

LifeCycleQM – Ziel- und lebenszyklusorientiertes Qualitätsmanagement wieder verwendbarer Software

Christian Denger und Ronny Kolb
Fraunhofer Institut für Experimentelles
Software Engineering (IESE)
Fraunhofer-Platz 1
67663 Kaiserslautern
{denger, kolb}@iese.fraunhofer.de

Torsten Schulz
IBS AG
Rathausstr. 56
56203 Höhr-Grenzhausen
Torsten.Schulz@ibs-ag.de

Kurzfassung

Wiederverwendung von Softwarekomponenten spielt insbesondere für kleine und mittelständische Unternehmen eine immer größere Rolle, um in wachsenden und komplexer werdenden Märkten Softwareprodukte hoher Qualität auf effiziente und produktive Art und Weise zu erstellen. Das Gesamtziel dieses Vorhabens ist die Entwicklung einer ziel- und lebenszyklus-orientierten Qualitätsstrategie zur Optimierung des Qualitätsmanagements wieder verwendbarer Softwarekomponenten im Kontext der Informationssystementwicklung. Die Herausforderung des Vorhabens besteht darin festzustellen, wie die Qualität wieder verwendbarer Komponenten von der Entwicklung, über die Integration, bis zu ihrer Anwendung in verschiedenen Produktvarianten bewertet werden kann. Es wird erarbeitet, wie das Wissen über die Qualität der wieder verwendbaren Komponenten genutzt werden kann, um Qualitätssicherungstechniken über den Lebenszyklus der Komponenten hinweg effizient miteinander zu verknüpfen. Zu diesem Zweck werden geeignete Modelle zur a) systematischen Kontrolle und Bewertung der Qualität der Komponenten und b) Entscheidungsunterstützung welche Qualitätssicherungstechniken wann im Lebenszyklus eingesetzt werden sollten, um wieder verwendbare Komponenten und Produkte garantierter Qualität zu erzeugen, definiert. Grundlagen der Modellbildung sind Fehlerklassifikationen und Fehlerstrommodelle die an den Kontext von Softwarewiederverwendung angepasst werden. Das entwickelte Modell und der Ansatz werden in Form eines Werkzeuges implementiert und durch die Industriepartner in künftigen Softwareentwicklungsprojekten eingesetzt.

Einleitung

Wiederverwendung von Softwarekomponenten spielt insbesondere für kleine und mittelständische Unternehmen (im weiteren KMUs genannt) eine immer größere Rolle, um in wachsenden und komplexer werdenden Märkten Softwareprodukte hoher Qualität auf effiziente und produktive Art und Weise zu erstellen. Bei der Entwicklung von Informationssystemen müssen Firmen sehr häufig kunden-spezifische Lösungen anbieten. Diese setzen sich meist aus wieder verwendeten, angepassten Softwarekomponenten und kundenspezifischen Softwarekomponenten (d.h. Softwareteile, die speziell für einen Kunden erstellt und integriert werden) zusammen. Die Wiederverwendung der Software erfolgt dabei zum Teil systema-

tisch (z.B. über komponentenbasierte Entwicklung, Variabilitäten) und zum Teil unsystematisch (z.B. kopieren und adaptieren von Code aus früheren Produktvarianten, frühen Versionen und Releases eines Produktes und dessen Integration in das neue Produkt / die neue Release). Daraus resultieren eine Vielzahl von unterschiedlichen Produkt- und Komponentenvarianten und kundenspezifische Konfigurationen, die von den Firmen betreut und deren Qualität sichergestellt werden muss. In einem solchen Zusammenhang stellt sich die Frage, wie eine garantierte Qualität der entwickelten Produkte durch ein effizientes Qualitätsmanagement und eine systematische Qualitätsstrategie für wieder verwendbare **und** kundenspezifische Komponenten erreicht werden kann.

Der Forschungsschwerpunkt im Bereich Wiederverwendung lag in der Vergangenheit auf der Entwicklung optimierter Verfahren und Techniken zum Design und zur Implementierung wieder verwendbarer Software. Systematische Verfahren zum Qualitätsmanagement und zur Qualitätssicherung wurden in diesem Kontext kaum und nur sehr eingeschränkt betrachtet. Somit fehlt es vielen Firmen, insbesondere KMUs, an Wissen darüber:

- wie die Qualität wieder verwendbarer Komponenten über ihren gesamten Lebenszyklus hinweg möglichst effizient und fokussiert sichergestellt werden kann,
- wie sich Qualitätsmanagement und Releasemanagement (die Verwaltung von bestehenden Softwarereleases und Versionen eines Produktes) kombinieren lassen und
- welche Auswirkungen Softwarefehler in (wieder verwendbaren) Softwarekomponenten auf die einzelnen Softwareentwicklungsschritte und die verschiedenen Produktvarianten haben.

Das fehlende Wissen über die oben genannten Zusammenhänge resultiert darin, dass es Firmen unmöglich ist die verfügbaren Aufwände für Qualitätssicherungsaktivitäten optimiert im Entwicklungszyklus einzusetzen. Es fehlt insbesondere an Möglichkeiten, das Wissen über die Qualität wieder verwendbarer und wieder verwendeter Softwarekomponenten und Arbeitsprodukte in konkreten Produkten zu analysieren und daraus gezielt Rückschlüsse auf Optimierungsmaßnahmen und somit eine Balancierung der Qualitätssicherungsaufwände im Entwicklungszyklus abzuleiten. Für viele KMUs resultiert das fehlende Wissen in überhöhten Kosten für die Qualitätssicherung aufgrund von redundanten Prüfungen (z.B. doppeltes Testen) und unfokussiert durchgeführten Qualitätssicherungsaktivitäten (z.B. durch Inspektionen oder Tests alle denkbaren Qualitäten des Endproduktes sicherstellen ohne Beachtung der Relevanz dieser Qualitäten).

Darüber hinaus resultieren aus der Wiederverwendung von Software verschiedenste Herausforderungen an die Qualitätssicherung, die sowohl für systematische als auch für unsystematische Wiederverwendung Gültigkeit besitzen. Zum einen resultiert aus der Tatsache das wieder verwendbare Komponenten oft generisch sind eine sehr große Anzahl von möglichen Konfigurationen die zu Testen sind. Diese Anzahl steigt darüber hinaus exponentiell bei der Integration einer Vielzahl von Komponenten. Probleme und Herausforderungen ergeben sich oft auch hinsichtlich des Regressionstests. So stellt sich die Frage ob bei einer Änderung an einer wieder verwendeten Komponenten alle Produkte die diese verwenden erneut getestet werden müssen oder ob man die Tests ohne später Qualitätsprobleme in Kauf nehmen zu müssen auf ein paar wenige Produkte beschränken kann. Ein weiteres Problem aus der Wiederverwendung von Software ergibt sich für die Qualitätssicherung aus organisatorischen Aspekten wie der größere Anzahl von Stakeholdern mit zum Teil widersprüchlichen Anforderungen sowie der Trennung und Verteilung von Entwicklungsaktivitäten.

Das Gesamtziel des Vorhabens ist daher die Entwicklung eines systematischen Qualitätsmanagementansatzes für wieder verwendbare Software. Kern dieses Ansatzes ist die Definition einer Qualitätssicherungsstrategie. Diese Strategie wird ziel- und lebenszyklusorientiert sein. Das bedeutet die Strategie definiert, aufbauend auf den Qualitätszielen und dem Kontext einer Firma oder eines speziellen Projektes, wie verschiedene qualitätssichernde Techniken im Entwicklungsprozess der Software eingesetzt werden sollten. Dazu wird ein messbasiertes Verfahren entwickelt, das es erlaubt die Qualität wieder verwendbarer und wieder verwendeter Komponenten von der Entwicklung, über die Integration, bis zu ihrer Anwendung in verschiedenen Produktvarianten zu bewerten. Weiter erlaubt das Verfahren Informationen bezüglich der Qualität der wieder verwendbaren und wieder verwendeten Komponenten so zu nutzen, dass Qualitätssicherungstechniken möglichst effizient im Entwicklungsprozess miteinander verknüpft werden können.

Problemlage

Stand der Wissenschaft

In verschiedenen Forschungsgruppen und Projekten wurde in den letzten Jahren sehr stark auf Techniken zur Entwicklung wieder verwendbarer Softwarekomponenten fokussiert. Dabei spielen vor allem Ansätze zur systematischen Wiederverwendung, wie etwa komponentenbasierte Software-Entwicklung [AB+01] und Produktlinien [BK+04, CN01] eine wesentliche Rolle. Im Kontext von Software-Wiederverwendung im Allgemeinen und in den oben genannten Bereichen im Speziellen wurden systematische Ansätze zum Qualitätsmanagement und zur Qualitätssicherung bisher kaum betrachtet. Techniken wie Testen und Inspektionen im Allgemeinen wurden zwar intensiv in Forschung und Industrie untersucht, zu deren spezialisierten Anwendung im Kontext der Wiederverwendung finden sich jedoch nur sehr wenige Ansätze [KMM04]. Die bestehenden Ansätze fokussieren dann meist auf bestimmte Phasen im Entwicklungszyklus und setzen häufig das Vorhandensein bestimmter Entwicklungsmethodiken voraus.

Im Bereich systematischer Testverfahren im Kontext von Wiederverwendung existieren erste Ansätze, die hauptsächlich im Bereich der Produktlinienentwicklung entworfen wurden. McGregor [McG01] beschreibt hierzu Ideen, wie Testprozesse in den Kontext systematischer Wiederverwendung in einer Produktlinienorganisation eingebettet werden sollten. Dazu werden insbesondere Möglichkeiten aufgezeigt, Testaktivitäten so zu gestalten, dass sie den Herausforderungen systematischer Wiederverwendung gerecht werden. Zentraler Aspekt des Ansatzes ist die Übertragung der Idee der Wiederverwendung von Code auf sämtliche Testartefakte (Testpläne, Testfälle, etc.). Dadurch werden Testaktivitäten und Testprozesse mit dem Entwicklungsprozess synchronisiert. Allerdings sind die beschriebenen Ideen nicht weitgehend genug, um die Herausforderungen von Qualitätssicherung im Kontext von Wiederverwendung hinreichend zu adressieren [KM03, MH03]. Ein Ansatz zur systematischen und effizienten Ableitung von Integrations- und Systemtestfällen für Produktfamilien (ScenTED) wird in [KP+03] beschrieben. Die Idee des Ansatzes besteht darin, generische, wieder verwendbare Testfälle aus Use Cases herzuleiten, welche mit Informationen bezüglich möglicher Variabilitäten erweitert wurden. Anstelle der Entwicklung mehrerer Testfälle wird in diesem Ansatz die Variabilität der Use Cases dazu in einem generischen Testfall abgebildet, der die verschiedenen funktionalen Ausprägungen (bedingt durch die Variabilität) berücksichtigt. Die meisten Ansätze setzen das Vorhandensein bestimmter Artefakte voraus, aus denen Testfälle abgeleitet werden können (z.B., UML Design, Use Cases, generische Komponentenbeschreibungen). Allerdings werden in vielen KMUs diese Artefakte in den Entwicklungsprozessen nicht erzeugt und die Verfahren sind dadurch schwierig anwendbar. Darüber hinaus sind die

beschriebenen Ansätze im Wesentlichen für den Einsatz in Produktlinienorganisationen bestimmt, in denen systematische Wiederverwendung stattfindet. Diese Voraussetzungen können aber insbesondere viele KMUs nicht erfüllen. Gängige Praxis ist dort auf bestehenden Produktvarianten (Releases) aufzusetzen und diese entsprechend bestimmter Anforderungen zu erweitern und anzupassen. Dadurch wird der bestehende Kern der Software wieder verwendet und entsprechende kundenspezifische Aspekte werden ergänzt. Es bedarf daher weniger restriktiver Ansätze, die auf vorhandenen Strukturen der KMUs aufbauen und insbesondere auch weniger systematische Formen der Wiederverwendung berücksichtigen.

Neben Testansätzen sind Inspektionen eine der effektivsten und effizientesten Qualitätssicherungsaktivitäten [Fag76, GG93, Wie02]. Die Idee von Inspektionen besteht darin, Zwischenartefakte, die in der Entwicklung erzeugt werden einem intensiven Review zu unterziehen, mit dem Ziel möglichst viele Fehler aus diesen Artefakten herauszunehmen. Die Folge davon ist, dass Fehler nicht in spätere Entwicklungsphasen weitergegeben werden und somit keine Folgefehler verursachen. Die Konsequenz daraus ist eine verbesserte Qualität des Endproduktes. Darüber hinaus werden Entwicklungsprozesse stabiler und Testaufwände können zum Teil reduziert werden. Fehlerfolgekosten, die durch Reworkaufwände entstehen, können durch das frühe Finden der Fehler erheblich reduziert werden [BB01]. Die Anwendung von Inspektionen und deren Nützlichkeit ist in Forschung und Industrie weitestgehend akzeptiert. Wie in [DT04] beschrieben gibt es zurzeit allerdings keine Ansätze, die beschreiben, wie Inspektionen im Kontext von Wiederverwendung eingesetzt werden sollten. Dabei stellt sich insbesondere die Frage nach einer geeigneten Strategie Aufwände für Inspektionen möglichst effizient zu nutzen. Werden Entwicklungsartefakte und Code im Entwicklungsprozess wieder verwendet, erweitert oder korrigiert, sollten diese Artefakt mit einem möglichst optimalen Aufwand inspiziert (und natürlich auch getestet) werden. Dazu sind insbesondere redundante Prüfungen, soweit dies möglich ist, zu vermeiden. Die Flexibilität der Inspektionen erlaubt eine systematische Anpassung der Technik an einen Entwicklungsprozess, in dem Wiederverwendung systematisch oder weniger systematisch betrieben wird. In [DT04] werden hierzu initiale

Ideen beschrieben, die noch weiter auszuarbeiten sind.

Eine allgemeine Betrachtung bestehender Qualitätsmanagementansätze und der industriellen Praxis zeigt, dass, unabhängig vom Kontext der Wiederverwendung, Inspektionen und vor allem Testen die wesentlichen Techniken bestehender Qualitätssicherungsstrategien sind. Allerdings werden die Techniken meist in Isolation voneinander betrachtet und eingesetzt und eine Nutzung möglicher Synergieeffekte wird bisher nicht oder nur unzureichend berücksichtigt [APW02]. Die Frage nach möglichen Kombinationen ist aber insbesondere im Zuge zunehmender Qualitätsansprüche bei gleichzeitig kürzeren Entwicklungszeiten und Kostenrestriktionen eine entscheidende Fragestellung. Verschiedene empirische Studien haben initial die Frage nach einer möglichen Kombination von Inspektionen und Testen untersucht [BS87, KL95, JV03, TAR+03]. Allerdings zielen diese Untersuchungen meist auf die allgemeine Fragestellung ab, welche der Techniken effektiver bzw. effizienter ist, wobei die Betrachtung nach Kombinationsmöglichkeiten, insbesondere über verschiedene Entwicklungsphasen hinweg unberücksichtigt bleibt. Die Motivation verschiedene qualitätssichernde Techniken miteinander zu verknüpfen zeigen verschiedene Studien die Nachweisen, dass ein gemeinsamer Einsatz der Techniken eine höhere Effektivität in der Fehlerfindung zur Folge hat als ein isolierter Einsatz [z.B. SSC+02]. Diese Studien betrachten die Anzahl der gefundene Fehler, wobei z.T. Aufwände aber insbesondere die Arten gefundener Fehler nicht berücksichtigt werden. Somit kann nur bedingt eine Qualitätssicherungsstrategie abgeleitet werden, welche die zur Verfügung stehenden Ressourcen möglichst optimal zur Erreichung der angestrebten Qualität einsetzt. Bestehende Qualitätsmanagementansätze berücksichtigen darüber hinaus

den Aspekt der Wiederverwendung und daraus resultierende Herausforderungen und Chancen in unzureichender Form. Hier fehlt es vor allem an einer systematischen Vorgehensweise, wie die Qualität von Komponenten und Artefakte über verschiedene Varianten und Produkte hinweg sichergestellt werden kann und welche Informationen durch den Wiederverwendungskontext zur Fokussierung und Steuerung aktueller Qualitätssicherungsaktivitäten genutzt werden können. Weiter ist zurzeit ungeklärt, wie aus Informationen über die Qualität von Komponenten in verschiedenen Produkten, Versionen und Releases systematisch Rückschlüsse auf Optimierungen der Qualitätssicherungsstrategie gezogen werden können.

Zur Entwicklung eines ganzheitlichen Qualitätsmanagementansatzes werden neben Ideen zur Anpassung und Kombination von Techniken insbesondere Fehlerinformationen benötigt. Nur durch kontinuierliches Messen und Bewerten von Fehlerinformation, die im Entwicklungsprozess gesammelt werden, können quantitative Daten erfasst werden, die eine Anpassung und Kombination von Techniken erlauben. Dazu werden entsprechende Modelle benötigt. Hier bieten Fehlerklassifikationsschemata [IEEE, Gra92, CBC92, IBM] und Fehlerstrommodelle [FDK05] einen ersten Ansatzpunkt. Die Idee der Fehlerklassifikationen ist es, aus Fehlerinformationen, die Effektivität und Effizienz und qualitätssichernden Techniken über den gesamten Entwicklungsprozess hinweg zu bewerten. In [CHB+93] und z.T. in [Gra92] wird aufbauend auf dem Klassifikationsschema von IBM bzw. HP beschrieben, wie Fehlerinformationen genutzt werden können, um Test- und Inspektionsaktivitäten zu bewerten (welche Arten von Fehlern werden gefunden), Fehlerprofile zu entwickeln (welche Technik sollte welche Fehler typischerweise finden) und Fehlerursachen systematisch zu identifizieren (was war der Grund für das Auftreten des Fehlers). Solche Informationen, bilden die Basis zur Definition einer systematischen Qualitätssicherungsstrategie. Allerdings werden Aufwandsdaten für qualitätssichernde Techniken in diesen Ansätzen nicht berücksichtigt, was zur Ableitung einer optimierten Strategie Voraussetzung ist. Darüber hinaus werden „standardisierte“ Fehlerklassifikationen in den Ansätzen verwendet, die nicht ohne Anpassungen auf einen anderen Kontext (Wiederverwendung und KMU-Charakteristika) übertragbar sind. In [FDK05] wird dazu ein Verfahren beschrieben, wie Fehlerklassifikationen und Fehlerstrommodelle systematisch an den Entwicklungskontext eines Unternehmens angepasst werden können. Dieser Ansatz bietet die Ausgangsbasis für die Arbeiten in LifeCycleQM.

Als Fazit zum Stand der Wissenschaft lässt sich festhalten, dass für Unternehmen, die Wiederverwendung von Softwarekomponenten betreiben zurzeit keine Systematik zur effizienten Sicherstellung der Qualität der Produkte zur Verfügung steht. Daraus resultiert, dass Qualitätssicherungsaktivitäten ineffizient und sub-optimal eingesetzt werden, wodurch Kosten erhöht, Qualität der Produkte und die Produktivität der Entwicklung reduziert werden.

Stand der Technik

Qualitätssicherung und Qualitätsmanagement werden insbesondere in vielen kleinen und mittelständischen Unternehmen ad-hoc und manchmal unsystematisch durchgeführt. Maßnahmen wie Testen auf Modul- und Integrationsebene sowie Systemtests werden häufig unzureichend geplant und durchgeführt. In größeren Firmen etablierte Techniken der frühen Qualitätssicherung, wie zum Beispiel Anforderungs- oder Codeinspektionen, werden im Umfeld kleiner und mittelständischer Unternehmen bestenfalls ebenfalls ad-hoc und ohne zugrunde liegende Planungs- bzw. Managementaktivitäten durchgeführt. Qualitätssichernde Maßnahmen werden drüber hinaus ohne umfassende Planung über den Entwicklungszyklus hinweg durchgeführt, d.h., dass die Techniken sehr oft in Isolation und unabhängig voneinander eingesetzt werden. Informationen, die während den verschiedenen qualitätssichernden Aktivitäten entstehen, bleiben häufig ungenutzt, um eine systematischere Vorgehensweise in zukünftigen Projekten

bzw. Iterationen einer Produktvariante zu verwenden. So werden zum Beispiel Testergebnisse nicht systematisch hinsichtlich gefundener Fehlertypen und dafür benötigter Aufwände analysiert, um etwa Code- oder Anforderungsinspektionen in einer zukünftigen Variante zu steuern. Umgekehrt, werden Inspektionsergebnisse nicht verwendet, um nachfolgende Inspektions- oder Testaktivitäten zu fokussieren.

Ein optimales Umfeld zum Aufbau eines systematischen Qualitätsmanagementansatzes bietet die Wiederverwendung von Software. Über die Wiederverwendung lassen sich Fehlerinformationen der verschiedenen Produktvarianten, Releases und wieder verwendeter Komponenten nutzen, um zu analysieren mit welcher Test- und Inspektionstechniken Fehler gefunden bzw. nicht gefunden werden können. Die Erfahrungen des Fraunhofer IESE in einer Vielzahl von Projekten mit Industrie- und Forschungspartnern verdeutlichen in diesem Zusammenhang allerdings, dass Qualitätssicherung im Kontext von Wiederverwendung ebenfalls ad-hoc und unsystematisch geschieht. Viele Firmen berücksichtigen nicht die Charakteristika dieses Kontextes: Einsatz einer Komponente in mehreren Produktvarianten, Notwendigkeit aus Fehlern in wieder verwendeten Komponenten systematisch zu lernen, Notwendigkeit Fehler in einer Produktvariante oder einem Release frühzeitig zu beheben anstelle in einer Vielzahl von daraus generierten oder abgeleiteten Produktvarianten. Test- und Inspektionstechniken werden somit wie bei der traditionellen Einzelsystementwicklung eingesetzt. Dadurch werden Qualitätssicherungsaufwände sub-optimal genutzt und die Kosten der Entwicklung erhöht. Vielfach ergeben sich dadurch auch Verzögerungen in der Projektabwicklung selbst, da die Qualitätssicherung zum Flaschenhals in der Entwicklung wird. Es können zwar Entwicklungszeit und Aufwand durch die Wiederverwendung der Software reduziert werden, allerdings zahlt sich dies nicht aus, da die eingesparten Zeiten und Aufwände durch unangepasste Qualitätssicherungstechniken wieder verloren gehen.

Betrachtet man den Bereich der systematischen Wiederverwendung, zum Beispiel im Produktlinienkontext, wo Variabilitäten eine wesentliche Rolle spielen, ist festzustellen, dass hier noch weniger angepasste Qualitätssicherungsstrategien in der industriellen Praxis eingesetzt werden. Hier sind insbesondere Aspekte wie generische Anforderungen und Code-Komponenten zu betrachten, die dann auf einen Kunden angepasst (instanziiert werden). Es werden zur Prüfung der generischen Komponenten, wie oben erwähnt, die typischen Techniken der Einzelsystementwicklung eingesetzt. Synergieeffekte, die zum Beispiel durch eine Inspektion einer generischen Komponente und deren Instanzen entstehen können bleiben weitestgehend unberücksichtigt.

Eine weitere Voraussetzung für die Entwicklung eines ganzheitlichen, lebenszyklusorientierten Ansatzes ist die systematische Nutzung von Fehlerdaten im Qualitätsmanagement. Auch hier zeigt der Stand der Praxis, dass viele Firmen nicht die Bedeutung von Fehlerinformationen auf systematische Art und Weise verwenden. Werden Fehlerdaten gesammelt, so sind die am häufigsten vorhandenen Fehlerinformationen (unabhängig von der Größe der Firmen) die Anzahl gefundener Fehler und die Schwere des Fehlers (häufig in Klassifikationen wie z.B., Schwer, Mittel, Leicht). Diese Fehlerinformationen erlauben allerdings keine systematischen Auswertungen hinsichtlich potentieller Verbesserungsmaßnahmen der Entwicklungsaktivitäten bzw. der Qualitätssicherung. Eine Studie des Fraunhofer IESE hat gezeigt, dass sehr viele Firmen gar keine Fehlerinformationen erfassen (40% der befragten Firmen sammeln überhaupt keine Reviewdaten, 36% erfassen lediglich die Anzahl der Fehler, die in einem Review gefunden werden). Weiter zeigte die Studie, dass bei einer Datenerfassung in den seltensten Fällen eine systematische Auswertung dieser Daten durchgeführt wird: je nach Reviewtechnik, d.h., Walkthrough, Peer-Review oder Inspektion werden bei ca. 25% der Befragten die Daten überhaupt nicht ausgewertet, 50% führen einfache Berechnungen wie Durchschnittsbildung durch, während 25% systematische Auswertungen durchführen. Allerdings geht aus

der Studie nicht hervor, was genau unter systematischen Auswertungen verstanden wird. In Verbindung mit der Tatsache, dass häufig nur sehr abstrakte Daten erfasst werden ist anzunehmen, dass der Anteil der Firmen die tatsächlich systematische Auswertungen durchführen noch geringer ist.

Darüber hinaus sind heutige Werkzeuge nicht in der Lage ein Entwicklungsteam gezielt in der Erhebung von Fehlerdaten zu unterstützen. Hier mangelt es vor allem an der Möglichkeit, Fehlerdaten über den gesamten Softwareentwicklungsprozess hinweg zu sammeln und diese dann systematischen Analysen zuzuführen. Insbesondere der letzte Aspekt ist entscheidend für ein zielgerichtetes, ganzheitliches Qualitätsmanagement.

Zusammenfassend lässt sich somit sagen, dass bestehende qualitätssichernde Techniken und systematische Ansätze des Qualitätsmanagement sehr viel stärker an das Umfeld von kleinen und mittelständischen Unternehmen und an die Charakteristika der Softwarewiederverwendung angepasst werden müssen. Darüber hinaus fehlt es an systematischen Ansätzen Fehlerdaten der Softwareentwicklung über den Lebenszyklus zu sammeln und entsprechend auszuwerten, um eine Optimierung des Qualitätsmanagements über den Entwicklungsprozess hinweg erzielen zu können. Ansätze wie dies geschehen kann, fehlen in der täglichen Praxis von KMUs vollständig und sollen in dem Projekt LifeCycleQM erarbeitet werden. Da Datenerhebung und Datenanalyse in einem solchen Ansatz eine zentrale Rolle spielen, ist eine entsprechende Werkzeugunterstützung unbedingt notwendig.

Lösungsansatz und Vorteile

Das Gesamtziel des Vorhabens, die Entwicklung einer Qualitätsstrategie für den Lebenszyklus wieder verwendbare Software, stellt sicher, dass sich Wiederverwendung von Software in höherer Qualität und Produktivität auszahlen kann und Qualitätssicherungskosten signifikant reduziert werden, indem Wissen über die Qualität wieder verwendbarer Komponenten systematisch nutzbar gemacht wird. Dies erfordert die systematische Erfassung von entsprechenden Qualitätsinformationen sowie deren zielgerichtete Auswertung. Aufbauend darauf ist dann der gezielte und balancierte Einsatz von Qualitätssicherungsaktivitäten über den gesamten Lebenszyklus wieder verwendbarer Komponenten möglich. Die folgende Graphik verdeutlicht die Ausgangssituation für das Vorhaben.

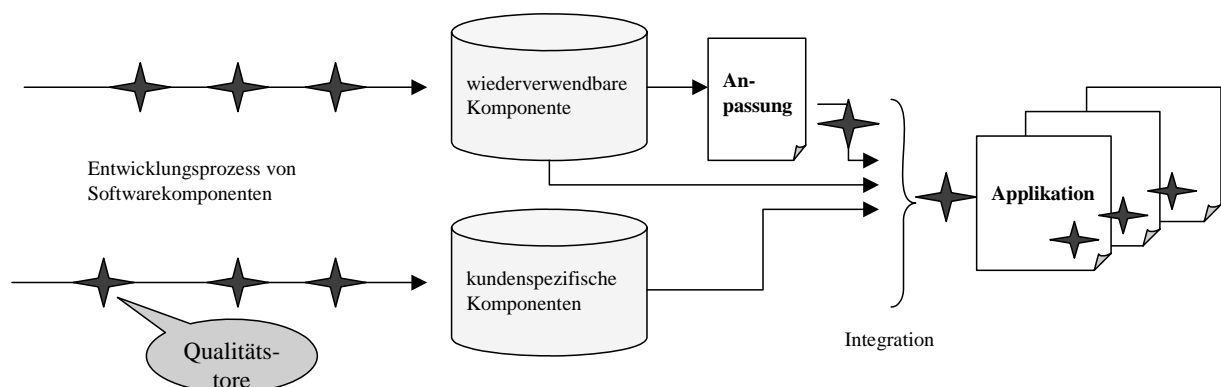


Abbildung 1: Ausgangssituation Entwicklung unter Wiederverwendung von Komponenten

Bei der Entwicklung von Applikationen für die einzelnen Kunden integrieren KMUs häufig kundenspezifische Komponenten, wieder verwendbare Komponenten und unterschiedliche Versionen (Anpassungen) der wieder verwendbaren Komponenten zu einem Gesamtprodukt. In der Entwicklung der Komponenten werden dabei verschiedene Softwareartefakte (z.B.

Design-Spezifikationen, Anforderungsdokumente, Code, etc.) erstellt. Je nach Entwicklungsprozess der Firma durchlaufen diese Artefakte und die daraus erstellten Komponenten unterschiedliche Qualitätstore von der initialen Entwicklung bis zur Integration zum Gesamtprodukt. Qualitätstore sind dabei definierte Punkte im Prozess, an denen die Qualität der Artefakte und der Komponente mit Hilfe entsprechender Techniken überprüft wird. Nach der Integration werden weitere qualitätssichernde Maßnahmen durchgeführt (zum Teil auf Seiten der entwickelnden Firmen, zum Teil auf Seiten der Kunden auf konkreten Applikationen). Sämtliche Qualitätstore liefern Informationen bezüglich der Qualität der einzelnen Artefakte und den darin beschriebenen Komponenten. Hierbei handelt es sich um Fehlerdaten (z.B., Anzahl von Fehlern in einem Produkt, Typen von Fehlern) und Aufwandsdaten (wie viel Aufwand wurde an den Qualitätstoren zur Qualitätssicherung aufgewendet).

Die Aufgabe eines systematischen Qualitätsmanagement ist es, diese Daten zu bewerten und zu einer integrierten Qualitätssicherungsstrategie zu verknüpfen. In einem Wiederverwendungskontext ergeben sich darüber hinaus Möglichkeiten Qualitätsdaten über verschiedene Versionen und Konfigurationen der Software hinweg zu analysieren. Diese können dann zusätzlich zur Definition der Qualitätssicherungsstrategie herangezogen werden. Ein solches zielgerichtetes Qualitätsmanagement fehlt zurzeit und soll im Rahmen dieses Vorhabens erstellt werden. Das Qualitätsmanagement nutzt dabei Fehler- und Aufwandsdaten zur Balancierung und Fokussierung der Qualitätssicherungsaktivitäten. Durch den Kontext der Wiederverwendung der Komponenten ist es dann möglich gezielt und über einen längeren Zeitraum die Fehlerdaten wieder verwendeter Komponenten (insbesondere der verschiedenen Varianten und Konfigurationen) zu analysieren, die Fokussierung durchzuführen und deren Auswirkungen auf die Applikation und andere Qualitätstore zu untersuchen. Die folgende Abbildung 2 erweitert Abbildung 1 um den zu erarbeitenden Qualitätsmanagementansatz.

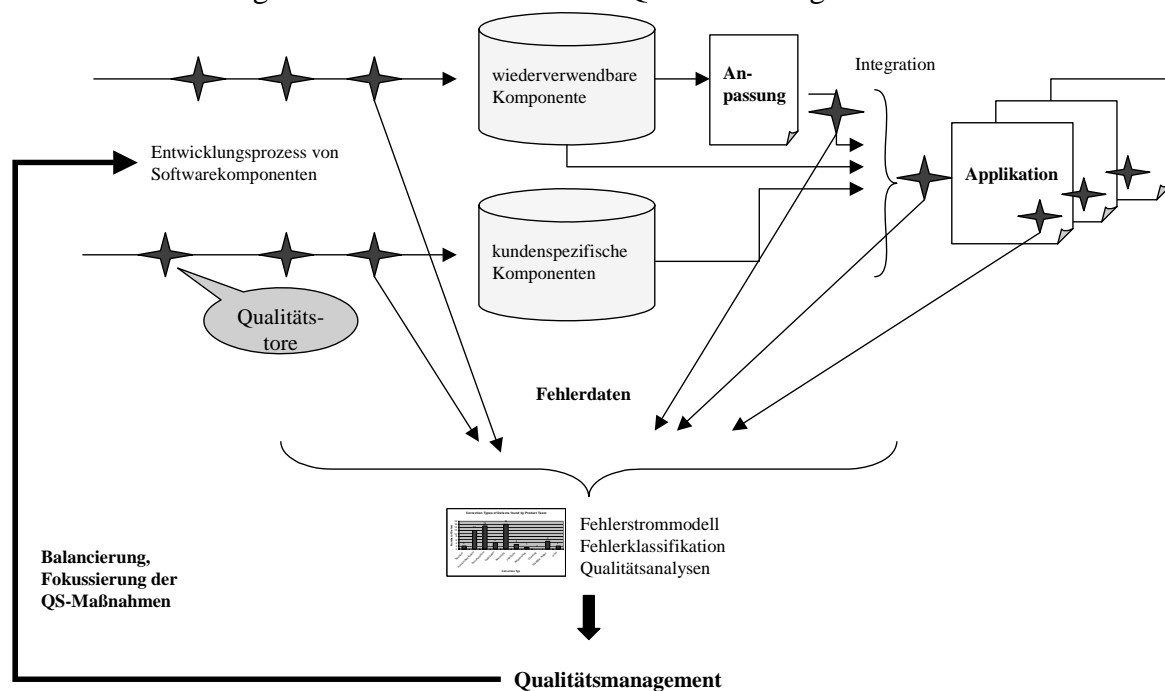


Abbildung 2: Ganzheitliches Qualitätsmanagement für wieder verwendbare Komponenten

Daten aus allen Phasen des Lebenszyklus einer Komponente werden in diesem Ansatz integriert. Mit Hilfe von Fehlerstrommodellen und Fehlerklassifikationen sowie statistischen Analysen werden dem Qualitätsmanagement aufbereitete Daten zur Verfügung gestellt, die es erlauben die Qualitätstore über den Lebenszyklus zu fokussieren. Bei der Auswertung spielen

insbesondere Fehlerdaten der wieder verwendbaren Komponenten eine große Rolle, da diese Daten bzgl. der Qualität über mehrere Varianten, Konfigurationen und Produkte hinweg bereitstellen, die wiederum zur Fokussierung der aktuellen Qualitätssicherungsaktivitäten herangezogen werden können.

Das Gesamtvorhaben untersucht daher im Wesentlichen folgende Fragestellungen:

- Wie lassen sich bestehende Test- und Inspektionstechniken an den Wiederverwendungskontext anpassen, damit die dort geltenden Faktoren möglichst effizient berücksichtigt werden?
- Wie können Informationen zur Qualität und zum Qualitätssicherungsprozess wieder verwendeter Softwarekomponenten genutzt werden, um aktuelle Qualitätssicherungsaktivitäten zu fokussieren?
- Wie lassen sich Test- und Inspektionstechniken, aufbauend auf erfassten Fehlerdaten, zu einer Qualitätssicherungsstrategie verbinden, die vorgegebene Qualitätsziele möglichst effizient im Lebenszyklus der Software sicherstellt?

Im Rahmen des Projektes werden diese Fragestellungen beantwortet und entsprechende Verfahren und Vorgehensweisen definiert. Aufbauend darauf wird die Hypothese untersucht, ob und in wie weit:

- an den Kontext von Wiederverwendung angepasste Techniken eine effizientere und systematischere Qualitätssicherung als Standardmethoden ermöglichen
- eine ganzheitliche Qualitätssicherungsstrategie über den Lebenszyklus hinweg gewünschte Qualitäten von Endprodukten effizienter sicher stellt als bisherige Strategien

Folgende Abbildung zeigt die wesentlichen Konzepte des angestrebten Lösungsansatzes:

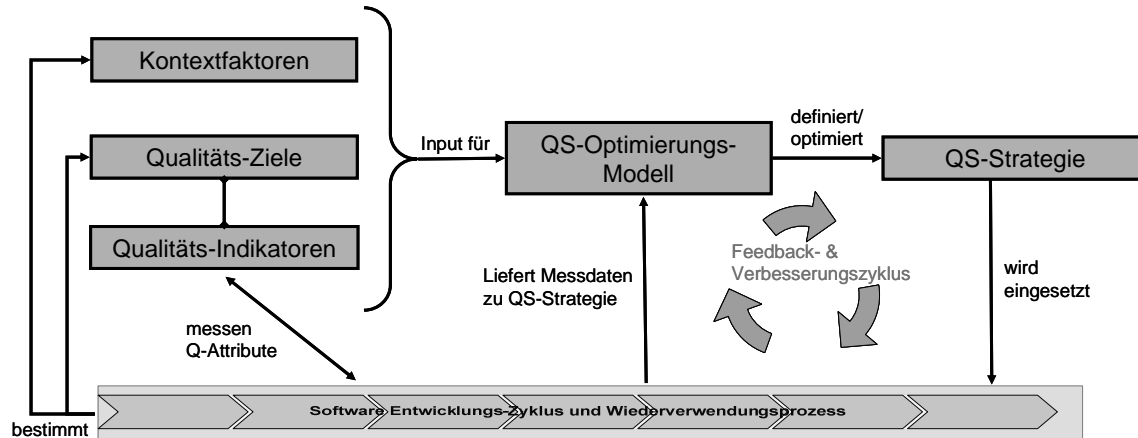


Abbildung 3: Komponenten des Ansatzes

Im Zentrum des Ansatzes steht die Definition eines Modells, das aufbauend auf Qualitätsdaten aus dem Softwareentwicklungsprozess Qualitätsmanager oder Projektmanager in der Definition einer kontext- und projektspezifischen Qualitätssicherungsstrategie unterstützt. Um ein solches Modell definieren zu können werden Konzepte und Verfahren erarbeitet, die dabei unterstützen in einem gegebenen Kontext

- systematisch Kontextfaktoren erheben,
- Qualitätsziele zu identifizieren und
- Qualitätsindikatoren zu definieren.

Ein Verfahren zur Erhebung der Kontextfaktoren wird dazu erstellt, welches typische Konzepte von Assessments beinhaltet aber auf den LifeCycleQM-Fokus angepasst ist. Dieses Verfahren unterstützt die Identifikation eines verbindlichen Entwicklungsprozess in einem gegebenen Kontext, darin enthaltene (relevante) konstruktive- und QS-Aktivitäten zu definieren, für die Qualitätssicherung relevante Arbeitsprodukte zu identifizieren sowie die Bestimmung der wesentlichen Restriktionen im Entwicklungsprozess (Kosten, Qualität, Zeit).

Als zweiter wesentlicher Aspekt wird ein Verfahren entwickelt, das die Definition von Qualitätszielen erlaubt. Die zugrunde liegende Idee ist, dass sich die Ziele durch Qualitätsattribute des Endproduktes und deren Wichtigkeit bestimmen. So haben etwa Safety, Security, Reliability, Maintainability, Reusability unterschiedliche Bedeutungen für den Erfolg eines Softwareproduktes. Diese Unterschiede müssen bei der Definition der Qualitätsstrategie berücksichtigt werden. Dazu wird ein Verfahren entworfen, das es erlaubt ein so genanntes Qualitäts-Profil in einem gegebenen Projektkontext zu erheben. Folgende Abbildung zeigt ein Beispiel für ein solches Qualitätsprofil:

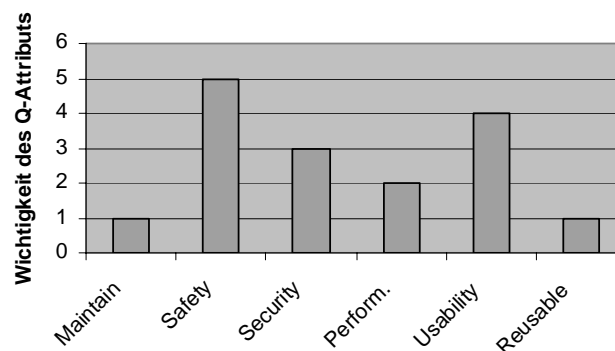


Abbildung 4: Beispiel eines Qualitätsprofils

Die x-Achse zeigt die Qualitätsattribute, die für das Endprodukt relevant sind, die y-Achse zeigt die Wichtigkeit der Attribute für den Erfolg des Endproduktes. Dabei werden wir auf bewährte Standards (z.B. ISO 9126 Standard) aufsetzen und diese entsprechend an die Bedürfnisse von LifeCycleQM anpassen.

Das Qualitätsprofil wird weiter verfeinert durch Qualitäts-Indikatoren. Dieser werden für jede Entwicklungsphase definiert. Sie stellen die Verbindung zwischen den zu erreichenden Qualitäten und Aspekten in verschiedenen Arbeitsprodukten der Softwareentwicklung her. Erst die Definition der Indikatoren erlaubt es die Qualitätssicherungstechniken gezielt auf gewünschte Endproduktqualitäten auszurichten.

Um solche Indikatoren und relevante Qualitätsattribute definieren zu können, werden wir ein entsprechend angepasstes Messprogramm definieren. Dieses ist angelehnt an Ideen des Fehlerstrommodells und bestehender Fehlerklassifikationsschemata. Allerdings werden diese bestehenden Ansätze gezielt an den Wiederverwendungskontext angepasst. Das heißt, sie werden erweitert um Informationen und Aspekte die nur bei der Wiederverwendung von Software erhoben und ausgewertet werden können.

Das Messprogramm wird im gesamten Entwicklungszyklus implementiert, um eine ganzheitliche Sichtweise auf die Effektivität und Effizienz bestimmter Qualitätstore und deren Auswirkung auf die Qualität des Endproduktes bewerten zu können. Die so erfassten Daten werden analysiert und in ein entsprechendes (mathematisches) Modell umgesetzt, das über die Zeit (d.h. mit dem Vorhandensein von immer mehr Daten) die Ableitung und Definition optimaler Qualitätssicherungsstrategien erlaubt. Daher ist das Projekt LifeCycleQM iterativ angelegt, um kontinuierlich Messdaten erfassen zu können und diese in Verfeinerungen des

Modells umsetzen zu können. Wie oben erwähnt spielen in dieser Betrachtung insbesondere Informationen eine Rolle, die im Wiederverwendungskontext erhoben werden können:

- Welche Fehler sind in der wieder verwendeten Software aufgetreten?
- Welche QS-Techniken wurden bereits auf die wieder verwendete Software angewendet?
- Wie spielt die wieder verwendete Software mit den kundenspezifischen Teilen zusammen?
- Auf welche Qualitäten des Endproduktes haben die wieder verwendeten Teile Einfluss?

Als Ergebnis des Modells wird eine an den Kontext angepasste QS-Strategie erwartet. Dazu werden zum einen Inspektions- und Test-Techniken entwickelt, die an den Kontext wieder verwendeter Software angepasst sind. Darüber hinaus wird eine Strategie entwickelt, wie diese Techniken optimal im Entwicklungszyklus eingesetzt, d.h. kombiniert werden.

Auf technischer Seite steht die Entwicklung eines Werkzeuges im Mittelpunkt des Vorhabens. Dieses Werkzeug soll die entwickelte Qualitätsstrategie für wieder verwendbare Software durch Datenerhebungen und systematischen Analysen (Auswertungen) unterstützen. Dazu werden in einem iterativen Vorgehen, die erworbenen und konsolidierten wissenschaftlichen Ergebnisse in dem Werkzeug umgesetzt.

Projektstatus

Das Projekt LifeCycleQM hat eine Laufzeit von Januar 2006 bis Dezember 2008 und ist ein Verbundprojekt der Partner IBS AG, Fraunhofer IESE und Polarion. Im Rahmen des Arbeitspaketes 1 erfolgte bis Ende Juni 2006 zunächst eine Standortbestimmung und systematische Analyse des Entwicklungs- und Qualitätssicherungsprozesses bei dem Industriepartner IBS AG. Dazu wurden zum einen, angelehnt an typische Assessmentverfahren, strukturierte Interviews mit Prozessverantwortlichen und Experten geführt und Prozessdokumentationen analysiert. Die Analyse der eingesetzten Qualitätssicherungsaktivitäten hinsichtlich ihrer Effektivität (Anzahl gefundener Fehler, welche Fehler werden entdeckt) und ihrer Effizienz (gefundene Fehler in Relation zum eingesetzten Aufwand) erfolgte mit Hilfe statistischer Auswertungen bestehender Fehlerdatenbanken und Testergebnisse. Dieses wurde ergänzt durch Experteninterviews zur Erfassung des qualitativen Bildes bezüglich der Qualität (welche Probleme bestehen zurzeit) und der Qualitätssicherungsaktivitäten. Aufbauend auf diesen Ergebnissen erfolgt aktuell die Entwicklung und Einführung eines Fehlerstrommodells und der zugehörigen Fehlerklassifikation. Dazu wird ein am Fraunhofer IESE entwickeltes Verfahren zur Definition und Einführung von Fehlerstrommodellen und Klassifikationen an den Kontext von KMUs und insbesondere an den Kontext der Wiederverwendung angepasst und mit dem Anwendungspartner durchgeführt. In späteren Phasen des Projekts werden Qualitätstore für den Entwicklungs- und Wiederverwendungsprozess bei dem Anwendungspartner definiert. Um den Qualitätsmanagementansatz möglichst effizient für KMUs einsetzbar zu machen und die Ableitung einer Qualitätsstrategie zu ermöglichen wird die IBS AG in Kooperation mit Polarion ein Werkzeug zur Visualisierung der Fehlerströme im Lebenszyklus wieder verwendbarer Komponenten erstellen.

Zusammenfassung und Ausblick

Das Ziel des Vorhabens LifeCycleQM ist es das Problem der Qualitätssicherung wieder verwendbarer Softwarekomponenten im Kontext der Informationssystementwicklung zu verbessern. Zu diesem Zweck wird eine ziel- und lebenszyklus-orientierte Qualitätsstrategie zur Optimierung des Qualitätsmanagements entwickelt. Mit Hilfe der Qualitätsstrategie, die Wissen über die Qualität wieder verwendbarer Komponenten systematisch nutzbar macht, soll sichergestellt werden, dass sich Wiederverwendung von Software in höherer Qualität und Produktivität auszahlen kann und Qualitätssicherungskosten signifikant reduziert werden. Der Hauptbeitrag des Vorhabens ist ein Verfahren zur Definition kontext-spezifischer Qualitätssicherungsstrategien die unterschiedliche QS-Maßnahmen in verschiedenen Entwicklungsstufen sinnvoll miteinander kombinieren. Dazu werden a) ein Verfahren zur Bestimmung von Qualitätszielen, b) Inspektions- und Testverfahren, die an den Kontext von Wiederverwendung angepasst sind, und c) ein Modell zur „Berechnung“ einer Qualitätssicherungsstrategie aufbauend auf Qualitätszielen, Kontextfaktoren und Rahmenbedingungen entwickelt. Das entwickelte Verfahren und zugrunde liegende Modell werden in Form eines Werkzeuges implementiert und durch die Industriepartner in künftigen Softwareentwicklungsprojekten eingesetzt.

Literatur

- [AB+01] C. Atkinson et al., Component-based Product Line Engineering with UML, Addison-Wesley, 2001.
- [APW02] A. Aurum, H. Petersson, C. Wohlin. State-of-the-Art: Software Inspections Turning 25 Years. *Journal on Software Testing, Verification and Reliability*, 12(3):133–154, 2002.
- [BB01] B. W. Boehm, V. R. Basili. Software Defect Reduction Top 10 List. *IEEE Computer* 34(1):135–137, 2001.
- [BK+04] G. Böckle, P. Knauber, K. Pohl, K. Schmid. *Software-Produktlinien – Methoden, Einführung und Praxis*, dpunkt-Verlag, 2004.
- [BS87] Basili, V.R.; Selby, R.W.; Comparing the Effectiveness of Software Testing Techniques, *IEEE Transactions on Software Engineering* Vol. SE 13, No. 12, pp. 1278 – 1295, 1987.
- [CBC+92] R. Chillarege et al., Orthogonal defect classification – A concept for in-process measurements, *IEEE Transactions on Software Engineering*, 1992.
- [CHB+93] J. K. Char et al., In-Process Evaluation for Software Inspection and Test, *IEEE Transaction on Software Engineering*, 1993.
- [CN01] P. Clements and L. M. Northrop, *Software Product Lines: Practices and Patterns*. Addison-Wesley, 2001.
- [DT04] C. Denger, H. Teranishi. Inspections in Reuse Intensive Software Development Processes. *Proceedings of the First International Workshop on Quality Assurance in Reuse Contexts (QUARC)*, 2004.
- [Fag76] M. E. Fagan. Design and Code Inspections to Reduce Errors in Program Development, *IBM Systems Journal* 15(3):182–211, 1976.

- [FDK05] Freimut, B.; Denger, C.; Kettterer M.; An Industrial Case Study of Implementing and Validating Defect Classification for Process Improvement and Quality Management. Proceedings of the 11th International Software Metrics Symposium (Metrics), 2005.
- [Gra92] Robert B. Grady, Practical Software Metrics for Project Management and Process Improvement. Prentice Hall, 1992.
- [GG93] T. Gilb, D. Graham. Software Inspection. Addison-Wesley, 1993.
- [IBM] IBM, Center for Software Engineering
<http://www.research.ibm.com/softeng/ODC/ODC.HTM>
- [IEEE] Institute of Electrical and Electronics Engineers, IEEE Standard Classification for Software Anomalies, IEEE Std. 1044-1993, 1994.
- [JV03] N. Juristo, S. Vegas, Functional testing, structural testing and code reading: what fault type do they each detect?, Empirical Methods and Studies in Software Engineering – Experiences from ESERNET, 2003.
- [KL95] E. Kamsties, C. M. Lott, An Empirical Evaluation of Three Defect Detection Techniques, Proceedings of the 5th European Software Engineering Conference, 1995.
- [KM03] R. Kolb and D. Muthig. Challenges in Testing Software Product Lines. In Proceedings of CONQUEST'03, pp. 81–95, Nürnberg, Germany, September 2003.
- [KMM04] R. Kolb, John D. McGregor, and D. Muthig, Proceedings of the First International Workshop on Quality Assurance in Reuse Contexts (QUARC), Boston, MA, Fraunhofer IESE Report No. 096.04/E, August 2004.
- [KP+03] E. Kamsties, K. Pohl, S. Reis, A. Reuys. Testing Variabilities in Use Case Models. In Proceedings of 5th International Workshop on Product Family Engineering (PFE-5). Springer Verlag, Siena, Italien, November 2003.
- [McG01] J. D. McGregor. Testing a Software Product Line. Technical Report, CMU/SEI-2001-TR-022, Software Engineering Institute, Carnegie Mellon University, December 2001.
- [MH03] H. Muccini and A. van der Hoek. Towards Testing Product Line Architectures. In ETAPS 2003 Workshop Test and Analysis of Component Based Systems (TaCOS), Warsaw, Poland, April 2003.
- [SSC+02] S. S. So, S. D. Cha, T. J. Shimeall, Y. R. Kwon, An Empirical Evaluation of Six Methods to Detect Faults in Software, Software Testing, Verification and Reliability, 2002.
- [TAR+03] T. Thelin et al., An Experimental Evaluation of Inspection and Testing for Detection of Design faults, Proceedings of the 2003 International Symposium of Empirical Software Engineering, 2003.
- [Wie02] K. E. Wiegers. Peer Reviews in Software – A Practical Guide. Addison-Wesley, 2002.