

# Softwarequalität erfassen und vergleichen: Von der Messung bis zum abstrakten Qualitätsattribut

*Qualität ist ein wichtiges Merkmal zur Differenzierung von Software. Aber wie kann diese überprüft werden? Wie kann die Qualität einer Software mit der einer anderen Software verglichen werden? Im Forschungsprojekt Quamoco wurde ein praktikables Vorgehen zur Messung und Beurteilung von Softwarequalität entwickelt. Dieser Artikel stellt die Ergebnisse des Forschungsprojekts und die entstanden, frei verfügbaren Werkzeuge und Modelle vor, mit denen Softwarequalität bewertet werden kann.*

Wie entscheiden Sie, wenn Sie eine neue Software anschaffen? Ein wichtiges Kriterium ist, dass sie die notwendige und im Idealfall sogar zusätzliche Funktionalität besitzt. Sie sollte einen angemessenen Preis haben und auch schnell lieferbar sein. Mehr und mehr verschiebt sich der Fokus aber weg von diesen Kriterien hin zu einem schwerer fassbaren und damit auch schwerer bewertbaren Konzept: der Qualität.

In der Funktionalität ähneln sich heutige Softwareprodukte sehr stark und nur selten können sie sich durch neue Features hervorheben. Der Anschaffungspreis rückt im Vergleich zu den Betriebs- und Wartungskosten oft in den Hintergrund. Denken Sie beispielsweise an den Web-Browser Ihrer Wahl. Haben Sie ihn danach ausgesucht, welche Features er anbietet, oder danach, wie viel er kostet? Für viele Nutzer sind bei dieser Entscheidung stattdessen die Performanz oder die Sicherheit entscheidende Aspekte. Aus Entwicklersicht sind zudem Qualitätseigenschaften – wie Wartbarkeit und Portierbarkeit – wichtig, um eine effiziente Weiterentwicklung der Software sicherstellen zu können.

Wie können wir dieses oft schwer zu greifende Konzept „Qualität“ bei Software erfassen? Wir müssen es überprüfbar und messbar machen. Dies ist keine einfache Aufgabe und bedarf einer Vielzahl von Messungen und Werkzeugen. Im Forschungsprojekt *Quamoco (Softwarequalität: Flexible Modellierung und Integriertes Controlling)* hat das Konsortium – bestehend aus den Forschungspartnern TU München und Fraunhofer IESE sowie den Industriepartnern Capgemini, itestra, SAP und Siemens – daran gearbeitet, eine Qualitätsbewertung von Software zu ermöglichen und zugänglich zu machen. Die Arbeit an Quamoco wurde durch das Bun-

desministerium für Bildung und Forschung gefördert (Förderkennzeichen: 01IS08023).

## Softwarequalität

Produktqualität ist schon an sich ein komplexes und vielschichtiges Konzept (vgl. [Gar84]). Da Software ein immaterielles Produkt ist, das sich nicht berühren und nur schwer vermessen lässt, ist der Begriff Softwarequalität schwer zu fassen. Dabei geht er deutlich über das klassische Zählen von Fehlern hinaus, da Fehler meist nur fehlende funktionale Korrektheit beschreiben.

Auf der einen Seite gibt es Normen wie die ältere ISO 9126 (vgl. [ISO01]) oder deren Nachfolger ISO 25010 (vgl. [ISO11]), die Softwarequalität in verschiedene Qualitätsattribute, wie Wartbarkeit oder Zuverlässigkeit, aufspalten. Die Einhaltung dieser Normen wird von Unternehmen erwartet und teilweise auch in Verträgen festgeschrieben. Sie verbleiben aber auf einem sehr abstrakten Niveau und helfen nur bedingt bei der Beantwortung der Frage, wie hoch die Softwarequalität tatsächlich ist. Ihre Wirkung ist aus den folgenden Gründen beschränkt:

- Die Normen sind sehr abstrakt und deshalb nicht unmittelbar operativ einsetzbar.
- Sie besitzen keine ausreichende Struktur, um konkrete Messungen von Qualitätsattributen zu ermöglichen.
- Die Normen stellen nicht klar dar, wie die enthaltenen Qualitätsattribute in Bezug zu Geschäfts-, Projekt- und Produktzielen stehen.
- Sie sind nicht flexibel genug, um an unterschiedlichste Bereiche der Softwareentwicklung und -verwendung angepasst werden zu können.

- Die Normen gehen nicht auf die Besonderheiten spezifischer Softwarekategorien, wie beispielsweise eingebettete Softwaresysteme, ein.

So ist es praktisch kaum möglich, die Einhaltung der Normen zu prüfen und somit eine definierte und vergleichbare Qualität der entwickelten Software nachzuweisen (vgl. [Wag10]).

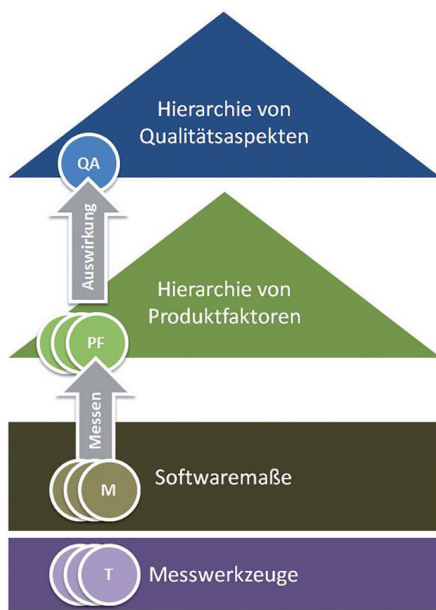
Auf der anderen Seite existiert eine Vielzahl von Werkzeugen und Maßen, die Konzepte vermessen, die potenziell einen Einfluss auf die Softwarequalität haben. Heute kommen in Softwareprojekten zudem immer häufiger Werkzeuge zur automatisierten statischen Codeanalyse zum Einsatz, wie „FindBugs“ (vgl. [Sou12]), „PMD“ (vgl. [Sou13]) oder „Gendarme“ (vgl. [Mon13]). Es ist jedoch nicht klar, welche Attribute der Softwarequalität durch diese Maße und Kennzahlen auf welche Art und Weise beeinflusst werden und wie sie zusammenspielen.

Diese Lücke zwischen abstrakten Qualitätsattributen und konkreten Maßen und Kennzahlen füllt Quamoco. Im Rahmen dieses Forschungsprojekts wurden Qualitätsmodelle erarbeitet, die beide Seiten verbinden und damit eine Bewertung der Qualität mit ganz konkreten Messungen und Aggregationen hin zu Qualitätsattributen erlauben. Besonders wichtig war uns, dass dies möglichst automatisiert durch Werkzeuge erfolgt und weitgehend anpassbar ist, sodass ein Einsatz in unterschiedlichsten Kontexten und auch in einem kontinuierlichen Qualitätscontrolling (vgl. [Dei11]) möglich ist.

## Flexible Modellierung

Die Bewertung von Qualität basiert auf einer klaren Vorstellung, was Qualität im Einzelfall bedeutet. Ein Qualitätsmodell

Vereinfachte Darstellung der Modellstruktur



Beispielhafter Auszug aus dem Basismodell

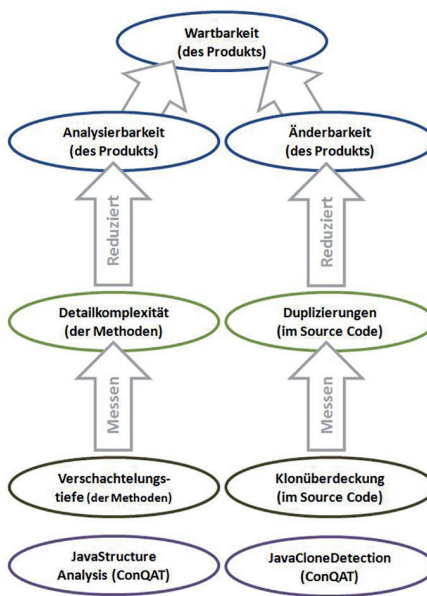


Abb. 1: Struktur des Quamoco-Metamodells mit einem Beispiel aus dem Basismodell.

hilft, diese Vorstellung festzuhalten und zu konkretisieren. Die bei Quamoco entwickelten Qualitätsmodelle enthalten konkrete Maße und sind für den operativen Einsatz ausreichend detailliert.

Jedoch hat Qualität nicht unbedingt für jeden und für jede Art von Software die gleiche Bedeutung. Eingebettete Software auf einem Automobil-Steuergerät stellt beispielsweise andere Anforderungen an bestimmte Aspekte der Softwarequalität als eine Web-Anwendung. Daher brauchen wir kein unveränderliches Qualitätsmodell als fixe Vorgabe für die Qualitätsbewertung, sondern eines, das flexibel um die Spezifika des gegebenen Anwendungskontexts ergänzt und an die eigenen Bedürfnisse angepasst werden kann. Der Quamoco-Ansatz trägt dieser Forderung in vielerlei Hinsicht Rechnung.

Zunächst sind Qualitätsmodelle, die im Rahmen von Quamoco entstanden sind, modular aufgebaut. Als Kern haben wir ein Basismodell entwickelt, das Qualität für Software im Allgemeinen beschreibt. Darüber hinaus gibt es weitere Modelle, die sich beispielsweise mit den Spezifika der objektorientierten Entwicklung oder der Messung von Qualität für bestimmte Programmiersprachen wie Java oder C# beschäftigen. Je nach Projektkontext und Domäne können diese Modelle für die eigene Qualitätsbewertung zusammengestellt beziehungsweise um weitere Teilmodelle ergänzt werden. So haben die Industriepartner domänen-

spezifische Erweiterungen für konkrete Anwendungsdomänen, wie Individualsoftware, SOA-Anwendungen und eingebettete Systeme, aufgebaut.

Des Weiteren können die Inhalte einzelner Module an die eigenen Anforderungen angepasst werden. Eine Anpassung darf aber nicht zu unvollständigen oder inkonsistenten Modellen und damit zu fehlerhaften Qualitätsbewertungsergebnissen führen.

Im Rahmen von Quamoco haben wir ein Metamodell erstellt, das einen detaillierten Bauplan für Qualitätsmodelle vorgibt. Es erlaubt zu beschreiben, wie in einem konkreten Modell die Aspekte der Softwarequalität (*Qualitätsfaktoren-Hierarchie*), beispielsweise aus der ISO 25010, von so genannten Produktfaktoren (*Produktfaktoren-Hierarchie*) beeinflusst werden. Diese Produktfaktoren beschreiben beobachtbare Eigenschaften eines Systems und werden mittels konkreter Maße (*Softwaremaße*) auf quantifizierbare Werte heruntergebrochen und mit Messwerkzeugen erhoben (siehe Abbildung 1).

So wird beispielsweise im Basismodell deutlich, dass Änderbarkeit ein Aspekt von Wartbarkeit ist. Die Änderbarkeit wiederum wird unter anderem durch den Produktfaktor „Code-Duplikate“ negativ beeinflusst. Codeduplikate können mit Werkzeugen zur statischen Codeanalyse erkannt werden und gehen mittels des Maßes

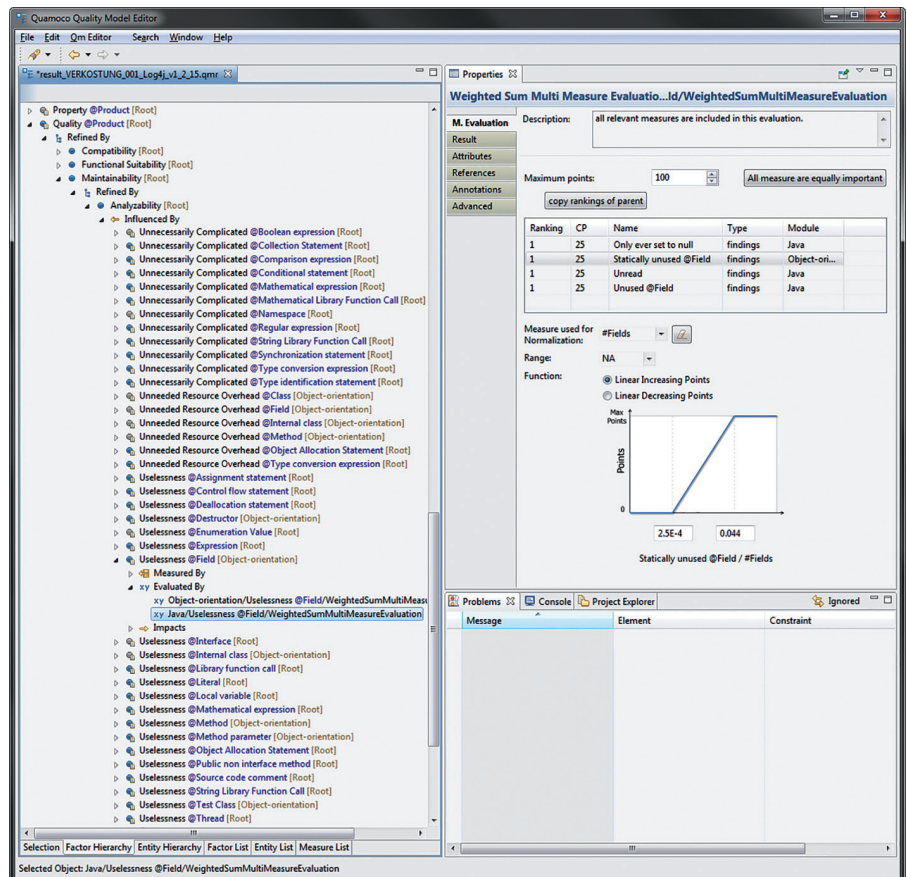


Abb. 2: Der Qualitätsmodell-Editor ermöglicht die Erstellung und Anpassung von Qualitätsmodellen sowie die Visualisierung von Bewertungsergebnissen.

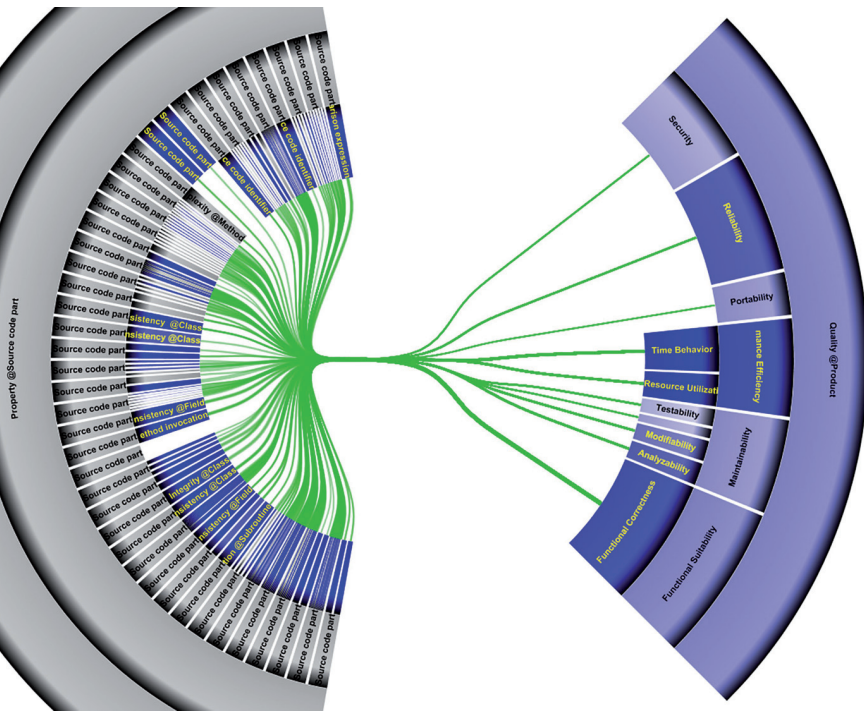


Abb. 3: Sunburst-Visualisierung der Modellvollständigkeit am Beispiel Java.

der so genannten Klonüberdeckung (*Clone Coverage*) in die Berechnung der Softwarequalität ein.

Damit beschreiben die konkreten Maße letztlich, was sich im Detail hinter den eher abstrakten Qualitätsattributen verbirgt. Zusätzlich erlaubt das Metamodell zu definieren, wie sich die tatsächlich gemessenen Werte zu einer Bewertung verarbeiten und entlang der Modellstruktur über mehrere Ebenen zu umfassenderen Qualitätsbewertungen aggregieren lassen.

Neben dem Metamodell, das Vorgaben zur Struktur der Qualitätsmodelle macht, existieren weitere im Projekt entwickelte Hilfen für den Qualitätsmodellierer:

- ein Qualitätsmodell-Editor
- ein Anpassungsprozess
- ein Anpassungsassistent

Abbildung 2 zeigt den frei verfügbaren Qualitätsmodell-Editor. Dabei ist links ein Ausschnitt aus dem Basismodell zu sehen, rechts eine Eingabemaske zur Festlegung der Bewertungsfunktion und darunter ein Bereich für Hinweise zu notwendigen Anpassungen.

Der Qualitätsmodell-Editor basiert auf dem *Eclipse Modeling Framework (EMF)*. Er ermöglicht die Bearbeitung von Qualitätsmodellen auf Grundlage des Quamoco-Metamodells. Dieser Editor wurde von den Projektpartnern verwendet, um das Basis-

modell und die domänenspezifischen Qualitätsmodelle zu erstellen.

Der in den Qualitätsmodell-Editor integrierte Anpassungsassistent führt den Benutzer zielorientiert durch die Anpassung und weist auf potenzielle Unvollständigkeiten und Inkonsistenzen im Modell hin (vgl. [Klä11]).

Neben der Werkzeugunterstützung für die Modellierung ist ein weiteres zentrales Ergebnis von Quamoco der Basis-Qualitätsstandard, auch Basismodell genannt. Für die Programmiersprachen Java und C# werden automatisierte Messungen und manuelle Bewertungen zu den Attributen der Norm ISO 25010 aggregiert. Dazu wurden in einer mehrstufigen Modellierung *top-down* die Qualitätsattribute auf mögliche Messungen heruntergebrochen und das entstandene Modell wurde anschließend *bottom-up* verfeinert.

In der Entwicklung des Basismodells haben sich sowohl Industrie- als auch Forschungspartner engagiert. In mehreren Iterationen wurde das Modell über den Projektzeitraum von drei Jahren sukzessiv aufgebaut und die Inhalte wurden vereinheitlicht. Das Modell umfasst derzeit 286 Faktoren, von denen 194 auf der Basis von insgesamt 526 Maßen vermessen werden. Die 194 Faktoren ergeben sich aus der Norm ISO 25010, ergänzt um eigene Faktoren. Die restlichen Faktoren dienen der Modellstrukturierung. Trotz des großen Aufwands in der Mo-

dellerstellung ist Softwarequalität ein so facettenreiches Konzept, dass das derzeitige Basismodell keinen Anspruch auf Vollständigkeit erheben kann. Um möglichst effizient analysieren zu können, haben wir uns auf automatische statische Analysen konzentriert und diese mit manuellen Analysen ergänzt. Damit sind manche Qualitätsattribute kaum oder nur zum Teil messbar. Beispielsweise finden zur Bestimmung der Performanz (*Performance Efficiency*) keine dynamischen Tests statt. Es werden aber potenziell problematische Anweisungen wie *sleep* mit einem Lock erkannt. **Abbildung 3** gibt mittels einer im Werkzeug bereitstehenden Visualisierung einen Überblick über den derzeitigen Stand einzelner Modellbereiche. In der Abbildung sind auf der linken Seite Produktfaktoren und auf der rechten Seite die ISO-Qualitätsattribute dargestellt. Je dunkler ein Bereich ist, umso vollständiger ist er ausmodelliert und durch Maße abgedeckt.

### Integriertes Controlling

Alle konstruktiven und analytischen Qualitätssicherungsmaßnahmen basieren auf dem an das jeweilige Projekt angepassten Qualitätsmodell. Das Modell wird nicht nur in der Entwicklung, sondern auch in der Wartung angewendet. So kann es dabei helfen, einem schleichenden Qualitätsverfall über den Software-Lebenszyklus hinweg entgegenzuwirken. Im Folgenden illustrieren wir eine solche Qualitätsbewertung am Beispiel des bekannten Open-Source-Projekts „Log4j“ in der Version 1.2.15 (vgl. [Apa12]).

Wie bereits in **Abbildung 1** dargestellt, findet die Bewertung der Messergebnisse auf drei Ebenen statt: von *konkreten Maßen (M)* über *Produktfaktoren (PF)* bis hin zu *Qualitätsattributen (QA)*.

- In die Bewertung gehen unterschiedlichste Maße ein, beispielsweise die Anzahl bestimmter Codeanomalien. In einem ersten Schritt werden diese Werte normalisiert, um eine Vergleichbarkeit zwischen unterschiedlichen Systemen herzustellen, beispielsweise als relativer Anteil des betroffenen Codes oder der Anzahl der Codeanomalien pro 1.000 Codezeilen. In der Bewertung von Log4j werden beispielsweise die zwei statisch ungenutzten Felder zu der Gesamtzahl von 415 Feldern in Bezug gesetzt. Im Anschluss bildet eine im Qualitätsmodell hinterlegte Bewertungsfunktion das normalisierte



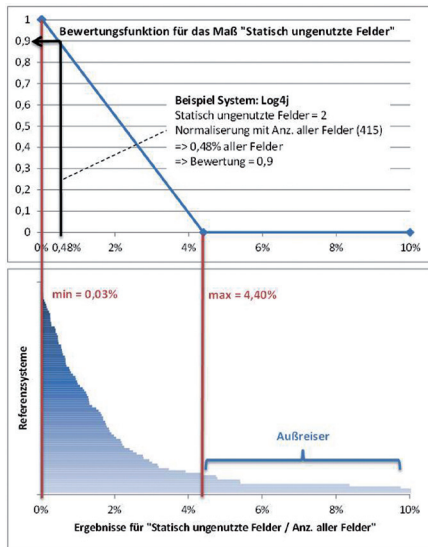


Abb. 4: Beispiel zur Abbildung eines Messergebnisses auf die Bewertungsskala [0,1].

Maß auf eine einheitliche Skala von 0 (schlechtester Wert) bis 1 (bestmöglicher Wert) ab (siehe **Abbildung 4**). Die Parameter der entsprechenden Bewertungsfunktion (minimaler und maximaler Schwellwert) sind vom Maß abhängig. Für die Java-Bestandteile des Modells wurden sie beispielsweise auf der Basis von mehr als 100 Open-Source-Produkten bestimmt. Hierbei wurden Ausreißer in den Messwerten identifiziert und Experten zur Plausibilitätsprüfung genutzt. In **Abbildung 4** wird dies beispielhaft für Log4j illustriert. Der Nutzer kann das vorgegebene Set von Schwellwerten aber auch für die eigene Domäne oder Umgebung anpassen.

- Produktfaktoren aggregieren mehrere bewertete Maße. Wie in **Abbildung 2** gezeigt, werden die Gewichte der verschiedenen Maße basierend auf einer festzulegenden Rangfolge oder so genannten *Contribution Points* berechnet. Für die Quamoco-Qualitätsmodelle wurden die Rangfolgen gemeinsam durch Experten aus Industrie und Forschung bestimmt. So trägt das Ergebnis der statisch ungenutzten Felder aus **Abbildung 4** mit 25 Prozent zur Bewertung des Produktfaktors „Uselessness @Field“ bei.
- Qualitätsattribute aggregieren Produktfaktoren ebenfalls über Gewichte. So trägt der Faktor „Uselessness @Field“ zu 0,23 Prozent der Bewertung des Attributs „Analyzability“ bei. Die (durchschnittliche) Wichtigkeit der einzelnen

Qualitätsattribute berechnet sich aus Ergebnissen einer Umfrage mit mehr als 100 Teilnehmern (vgl. [Wag12-a]).

Zur Automatisierung der Bewertung wurde „ConQAT“ (vgl. [Con13]), ein Werkzeug zur Durchführung von Softwaremessungen, um die Möglichkeiten zur Bewertung und Aggregation der Messergebnisse erweitert.

Zusammen mit dem bereits vorgestellten Qualitätsmodell-Editor steht somit eine komplette Werkzeugkette für die Anwendung von Quamoco zur Verfügung: Von der Erstellung eines Qualitätsmodells über seine projektspezifische Anpassung, bis hin zur automatisierten Vermessung und Bewertung der Software. Dies ist notwendig, um eine fortlaufende Ermittlung der Softwarequalität zu ermöglichen.

Die Ergebnisse der Bewertung können im Qualitätsmodell-Editor auch grafisch aufbereitet werden. Eine der Visualisierungsoptionen, die einen Überblick über die Gesamtbewertung von Log4j gibt, zeigt **Abbildung 5**.

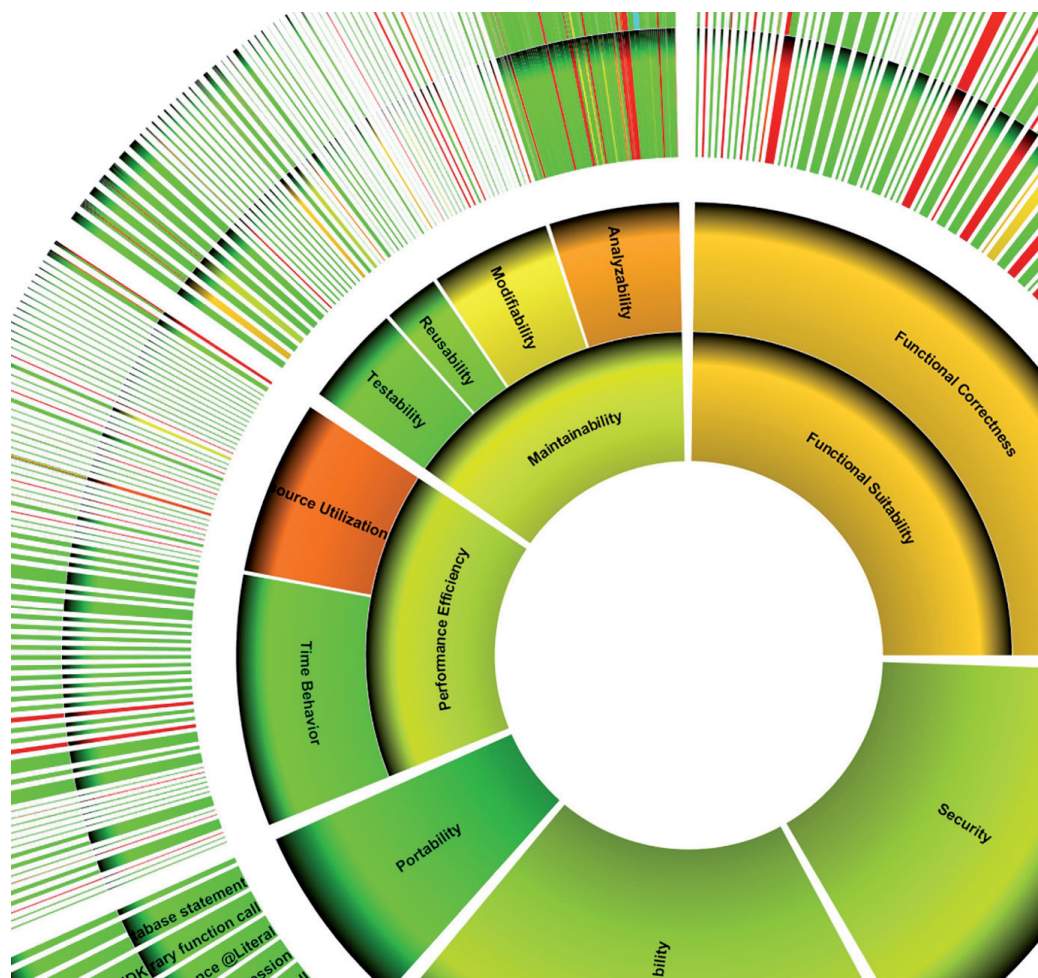
In den inneren beiden Ringen sind die Bewertungen der Qualitätsattribute dargestellt. Davon abgesetzt folgen die Produktfaktoren und im äußersten Ring die

einzelnen Maße. Die Bewertungsergebnisse (zwischen 0 und 1) werden für die Visualisierung in deutsche Schulnoten umgerechnet (1: grün, 6: rot). Die Breite der einzelnen Ringsegmente entspricht jeweils dem Beitrag zum Gesamtergebnis (Gewicht). Für Detailanalysen der Bewertungen ist es möglich, in die Visualisierung im Qualitätsmodell-Editor hinein zu zoomen (*drill-down*). Dabei werden ausgewählte Qualitätsattribute und Produktfaktoren vergrößert und somit die darin aggregierten Produktfaktoren und Maße erkennbar. **Abbildung 6** zeigt beispielhaft den Beitrag der anhand des Maßes „Statically unused @Field“ bestimmten Detailbewertung zur Gesamtbewertung.

### Erste Erkenntnisse

Wenn man sich das Basismodell anschaut, fällt auf, dass der Aspekt Wartbarkeit durch viele Messungen gestützt wird, während andere Aspekte der ISO 25010, wie zum Beispiel Sicherheit (*Security*), durch deutlich weniger Messungen gestützt werden. Das liegt an den Möglichkeiten der verwendeten Werkzeuge zur statischen Codeanalyse für die Ermittlung der konkreten Maße. Für Java-Programme kommen bekannte Werkzeuge wie FindBugs ([Sou12])

Abb. 5: Die Sunburst-Visualisierung der Bewertungsergebnisse ermöglicht einen schnellen Überblick über die Systemqualität.



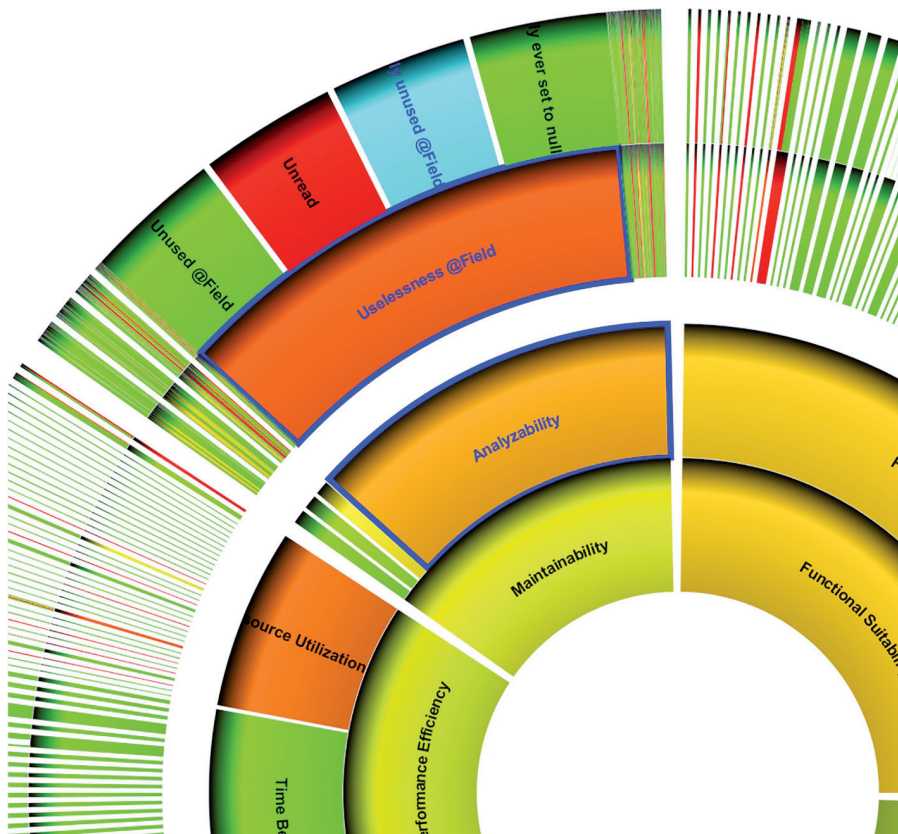


Abb. 6: Detailansicht in der Sunburst-Visualisierung der Bewertungsergebnisse.

oder PMD ([Sou13]) zum Einsatz. Für C# sind es vergleichbare Werkzeuge, wie z.B. „Gendarme“ (vgl. [Mon13]). Diese Werkzeuge erkennen potenziell problematische Stellen im Code, wie beispielsweise leere *catch*-Blöcke. Daher beziehen sich die Messungen hauptsächlich auf die Codequalität, die den stärksten Bezug zur Wartbarkeit besitzt. Für den Aspekt „Sicherheit“ kann beispielsweise die Kapselungsstärke (*Encapsulation Strength*) statisch gemessen

werden, wohingegen dynamische Messungen zur Authentizität (*Authenticity*) nicht automatisiert möglich sind.

Das zeigt, wo es heute bereits gute automatisierte Unterstützung bei der Bestimmung von Softwarequalität gibt. Noch wichtiger aber: Es zeigt auch, worauf Softwareingenieure in Zukunft verstärkt achten sollten, um einen umfassenden Eindruck von der Qualität einer jeweiligen Software zu bekommen. Im Rahmen des Forschungsprojekts wur-

den die erstellten Modelle evaluiert, um die Zuverlässigkeit der Bewertung zu prüfen. Diese Evaluation erfolgte mit proprietären Softwareprodukten und Open-Source-Projekten. Unabhängig von der Bewertung durch Quamoco wurden fünf Open-Source-Produkte, darunter auch Log4j, von neun Experten geprüft und bezüglich ihrer Softwarequalität bewertet. Wie **Tabelle 1** zeigt, korrelieren die Bewertungen von Quamoco und die gemittelten Experteneinschätzungen sehr gut.

Ein weiterer Aspekt der Evaluierung war die Entwicklung der Bewertungsergebnisse über verschiedene Entwicklungsstufen eines Produkts. Für ein proprietäres Softwareprodukt konnte die steigende Softwarequalität dargestellt werden. Während des Projekts selbst fand keine Qualitätsmessung statt und Verbesserungsmaßnahmen stützten sich rein auf das Qualitätsverständnis der Entwickler. Nachträglich wurden verschiedene Releases dieses Softwareprodukts mit Quamoco vermessen und die erwartete Verbesserung der Softwarequalität spiegelt sich in besser werden den Bewertungen wider.

Mehr Details zur empirischen Untersuchung unserer Modelle finden Sie in einer anderen Publikation zu Quamoco (vgl. [Wag12-b]). Insbesondere für das Basismodell zeigt sich eine hohe Treffsicherheit in der Bewertung von Softwarequalität.

### Fazit

Auch die Quamoco-Resultate werden nicht dazu führen, dass Sie sich um Softwarequalität nicht weiter kümmern müssen. Quamoco kann Ihnen aber helfen, ein umfassendes Qualitätsmodell auf Ihre Bedürfnisse hin anzupassen und dann kon-

## Die Autoren



|| Christian Schubert  
(christian.schubert@cappgemini.com)  
ist als Technical Architect bei Cappgemini in Stuttgart tätig. Seine Arbeitsschwerpunkte sind Testing und Software-Controlling in Projekten für Banken, Versicherungen und die Automobilindustrie.



|| Michael Kläs  
(michael.klaes@iese.fraunhofer.de)  
arbeitet als Wissenschaftler am Fraunhofer Institut für Experimentelles Software Engineering in Kaiserslautern und koordinierte die Arbeiten im Quamoco-Projekt seitens Fraunhofer.



|| Prof. Dr. Stefan Wagner  
(stefan.wagner@informatik.uni-stuttgart.de)  
leitet den Lehrstuhl für Software Engineering an der Universität Stuttgart. Er war Projektleiter von Quamoco und er führt mit seiner Forschungsgruppe die Arbeiten aus Quamoco fort.








## Literatur & Links

- [Apa12] Apache Software Foundation, Logging Services, siehe: <http://logging.apache.org/log4j/1.2/>, 2012
- [Con13] ConQAT, siehe: [www.conqat.de](http://www.conqat.de), 2013
- [Dei11] F. Deißeböck, B. Hummel, Kontinuierliches Qualitäts-Controlling: Mittel gegen den Qualitätsverfall in der Softwarewartung, in: OBJEKTSpektrum 05/2011
- Gar84] D. Garvin, What Does „Product Quality” Really Mean?, Sloan Management Review, 1984
- [ISO01] ISO/IEC 9126:2001 Software engineering – Product quality, 2001
- [ISO11] ISO/IEC 25010:2011 Systems and software engineering – Systems and software Quality Requirements and Evaluation (SQuaRE) – System and software quality models, 2011
- [Klä11] M. Kläs, C. Lampasona, J. Münch, Adapting Software Quality Models: Practical Challenges, Approach, and First Empirical Results, in: Proc. of 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2011), IEEE Computer Society, 2011
- [Mon13] Mono, Gendarme, siehe: [www.mono-project.com/Gendarme](http://www.mono-project.com/Gendarme), 2013
- [Sou12] Sourceforge.net, FindBugs – Find Bugs in Java Programs, siehe: [findbugs.sourceforge.net](http://findbugs.sourceforge.net), 2012
- [Sou13] Sourceforge.net, PMD, siehe: [pmd.sourceforge.net](http://pmd.sourceforge.net), 2013
- [Wag10] S. Wagner, M. Broy, F. Deißeböck, M. Kläs, P. Liggesmeyer, J. Münch, J. Streit, Softwarequalitätsmodelle. Praxisempfehlungen und Forschungsagenda, Informatik Spektrum, Springer, vo. 33, no. 1, pp. 37-44, 2010
- [Wag12-a] S. Wagner, K. Lochmann, S. Winter, A. Göb, M. Kläs, S. Nunnenmacher, Software Quality Models in Practice. Survey Results, Technischer Bericht TUM-I129, TU München, 2012
- [Wag12-b] S. Wagner, K. Lochmann, L. Heinemann, M. Kläs, A. Trendowicz, R. Plösch, A. Seidl, A. Göb, J. Streit, The Quamoco Product Quality Modelling and Assessment Approach, in: Proc. of 34<sup>th</sup> International Conference on Software Engineering (ICSE 2012), IEEE Computer Society, 2012

krete Bewertungen durchzuführen. Die abstrakten Qualitätsattribute aus der aktuellen ISO-Norm 25010 haben eine konkrete Grundlage bekommen, deren Verwendung einen hohen Mehrwert liefern kann.

Um die breite Verwendung zu ermöglichen, hat das Konsortium sowohl das Basismodell als auch die Werkzeuge frei unter der Apache-Lizenz zur Verfügung gestellt. Sie können diese beispielsweise über die Webseite [www.quamoco.de](http://www.quamoco.de) beziehen, auf der auch Informationen und Publikationen zum Thema Softwarequalität zu finden sind.

Produkt	Quellcode Zeilen	Quamoco Note	Rang bzgl. Quamoco	Rang bzgl. Experten
Checkstyle (4.4)	57 213	1 (1.87)	1 	1
RSSOwl (1.2.4)	82 258	3 (3.14)	2/3 	3
<b>Log4j (1.2.15)</b>	<b>30 676</b>	<b>3 (3.36)</b>	<b>2/3</b> 	<b>2</b>
TV-Browser (2.2.5)	125 874	4 (4.02)	4 	4
JabRef (2.3.1)	96 749	5 (5.47)	5 	5

*Tabelle 1: Vergleich der Qualitätsbewertung durch Quamoco und Experten.*

Wir arbeiten weiter am Ausbau der Werkzeugunterstützung und des Basismodells und werden dieses auch in einen entsprechenden Normierungsprozess einbringen,

damit Zertifizierungen möglich werden. Aber bereits jetzt kann Quamoco Sie dabei unterstützen, die Qualität für Java- und C#-Systeme objektiv nachzuweisen. ||