit höheren, geschäftsrelevanten Metriken (wie beispielsweise über eine Balanced Scorecard definiert) in Verbindung gebracht werden können.

Anpassung von Qualitätsmodellen

Software ist in den verschiedensten Bereichen und Domänen im Einsatz, wodurch sehr unterschiedliche Anforderungen an die Systeme gestellt werden. Dies trifft auch auf Qualitätsanforderungen zu. Die normierten Modelle, wie die ISO 9126, bieten heute keine konkreten Vorgaben, wie sie an diese Unterschiede anzupassen sind. Dies ist ein starkes Hemmnis bei ihrer operationalisierten Verwendung in Softwareprojekten. Wir haben für dieses Thema bewusst den Begriff „Anpassung“ gewählt, da die Alternative „Tailoring“ oft nur für das Weglassen von Teilen, aber nicht für die Erweiterung steht.

Erfahrungen und Herausforderungen

Es existiert bereits eine Reihe von Qualitätsmodellen aus unterschiedlichsten Quellen. Modelle finden sich in der wissenschaftlichen Literatur [2, 4, 6, 9, 17] und in diversen allgemeinen und domänenspezifischen Standards wie [11, 12]. Darüber hinaus definieren Unternehmen, die sich weitergehend mit Softwareentwicklung beschäftigen, typischerweise firmeninternen Richtlinien, die auf die konkreten Anforderungen in den jeweiligen Projekten angepasst sind. Schließlich kann auch der Einsatz von Messwerkzeugen ein entsprechendes Qualitätsmodell implizit vorgeben.

Grundsätzlich bewegt man sich bei Qualitätsmodellen für die Softwareentwicklung zwischen der kompletten Eigendefinition des Modells und der unveränderten Nutzung eines bestehenden Modells. Ersteres erlaubt das genaue Zuschneiden auf die konkreten Bedürfnisse, erfordert aber auch einen enormen

Aufwand und entsprechende Expertise. Letzteres hingegen minimiert die Aufwände, kann aber erhebliche Probleme verursachen, wenn das eingesetzte Modell nicht auf die tatsächlichen Gegebenheiten im Unternehmen oder Projekt passt. Weiterhin wurde in verschiedenen Projekten die Erfahrung gemacht, dass Qualitätsmodelle in der konkreten Anwendung sehr umfangreich werden können, da sie detaillierte Informationen zu verschiedensten Qualitätsaspekten abdecken müssen.

Aus diesen Gründen erachten wir ein Qualitätsmodell für sinnvoll, das ein normiertes Grundmodell definiert, aber auch einen Prozess zur Anpassung an eigene Anforderungen und Gegebenheiten umfasst. Zur Entwicklung eines solchen Modells müssen jedoch noch einige grundlegende Fragen geklärt werden. Einzig der SQUID-Ansatz [13] umfasst derzeit ein Vorgehen zur spezifischen Definition von Qualitätsmodellen, bietet aber weniger Details zu deren Operationalisierung.

Empfehlungen

Ein wichtiger Punkt in der Anpassung von Qualitätsmodellen ist die Identifizierung der wichtigsten Anpassungsdimensionen bzw. der Einflussfaktoren. Hierfür sind die Ziele des Unternehmens oder Projekts entscheidend. Es muss untersucht werden, wie sich diese im Qualitätsmodell auswirken und das Modell muss entsprechend angepasst werden. Anschließend muss eine Validierung stattfinden, bei der überprüft wird, ob die Anpassungen sinnvoll sind. Dies kann beispielsweise durch Rückkopplung mit 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. Es ist unwahrscheinlich, dass zu Beginn die Qualitätsanforderungen und das Qualitätsmodell vollständig verfügbar sind. Deshalb scheint ein Bootstrapping-Verfahren am sinnvollsten: Anforderungen und Qualitätsmodell werden im Projektverlauf iterativ und inkrementell abgeglichen [18, 19]. 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 kann dieses natürlich wiederum die Gewichtungen bestimmen.

Forschungsfragen

Aus diesen Herausforderungen und Empfehlungen leiten wir eine Reihe von Forschungsfragen ab, deren Beantwortung wir als notwendig für eine praktische Anpassung von Qualitätsmodellen sehen. Zuerst muss untersucht werden, welches Spektrum an Qualitätsmodellen zurzeit existiert, und wie groß die Variabilität in Qualitätsmodellen über Domänen, Projekte und Technologien hinweg ist. Die dargestellten Erfahrungen deuten auf eine signifikante Variabilität hin, die aber noch nicht wissenschaftlich untersucht wurde. Dies ist jedoch wichtig, um einen brauchbaren, ausreichend flexiblen Anpassungsmechanismus abzuleiten.

Weiterhin muss analysiert werden, welche Komponenten ein Qualitätsmodell notwendigerweise und optional umfasst, sowohl bezüglich der Struktur eines Qualitätsmodells als auch bezüglich der Inhalte. Bei den Kostenaspekten stellt sich die Frage, welche Beziehung zwischen bestehenden Kostenrechnungs- und Qualitätsmodellen besteht, da dies ein wichtiger Faktor in der Anpassung sein könnte. Auch ist noch unklar, wie ein umfassendes und umfangreiches Qualitätsmodell (nach seiner Anpassung) validiert werden kann.

Einführung von Qualitätsmodellen

Grundvoraussetzung für ein erfolgreiches Qualitätsmanagement ist die Schaffung der notwendigen Akzeptanz im entsprechenden Unternehmen. Abstrakt betrachtet hat die Einführung von Qualitätsmodellen das Ziel, „Qualitätseigenschaften die Freiwilligkeit zu nehmen“. Ähnlich wie bei funktionalen Anforderungen soll erreicht werden, dass Qualitätsanforderungen ganz selbstverständlich umgesetzt und nicht nach Belieben weggelassen oder verändert werden. Diese Einführung ist ein aufwändiger Prozess, der sorgfältig geplant und

im Vorhinein auf den zu erwartenden Nutzen hin analysiert werden muss. Welche Faktoren hierfür erfolgsentscheidend sind, muss genauer untersucht werden.

Erfahrungen und Herausforderungen

Der Einsatz von Qualitätsmodellen erfolgt meist im Rahmen langfristig angelegter Qualitätsinitiativen und nur selten zur Behebung akuter Probleme. Aufgrund dieser Langfristigkeit ist es oft schwierig, die beteiligten Personen zu motivieren, da diese den unmittelbaren Nutzen der Initiativen nicht einfach erkennen können. Im Gegenteil, Qualitätsinitiativen erfordern von den meisten Beteiligten, zumindest zu Beginn, eine gewisse Mehrarbeit. Ungünstige Randbedingungen, wie Lieferdruck, fehlende Ressourcen und fehlende Unterstützung des Managements erschweren die Einführung.

Ein weiteres Problem bei der Einführung von Qualitätsmodellen kann sich durch organisatorische Trennungen und den daraus resultierenden Perspektiven der Mitarbeiter unterschiedlicher Prozessphasen ergeben. So ist es schwierig, Entwickler, die nie mit Wartungsaufgaben betraut werden, von den Vorteilen von lesbarem Code zu überzeugen. Häufig gilt, dass nicht alle Projektbeteiligten über die entsprechende Ausbildung verfügen, die es ihnen erlauben würde, den langfristigen

Nutzen von Qualitätsinitiativen bzw. die Folgen einer Nicht-Einhaltung von Qualitätsstandards zu erkennen.

Neben diesen motivatorisch-personellen Problemen stellt die bereits 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. Speziell automatische Überprüfungsmethoden für Qualitätskriterien leiden oft unter einer hohen Zahl von False Positives, die schnell zu einer Demotivation der Projektbeteiligten führen können.

Empfehlungen

Es empfiehlt sich, die Einführung von Qualitätsmodellen möglichst behutsam vorzunehmen, um die Beteiligten nicht durch eine Fülle von Neuerungen zu überfordern. Als äußerst hilfreich hat es sich erwiesen, einen unmittelbaren Nutzen für die einzelnen Beteiligten zu identifizieren und darzustellen. Dies trägt dazu bei, die Akzeptanz für langfristige Neuerungen zu erhöhen. Jenseits dieses unmittelbaren Nutzens muss natürlich dafür gesorgt werden, dass alle Beteiligten ein einheitliches Verständnis für die Ziele, die mithilfe von Qualitätsmodellen erreicht werden sollen, haben und sich insbesondere über die langfristigen Konsequenzen einer Nicht-Berücksichtigung von Qualitätseigenschaften bewusst sind. Hierbei kann ein Qualitätsmodell, das Kosten/Nutzen-Aspekte berücksichtigt, wertvolle Dienste leisten.

Wichtig für die Akzeptanz von Qualitätsmessungen in der Software- und Systementwicklung ist, dass diese als ein Mittel zum Zweck und nicht als Selbstzweck gesehen wird. Messergebnisse sollten in regelmäßigem Feedback an die Beteiligten kommuniziert werden, und Erkenntnisse aus der Analyse sollten in Aktionen resultieren.

Der Inflexibilität von Qualitätsmodellen und den daraus resultierenden Problemen bei der Einführung kann mit den im letzten Abschnitt diskutierten Anpassungsmechanismen begegnet werden. Folgende Dimensionen der Anpassung sind hierbei von Bedeutung:

- Organisationsspezifische Anpassung. Eine Anpassung des Qualitätsmodells an die Organisation ist wichtig, um den Projektbeteiligten zu demonstrieren, dass es sich beim Einsatz eines Qualitätsmodells nicht um eine bürokratische Erfüllung von Normen, sondern um ein Mittel zur Steigerung der Effektivität und Effizienz in der Entwicklung handelt.
- Mitarbeiterspezifische Perspektiven. Neben der tatsächlichen Anpassung sollte das Qualitätsmodell für die Projektbeteiligten zugeschnittene Perspektiven bieten. Dadurch wird jedem Projektbeteiligten nur der für ihn relevante Anteil zur Verfügung gestellt, und eine Belastung mit unnötiger Information wird vermieden.
- Konfiguration der Überprüfungsmethoden. Es ist notwendig, Überprüfungswerkzeuge und -methoden so auf den jeweiligen Kontext anzupassen, dass die Anzahl der False Positives auf ein erträgliches Maß (< 5%) beschränkt wird.

Forschungsfragen

Aus diesen Erkenntnissen und Empfehlungen ergibt sich eine Reihe von Forschungsfragen bezüglich der Einführung von Qualitätsmodellen. Neben den eher technischen Fragen über verbesserte Anpassbarkeit und bessere Werkzeugunterstützung von Qualitätsmodellen spielen hierbei vor allem sozio-psychologische Aspekte (Change Management) eine Rolle. Hierbei wäre insbesondere eine strukturierte Aufarbeitung der

Erkenntnisse aus anderen Disziplinen, z.B. Prozesseinführung, ein vielversprechender Startpunkt, um besser zu verstehen, welchen Problemen die Einführung von Qualitätsmodellen auf einer sozio-psychologischen Ebene begegnet.

Jenseits dieser konkreten Fragestellungen ist es sinnvoll, das Thema Qualität 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.

Zusammenfassung

Dabei 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 Softwareentwicklung. 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.

Softwarequalität ist ein entscheidender Faktor für die Konkurrenzfähigkeit von Unternehmen und Wirtschaftsstandorten. Qualitätsmodelle sind ein akzeptiertes Mittel zum Umgang mit Softwarequalität. Jedoch gibt es sehr unterschiedliche Arten der Verwendung und auch sehr unterschiedliche Ausprägungen in den verschiedenen Domänen und Unternehmen. Für alle diese Ausprägungen stellt sich aber eine Reihe von Forschungsfragen, die beantwortet werden müssen, um Softwarequalität in der Praxis beherrschbar zu machen.

Dieses Papier liefert hierzu Praxisempfehlungen und eine Forschungsagenda mit den zentralen Punkten, basierend auf einer ausführlichen Diskussion mit akademischen und industriellen Experten. Es spiegelt damit ein umfangreiches Erfahrungsspektrum wider. Die Vorschläge bieten die Chance, die deutschen Ingenieursleistungen auch für Software in der Welt beliebt zu machen.

Danksagung

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

Literatur

1. Basili VR(1992)Software modeling and measurement: the Goal/Question/Metric paradigm. Technical Report UMIACS TR-92-96, University of Maryland at College Park
2. Boehm BW, Brown JR, Kaspar H, Lipow M, Macleod GJ, Merritt MJ (1978) Characteristics of Software Quality. Elsevier Science, North-Holland
3. Broy M, Jarke M, Nagl M, Rombach D (2006) Manifest: Strategische Bedeutung des Software Engineering in Deutschland. Informatik-Spektrum 29(3):210–221
4. Deissenboeck F, Wagner S, Pizka M, Teuchert S, Girard JF (2007) An activity based quality model for maintainability. In: Proc. 23rd International Conference on Software Maintenance (ICSM '07), IEEE Computer Society Press, Los Alamitos, CA, S 184–193
5. Deutschland schlampft bei Softwarequalität (2005) Computer Zeitung 35(21)
6. Dromey RG (1995) A model for software product quality. IEEE Trans Softw Eng 21(2):146–162
7. Garvin DA (1984) What does product quality really mean? MIT Sloan Manage Rev 26(1):25–43
8. Heidrich, J, Münch J, Trendowicz A (2009) Messbasierte Ausrichtung von Softwarestrategien an Geschäftszielen. Fachzeitschrift für Information Management & Consulting (IM), no. 1/2009, pp. 82-89, February 2009.
9. Heitlager I, Kuipers T, Visser J (2007) A practical model for measuring maintainability. In: Proc. Sixth International Conference on the Quality of Information and Communications Technology, IEEE Computer Society Press, Los Alamitos, CA, S 30–39
10. ISO 14598 (2000) Information technology – software product evaluation
11. ISO 9126-1 (2001) Software engineering – Product quality – Part 1: Quality model

12. ISO 9241-11 (1998) Ergonomic requirements for office work with visual display terminals (VDTs) – Part 11: Guidance on usability
13. Kitchenham B, Linkman S, Pasquini A, Nanni V (1997) The SQUID approach to defining a quality model. *Softw Qual Control* 6(3):211–233
14. Kläs M, Münch J (2008) Balancing Upfront Definition and Customization of Quality Models. In: *Workshop-Band Software-Qualitätsmodellierung und -bewertung (SQMB '08)*, Technischer Bericht TUM-I081, TU München, S 26–30
15. McCall J, Walters G (1977) *Factors in Software Quality*. The National Technical Information Service, Springfield, VA
16. Plösch R, Gruber H, Hentschel A, Körner C, Pomberger G, Schiffer S, Saft M, Storck S (2007) The EMISQ method – expert based evaluation of internal software quality. In: *Proc. 31st Annual IEEE/NASA Software Engineering Workshop (SEW-31)*, S 99–108
17. Seffah A, Donyae M, Kline RB, Padda HK (2006) Usability measurement and metrics: A consolidated model. *Softw Qual Control* 14(2):159–178
18. Wagner S, Deissenboeck F, Winter S (2008) Erfassung, Strukturierung und Überprüfung von Qualitätsanforderungen durch aktivitätsbasierte Qualitätsmodelle. In: *Software Engineering 2008 – Workshopband, LNI 122*, Bonner Köllen Verlag, S 464–469
19. Wagner S, Deissenboeck F, Winter S (2008) Managing quality requirements using activity-based quality models. In: *Proc. 5th Workshop on Software Quality (6-WoSQ)*, Leipzig, 10. Mai 2008. ACM Press, New York, NY, S 29–34