

dass er nur in Gegenwart eines Hardware-Schlüssels erfolgen kann, den der zugangsberechtigte Mitarbeiter bei sich trägt. Eine solche Lösung würde das Problem der Trennung von Privatem und Beruflichem in dem Fall vereinfachen, dass, wenn der betreffende Mitarbeiter das Unternehmen verlässt, lediglich der Schlüssel abgegeben werden müsste.

Quellen

- [01] <http://www.ercis.org/publication/401>
- [02] <http://www.gartner.com/technology/reprints.do?id=1-1AKKJNN&ct=120518>

Fabian Schomm
fabian.schomm@uni-muenster.de

Gottfried Vossen
g.v@wwu.de

	MobileIron	AirWatch	Good	Fiberlink	Zenprise
In-House App Store	grün	grün	grün	grün	grün
Push Apps	grün	grün	grün	grün	grün
Remote-Konfiguration	grün	grün	grün	grün	grün
Push-Benachrichtigungen	grün	grün	grün	grün	grün
Inventarisierung	grün	grün	grün	grün	grün
Backup/Restore	grün	grün	grün	grün	grün
Remote Lock & Wipe	grün	grün	grün	grün	grün
Containerisierung	rot	grün	grün	grün	grün
Zugriffskontrolle	grün	grün	grün	grün	grün
SaaS-Angebot	grün	grün	rot	grün	grün
On-Premise-Deployment	grün	grün	grün	rot	grün
Android	grün	grün	grün	grün	grün
BlackBerry	grün	grün	rot	grün	grün
iOS	grün	grün	grün	grün	grün
Symbian	grün	grün	rot	grün	grün
Windows Phone	grün	grün	grün	grün	grün

Tabelle 1: grün = vorhanden, rot = nicht vorhanden

Qualitätssicherung in Integrationsprojekten

Dr. Michael Gebhart, Gebhart Quality Analysis (QA) 82

Die Automatisierung von Geschäftsprozessen im Rahmen des Business Process Management erfordert meist die Integration bestehender Anwendungen. Neben der rein technischen Umsetzung wird von Integrationsprojekten dabei zunehmend gefordert, eine flexible und wartbare Lösung zu erstellen, wobei die Realisierung gleichzeitig kosteneffizient zu erfolgen hat. Um die qualitativen Eigenschaften sicherzustellen, ist eine Abstimmung zwischen allen Beteiligten erforderlich, die wiederum mit hohem Aufwand einhergeht. Der vorliegende Artikel zeigt daher auf, wie die Erstellung eines Qualitätsmodells dabei helfen kann, die Qualitätssicherung in Integrationsprojekten zu strukturieren und gleichzeitig die Effizienz zu steigern.

Unternehmen streben zunehmend eine Automatisierung von Geschäftsprozessen im Rahmen eines ganzheitlichen Business Process Management (BPM) an. Ziel ist es dabei, die Geschäftsprozesse zu optimieren und gleichzeitig die Kosten für die Durchführung zu reduzieren. Aufgrund der steigenden Zahl betriebener Anwendungen, die in den Geschäftsprozessen genutzt werden sollen, resultieren derartige Vorhaben verstärkt in Integrationsprojekten. Neben der rein tech-

nischen Umsetzung, also der technischen Integration von Anwendungen in neu umzusetzende Geschäftsprozesse, sind mit derartigen Projekten jedoch auch strategische Ziele verknüpft: So geht damit häufig auch eine Restrukturierung der IT einher, wie es insbesondere beim Wechsel hin zu serviceorientierten Architekturen der Fall ist. Unabhängig von dem Paradigma wird von der resultierenden IT-Architektur ein hohes Maß an Nachhaltigkeit erwartet, sodass diese auf

der einen Seite kostengünstig gewartet werden kann, sich aber auch flexibel an neue geschäftliche Anforderungen anpassen lässt.

Gleichzeitig hat sich in der Vergangenheit der Druck auf die Softwarebranche weiter erhöht. So wird verlangt, dass derartige Lösungen in noch kürzerer Zeit mit noch geringeren Kosten umgesetzt werden können. Bei Projekten dieser Größe sind eine regelmäßige Abstimmung zwischen allen Beteiligten sowie eine Qualitätssicherung

erforderlich, um die gewünschten Ziele zu erreichen. Es existiert also die Herausforderung, eine qualitativ hochwertige Lösung zu erstellen, die sich kosteneffizient realisieren lässt. Das Qualitätsmanagement ist somit gefordert, diesem Konflikt gerecht zu werden. Mithilfe eines Qualitätsmodells kann eine strukturierte und letztlich effiziente Qualitätssicherung in Integrationsprojekten erfolgen.

Aufbau eines Qualitätsmodells

Die Idee eines Qualitätsmodells in der Softwaretechnik entsteht vor allem aus dem Bedarf heraus, die Qualität von Software bestimmen zu können. Die in der Software-Technik etablierten Qualitätsmerkmale wie Wartbarkeit und Flexibilität sind zunächst nicht direkt ersichtlich. Es ist daher eine Beschreibung erforderlich, wie diese abstrakten Qualitätsmerkmale auf konkreten Software-Artefakten ermittelt werden können.

Ein Qualitätsmodell nimmt eine derartige Beschreibung durch schrittweise Verfeinerungen vor. Im Folgenden wird dabei vor allem die 1977 von McCall et al. eingeführte Methode „Factor, Criteria, Metric“ (FCM) betrachtet [1]. Ein Qualitätsmerkmal (Factor) wie die Wartbarkeit wird dabei zunächst in ein oder mehrere Qualitätsteilmerkmale (Criteria) zerlegt. Dieser Schritt wird solange fortgesetzt, bis sich ein Qualitätsteilmerkmal ergibt, das auf konkreten Software-Artefakten bestimmt werden kann. Hierbei handelt es sich um einen Qualitätsindikator, also einen Hinweis auf die Ausprägung eines Qualitätsteilmerkmals.

Um einen Qualitätsindikator zu beschreiben, werden häufig Metriken eingesetzt, die den betrachteten Qualitätsindikator quantifizieren. Dazu ist eine Abbildung des Qualitätsindikators auf konkrete Software-Artefakte und ihre Bestandteile erforderlich. Abbildung 1 veranschaulicht den Aufbau eines Qualitätsmodells.

Es gilt hierbei zu berücksichtigen, dass es sich bei dem Begriff der Qualität um etwas Individuelles handelt. So existieren zwar standardisierte Qualitätsmodelle wie ISO/IEC 9126 beziehungsweise ISO/IEC 25000 im Kontext von Software-Produkten. Diese sind jedoch zunächst so abstrakt gehalten, dass sie sich ohne weitere Verfeinerung nicht in Projekten anwenden lassen. Zudem ist zu bedenken, dass ein Qualitätsmodell

nur einen Hinweis darauf gibt, inwieweit ein bestimmtes Qualitätsmerkmal ausgeprägt ist. Es findet keine Wertung statt. Dies ist schon allein deswegen nicht möglich, da sich Qualitätsmerkmale auch widersprechen können. Es liegt daher immer noch in der Verantwortung des Qualitätsmanagers zu entscheiden, ob die Ergebnisse einer Qualitätsanalyse zufriedenstellend sind oder nicht.

Die Erstellung eines Qualitätsmodells bringt dennoch zwei zentrale Vorteile mit sich: Zum einen ermöglicht ein Qualitätsmodell die Bestimmung der Qualität und zeigt überhaupt auf, welche Qualitätsmerkmale inwieweit ausgeprägt sind. Auch im Falle einer manuellen Prüfung kann ein Qualitätsmodell stets als Kriterienkatalog herangezogen werden, da es beschreibt, welche Aspekte relevant sind und worauf geachtet werden sollte. Es schafft somit die Grundlage für zielgerichtete Überarbeitungen. Selbst wenn in einer bestimmten Projektsituation keine Optimierung möglich ist, wird dennoch offensichtlich, dass gewisse Defizite existieren, auch wenn diese nicht beseitigt werden können. Auf diese Weise können jedoch gegebenenfalls notwendige Vorkehrungen getroffen werden.

Zum anderen erinnert ein Qualitätsmodell stets daran, welche Auswirkungen eine Änderung haben kann. So wird ersichtlich, dass eine – wenn auch nur kurzzeitige – Namensänderung einer Webservice-Operation im JDeveloper von „manufacture“ in „execute“ die funktionale Benennung des Service betrifft und somit die Auffindbarkeit und letztlich Wartbarkeit der Architektur reduzieren kann. Abbildung 2 zeigt, wie die Auswirkungen derartiger Änderungen im JDeveloper aufgezeigt werden können. Ein Qualitätsmodell schafft also einerseits Transparenz, andererseits sensibilisiert es alle Beteiligten für das Thema „Qualität“.

Erstellung eines Qualitätsmodells

Um von den genannten Vorteilen zu profitieren, muss ein Qualitätsmodell erstellt werden, das die individuellen Anforderungen abbildet und demnach den jeweiligen kunden-, branchen- und projektspezifischen Bedingungen entspricht. Die Herausforderung liegt dabei darin zu erkennen, wovon der Erfolg eines Projekts abhängt, und diese Faktoren entsprechend zu be-

nennen, zu strukturieren und schließlich auf Elemente in konkreten Software-Artefakten abzubilden. Hierzu ist ein hohes Fachwissen in dem jeweils angewendeten Paradigma und gleichzeitig über die eingesetzten Technologien erforderlich.

Die Erstellung eines Qualitätsmodells kann dabei auf verschiedene Varianten erfolgen: Top-down oder Bottom-up. Im ersten Fall wird das Qualitätsmodell beginnend mit Qualitätsmerkmalen erstellt. Hierfür kann beispielsweise das Qualitätsmodell aus ISO/IEC 9126 beziehungsweise ISO/IEC 25000 herangezogen werden. Dies gibt erste Anhaltspunkte darüber, welche Qualitätsmerkmale für Software-Produkte existieren. Anschließend müssen diese für das genutzte Paradigma verfeinert werden, es muss also überlegt werden, in welcher Weise sich diese Qualitätsmerkmale und -teilmerkmale in der zu erstellenden Architektur widerspiegeln.

Ebenso können hier neben reinen Software-Artefakten auch Projektmanagement-Artefakte eine Rolle spielen. Der Vorteil dieses Vorgehens liegt darin, dass ein ganzheitliches Qualitätsmodell erstellt wird. Auf der anderen Seite erzeugt dieser Ansatz einen erheblichen Aufwand. Dieser kann reduziert werden, indem nur ausgewählte Qualitätsmerkmale berücksichtigt werden. Liegt der Schwerpunkt eines Projektes auf der Wartbarkeit und Flexibilität der Lösung, so kann sich im Rahmen der Erstellung des Qualitätsmodells auf diese Qualitätsmerkmale beschränkt werden.

Der Bottom-up-Ansatz betrachtet zunächst die Erstellung von Qualitätsindikatoren beziehungsweise Metriken und ordnet diese im Nachhinein bestimmten Qualitätsteilmerkmalen und somit Qualitätsmerkmalen zu. Ein Ansatz, um relevante Metriken zu identifizieren, stellt die Goal-Question-Metric-Methode dar, die 1984 von Basili und Weiss [2] eingeführt wurde. Hierbei werden ausgehend von Zielen konkrete Fragestellungen formuliert, die beantwortet werden müssen, und hierfür geeignete Metriken ausgearbeitet. Dieser Ansatz geht somit stark davon aus, dass zunächst definiert wird, was für eine bestimmte Software relevant ist. Diese Informationen werden anschließend mithilfe der Metriken zusammengetragen. Wesentlicher Vorteil dieses Ansatzes ist es, dass sich die

Metriken von Beginn an direkt an den Bedürfnissen orientieren. Häufig jedoch ist am Anfang nicht direkt ersichtlich, welche Aspekte überhaupt relevant sind. Ebenso droht die Gefahr, ein sehr unvollständiges Bild zu schaffen. So muss darauf geachtet werden, dass die Metriken nicht nur die Aspekte abbilden, die in der Vergangenheit zu Schwierigkeiten geführt haben und die daher verstärkt geprüft werden sollen. Werden diese Schwierigkeiten etwa auf Kosten anderer Qualitätsmerkmale behoben, so ist dies nicht offensichtlich, da entsprechende Metriken nicht abgebildet wurden. Es gilt daher, einen Kompromiss zwischen dem Umfang und der Fokussierung des Qualitätsmodells zu finden.

Darüber hinaus ist die Auswahl relevanter Qualitätsmerkmale mit großer Sorgfalt vorzunehmen. Hierfür sollte sich zunächst bewusst gemacht werden, worauf das zu erstellende Qualitätsmodell abzielt. So kann die Sicherstellung der funktionalen Korrektheit beispielsweise durch eines der zahlreichen Testwerkzeuge erfolgen. Syntaktische

Korrektheit der zu erstellenden Artefakte erfolgt dabei ebenfalls bereits innerhalb der genutzten Entwicklungsumgebung. Ein wie in Integrationsprojekten genutztes Qualitätsmodell kann diese Aspekte mit einschließen, sollte jedoch vor allem noch nicht in Betracht gezogene Bereiche abdecken. Als Beispiel kann die Berücksichtigung von Best-Practices genannt werden, die eher auf konzeptioneller Ebene angesiedelt sind und nicht direkt offensichtlich oder durch bereits bestehende Werkzeuge geprüft werden.

Einsatz bestehender Qualitätsmodelle

Insgesamt stellt der Aufbau eines Qualitätsmodells somit eine Herausforderung dar, deren Bewältigung einen nicht zu unterschätzenden Aufwand mit sich bringt. Dieser kann sich jedoch schnell lohnen, da ein Qualitätsmodell auch in allen Folgeprojekten, die ähnliche Strukturen aufweisen, eingesetzt werden kann. Um sich die Arbeit dennoch zu erleichtern, kann versucht werden, auf bereits existierende Qualitätsmodelle zurückzugreifen. Ein

Qualitätsmodell, das speziell auf die Erstellung wartbarer und flexibler serviceorientierter Architekturen ausgelegt ist und hierbei die Berücksichtigung bewährter Best-Practices prüft, kann in [3] eingesehen werden. Das beschriebene Qualitätsmodell basiert vollständig auf bestehenden Erkenntnissen aus Forschung und Praxis und formalisiert, welche Aspekte sich bei der Gestaltung von Services auf die Qualität der gesamten Architektur auswirken.

Nichtsdestotrotz muss auch in diesem Fall eine Abbildung der Qualitätsindikatoren auf konkrete Technologien erfolgen. Hierbei kann es sich sowohl um Modelle der Architektur, die zum Beispiel mithilfe der Unified Modeling Language (UML) und der Service oriented architecture Modeling Language (SoaML) erstellt wurden [7 und 8], als auch um konkrete Implementierungsartefakte handeln. Die Anwendung des Qualitätsmodells auf UML-Modelle mit SoaML als UML-Profil ist in [9, 10 und 11] aufgezeigt.

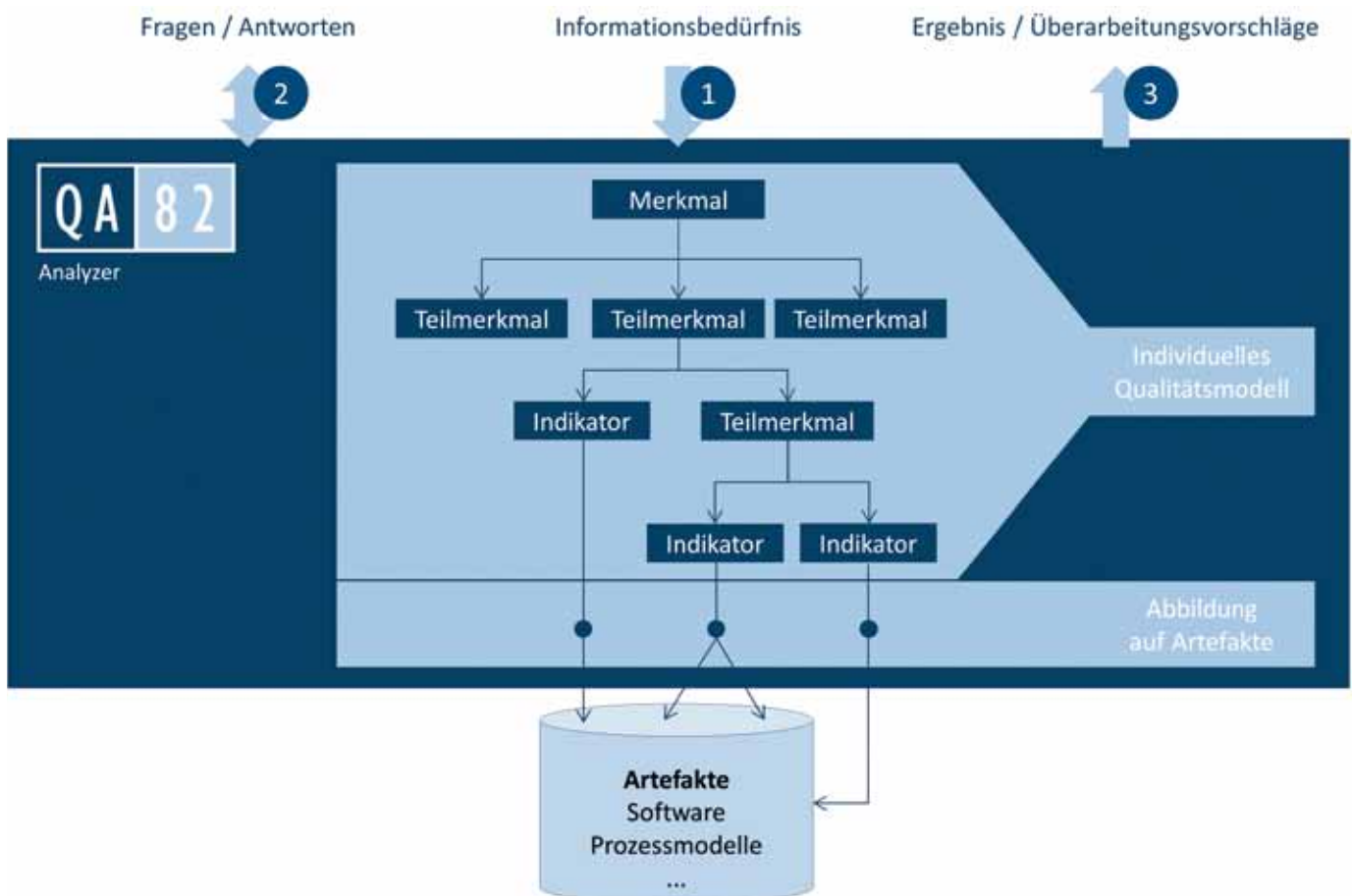


Abbildung 1: Aufbau eines Qualitätsmodells

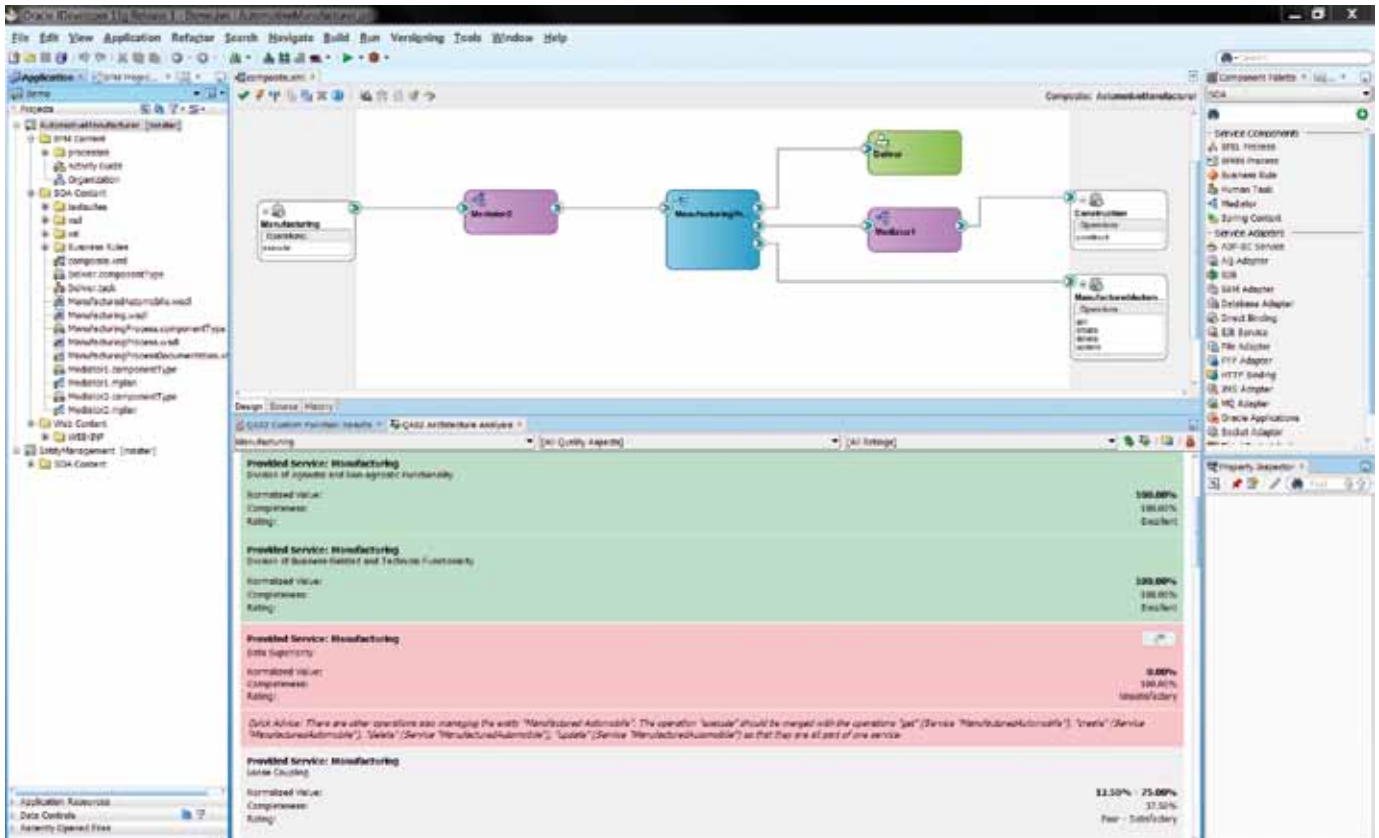


Abbildung 2: Hinweis auf eine nicht erfüllte funktionale Benennung im JDeveloper

Das folgende Beispiel veranschaulicht die Abbildung für Implementierungsartefakte der Oracle SOA Suite: Das Qualitätsmodell aus [3] beschreibt, dass die Auffindbarkeit als Qualitätsteilmerkmal durch die funktionale Benennung der angebotenen Operationen eines Service beeinflusst wird. Operationen sollten demnach wie in obigem Beispiel konkrete Begrifflichkeiten, die in der behandelten Domäne verbreitet sind, nutzen. Um diesen Qualitätsindikator zu prüfen, ist es erforderlich, die Operationen eines Service zu identifizieren. Dies kann je nach eingesetzten Technologien unterschiedlich erfolgen, sodass eine Abbildung des Qualitätsindikators durch eine konkrete Metrik auf eine bestimmte Technologie erforderlich ist. Im Falle der Nutzung der Service Component Architecture (SCA) müssen hierfür zunächst das SCA-Composite nach dem Service, anschließend die referenzierte WSDL-Datei und hierin die Operationen innerhalb des genutzten PortType identifiziert werden. Der Name ist dabei über das „name“-Attribut festgelegt.

Ein weiterer Qualitätsindikator betrachtet die Nutzung gemeinsamer Geschäftsentitäten durch die Operationen eines Service. So sollten diese nach Möglichkeit

dieselben Geschäftsentitäten oder deren Bestandteile nutzen [4]. Hierfür muss bei Nutzung der SCA erneut zunächst der Service innerhalb des SCA-Composite identifiziert und in der referenzierten WSDL-Datei der entsprechende PortType sowie die darin enthaltenen Operationen ermittelt werden. Anschließend müssen die hierbei genutzten Nachrichten und die darin enthaltenen Datentypen mit Geschäftsentitäten in Übereinstimmung gebracht und letztlich die Überlappung der genutzten Geschäftsentitäten der einzelnen Operationen geprüft werden. Dieser Qualitätsindikator wirkt sich dabei auf das Qualitätsteilmerkmal der eindeutigen Kategorisierung aus, die sich in bestehender Literatur häufig in Form von „Entity“- „Task“- und „Utility“-Services finden lässt [5 und 6]. Dieses Teilmerkmal ist mit der Kohäsion aus anderen Paradigmen vergleichbar.

Derartige Abbildungen müssen für alle Qualitätsindikatoren in einem Qualitätsmodell erfolgen. Um diesen Schritt zu vereinfachen, können auch bereits vollständig auf Technologie abgebildete Qualitätsmodelle genutzt werden. So existiert mit dem QA82 Analyzer für SOA und Webservices ein umgesetztes Qualitätsmodell, das be-

reits für die Oracle SOA Suite optimiert ist. Hierdurch ist es möglich, ein fundiert erarbeitetes Qualitätsmodell direkt für konkrete Technologien zu nutzen und somit ohne zusätzlichen Aufwand zu erkennen, welche Aspekte die Qualität der gesamten Architektur beeinflussen, sowie effiziente Qualitätsanalysen und gleichzeitig zielorientierte Überarbeitungen durchzuführen.

In ähnlicher Form können auch weitere Qualitätsmodelle erarbeitet und genutzt werden, die sich für den Einsatz in Integrationsprojekten eignen. So kann ein Qualitätsmodell für den Bereich „Business Process Management“ (BPM) dabei helfen, BPMN-Prozesse, die mit dem Oracle JDeveloper modelliert wurden, zu analysieren und dem Analysten konkrete Verbesserungsvorschläge zu präsentieren.

Auswertung von Qualitätsmodellen

Um auf Basis von Qualitätsmodellen eine Qualitätssicherung vorzunehmen, müssen die einzelnen Qualitätsindikatoren entweder manuell oder mithilfe der Metriken automatisiert ausgewertet werden. Eine manuelle Auswertung geht jedoch meist mit einem hohen Aufwand einher, da die einzelnen Indikatoren auf einer Vielzahl



Abbildung 3: Auswertung eines Qualitätsmodells für SOA im QA82 Analyzer

unterschiedlicher Software-Artefakte überprüft werden müssen. Dennoch liefert das Qualitätsmodell auch bei einer manuellen Analyse einen entscheidenden Vorteil: Es existiert hierdurch überhaupt eine Art „Kriterienkatalog“ und somit eine Checkliste darüber, welche Aspekte relevant und näher zu untersuchen sind. Auf diese Weise hilft das Qualitätsmodell dabei, die Qualitätssicherung zu strukturieren und klare Vorgaben zu schaffen.

Für eine automatisierte Auswertung ist eine entsprechende Werkzeugunterstützung erforderlich. Diese bringt Vorzüge wie beispielsweise einen Verlauf der Gesamtqualität über die Zeit sowie eine Verteilung der einzelnen Qualitäts(teil)merkmale und -indikatoren mit sich. Gleichzeitig kann das Werkzeug gegebenenfalls direkt Handlungsalternativen vorstellen, mit denen einzelne Qualitätsindikatoren und somit -merkmale optimiert werden können. Abbildung 2 veranschaulicht, wie mögliche Handlungsalternativen direkt innerhalb des JDeveloper dargestellt werden können. In Abbildung 3 hingegen ist dargestellt, wie der Verlauf der Qualität über die Zeit sowie entsprechende Verteilungen visualisiert werden können.

Fazit

Um die zunehmende Forderung nach einer wartbaren und flexiblen Lösung im Rahmen von Integrationsprojekten erfüllen zu können, ist eine Qualitätssicherung erforderlich, die speziell auf derartige nicht-funktionale Anforderungen ausgerichtet ist. Dieser Artikel hat gezeigt, wie der Einsatz von Qualitätsmodellen dabei helfen kann, eine Qualitätssicherung für diesen Zweck zu strukturieren. Da die Erstellung von Qualitätsmodellen einen hohen Aufwand erfordert, sollte – sofern möglich – auf bestehende Arbeiten zurückgegriffen werden. Im Falle regelmäßiger Qualitätssicherungen ist der Einsatz eines entsprechenden Werkzeugs zu empfehlen, um die Effizienz zu erhöhen und gleichzeitig die Fehlerquote zu reduzieren. Doch auch bei manuellen Qualitätssicherungen sollte auf die Erstellung eines Qualitätsmodells nicht verzichtet werden, da es alle Beteiligten als Kriterienkatalog und somit Leitfaden unterstützt.

Literatur

- [01] McCall, J. A.; Richards, P. K.; Walters, G. F.: Factors in Software Quality, Vols. 1-3. Rome Air Development Centre, Griffiss Air Force Base New York, 1977
- [02] Basili, V. R.; Weiss, D. M.: A Methodology for Collecting Valid Software Engineering Data.

IEEE Transaction on Software Engineering, SE-10, No. 6, pp. 728-738, 1984

- [03] Gebhart, M.: Qualitätsorientierter Entwurf von Anwendungsdiensten. KIT Scientific Publishing, Karlsruhe, 2011
- [04] Gebhart, M.; Abeck, S.: Metrics for Evaluating Service Designs based on SoaML. International Journal on Advances in Software, 4(1&2), pp. 61-75, 2011
- [05] Erl, T.: Principles of Service Design. Prentice Hall, 2007
- [06] Erl, T.: SOA Design Patterns. Prentice Hall, 2009
- [07] OMG: Service oriented architecture Modeling Language (SoaML), Version 1.0.1, 2012
- [08] Gebhart, M.: Service Identification and Specification with SoaML. In: Ionita, A. D.; Litoiu, M.; Lewis, G. (Hrsg.): Migrating Legacy Applications: Challenges in Service Oriented Architecture and Cloud Computing Environments, pp. 102-125, IGI Global, 2012
- [09] Gebhart, M.; Abeck, S.: Quality-Oriented Design of Services. International Journal on Advances in Software, 4(1&2), pp. 144-157, 2011
- [10] Gebhart, M.: Effiziente Qualitätsanalyse der Gestaltung serviceorientierter Architekturen auf Basis von SoaML. In: Schmietendorf, A.; Patzer, K. (Hrsg.): Berliner Schriften zu modernen Integrationsarchitekturen – 7. Workshop Bewertungsaspekte serviceorientierter Architekturen (BSOA) 2012, pp. 29-41, 2012
- [11] Gebhart, M.; Sejdovic, S.: Quality-Oriented Design of Software Services in Geographical Information Systems. International Journal on Advances in Software, 5(3&4), 293-307, 2012

Dr. Michael Gebhart
michael.gebhart@qa82.de