

DAS CAST-AIP-QUALITÄTSMODELL



Dipl. Inf. Michael Kläs
Dr. Ing. Adam Trendowicz
Dr. rer. nat. Jens Heidrich
Dipl. Inf. Axel Wickenkamp

Das Fraunhofer IESE ist ein Institut der Fraunhofer-Gesellschaft.

Das Institut transferiert innovative Software-Entwicklungstechniken, -Methoden und -Werkzeuge in die industrielle Praxis. Es hilft Unternehmen, bedarfsgerechte Software-Kompetenzen aufzubauen und eine wettbewerbsfähige Marktposition zu erlangen.

Das Fraunhofer IESE steht unter der Leitung von

Prof. Dr.-Ing. Peter Liggesmeyer
(Geschäftsführender Institutsleiter)

Prof. Dr. Dieter Rombach
(Institutsleiter Business Development)

Fraunhofer-Platz 1
67663 Kaiserslautern

Inhaltsverzeichnis

1	Überblick	1
2	Softwarequalität und Qualitätsmodelle	2
3	CAST AIP – das Qualitätsmodell im Überblick	4
4	Das IESE-Bewertungsverfahren	6
5	Unsere Ergebnisse	8
6	Fazit	13
	Weiterführende Literatur	15
	Mögliche Interessenkonflikte	17
	Über Fraunhofer IESE	17

1 Überblick

Statische Codeanalysen sind wichtiger Bestandteil einer effizienten Qualitätssicherung, um schon frühzeitig im Entwicklungszyklus Qualitätsrisiken und später auftretende Kosten, die so genannten „technischen Schulden“/ „technical debt“, zu minimieren. Mit der Application Intelligence Platform (AIP) bietet CAST eine weltweit eingesetzte und umfassende Lösung in diesem Segment an. Der vorliegende Bericht untersucht das Herzstück der AIP – nämlich das zur Bewertung der Codequalität verwendete Qualitätsmodell – im Hinblick auf den derzeitigen Stand der Praxis und Wissenschaft. Mit einem Qualitätsmodell wird beschrieben, welche Daten über den Code erhoben und wie diese zu einer Qualitätsaussage verdichtet werden. Der Bericht motiviert den Einsatz von Qualitätsmodellen (QMs) in der Softwareentwicklung, charakterisiert das CAST-AIP-QM, gibt einen Überblick über das in der Studie eingesetzte Untersuchungsverfahren und schlüsselt dessen Ergebnisse entlang von sieben Bewertungsdimensionen auf. Abschließend werden die wichtigsten Stärken sowie Schwächen zusammengefasst.

Die in diesem Dokument vorgestellte und durch das Fraunhofer IESE durchgeführte Evaluation des Qualitätsmodells ist Bestandteil einer umfassenderen Studie zur CAST AIP, die auch wirtschaftliche Aspekte betrachtet und bei Interesse über die Webseiten von CAST zu beziehen ist:

<http://www.castsoftware.com/worldwide/germany/CAST-Fraunhofer-IESE-PwC-Whitepaper?gad=fhr>

2 Softwarequalität und Qualitätsmodelle

Die Qualität einer Software wird immer mehr zum entscheidenden Kriterium, wenn es darum geht, die eigenen Geschäftsprozesse effizient zu unterstützen oder innovative Lösungen am Markt anzubieten.

Qualität als solche ist jedoch ein hochgradig abstraktes Konzept und damit schwer zu definieren und noch schwerer zu quantifizieren. Dies macht sowohl den Vergleich von Softwarelösungen hinsichtlich ihrer Qualität – beispielsweise bei Zukäufen – als auch das effektive Qualitätsmanagement eigener Entwicklungsprojekte zu einer Herausforderung [11].

In den letzten Jahrzehnten sind daher zahlreiche QMs entstanden, die aus unterschiedlichen Richtungen einen Blick auf Softwarequalität im Allgemeinen, aber auch auf spezifische Teilaspekte wie Wartbarkeit, Benutzbarkeit oder Zuverlässigkeit werfen. Hierbei kann grob zwischen primär wissenschaftlichen Entwicklungen, in der Praxis eingesetzten und erprobten Modellen und durch (internationale) Gremien entwickelte Normen unterschieden werden (vgl. Abb. 1).

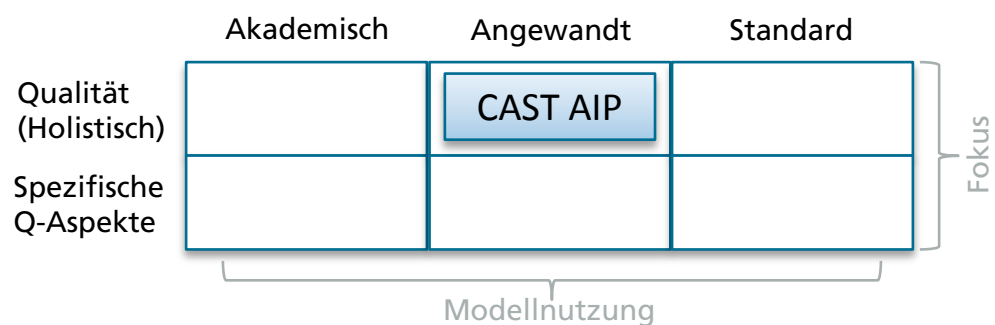


Abb. 1: Einordnung des CAST-AIP-Qualitätsmodells¹ (vgl. [12])

Im Gegensatz zu funktionalen Unzulänglichkeiten der Software werden Qualitätsdefizite durch klassische Testaktivitäten, wenn überhaupt, erst gegen Ende des Entwicklungsprozesses aufgedeckt. Zu diesem Zeitpunkt ist ihre Behebung jedoch häufig aufwändig und führt zu Überschreitungen des Entwicklungsbudgets und nicht einzuhaltenden Lieferterminen.

¹ CAST AIP ist eine kommerzielle Lösung, die für Unternehmen mit komplexen Applikationslandschaften konzipiert wurde.

Es macht daher Sinn, nicht nur die fertige Software zu prüfen, sondern auch Artefakte, die während des Entwicklungsprozesses entstehen. Im Prinzip ist so die Prüfung verschiedenster Dokumente von den Anforderungen über das Design bis hin zur Testfallspezifikation denkbar [2]. Im Bereich der automatischen Analysen ist derzeit jedoch ausschließlich die Prüfung des Quellcodes Stand der Praxis, da dieser das am häufigsten anzutreffende und am stärksten strukturierte Artefakt darstellt.

Nur in diesem Bereich stehen daher neben wissenschaftlichen Prototypen auch zahlreiche kommerzielle Werkzeuge zur Verfügung, um die Qualität mithilfe eines unterliegenden QM zu beleuchten.

Dabei wird der Quellcode im Gegensatz zum klassischen Testen hinsichtlich gewisser Eigenschaften statisch analysiert, ohne ausgeführt zu werden. Zum Beispiel lässt sich so die Komplexität einzelner Funktionen oder Klassen bestimmen, aber auch, ob bestimmte fehlerträchtige Strukturen im Code vorhanden sind oder gewisse Arten von Sicherheitslücken bestehen. Die Analyseergebnisse lassen sich dann anhand zuvor definierter Kriterien bewerten und zu einer Gesamtbeurteilung des analysierten Systems zusammenfassen.

Diese Informationen helfen dem Management dabei, Projekte und Verbesserungsinitiativen zu überwachen, einer Qualitätsdegeneration entgegenzuwirken oder die Qualität innerhalb extern vergebener Softwareentwicklungsaufträge effektiver zu steuern. Entwicklern helfen die Detailergebnisse, Qualitätsdefizite effizienter zu beseitigen und systematische Fehler früher zu erkennen.

Statische Codeanalyse kann dabei weitestgehend unabhängig vom Entwicklungsprozess eingesetzt werden. Auch bei agiler Entwicklung mit einer hohen Änderungsrate und vielen Iterationen helfen statische Analyseverfahren dabei, allgemeine Richtlinien der Organisation automatisiert zu prüfen und – sofern definiert – in Bezug auf die Einhaltung einer Referenzarchitektur zu testen.

3 CAST AIP – das Qualitätsmodell im Überblick

Der weitere Bericht beschäftigt sich mit dem durch CAST in seiner Werkzeuglösung integrierten Qualitätsmodell [1]. Die angebotene Lösung richtet sich primär an das IT-Management sowie an Applikationsverantwortliche aus Firmen, die Software im Bereich „Business Information Systems“ entwickeln oder deren Entwicklung beauftragen. Dort kann sie zur Verlaufskontrolle und Steuerung des Softwareportfolios eingesetzt werden, aber auch für Einzelbewertungen von Softwarelösungen (inklusive eigener Individualisierungen von Standardsoftware wie zum Beispiel SAP).

Das Modell vermisst hierzu den vorhandenen Software Quellcode, nutzt teilweise aber darüber hinausgehende Informationsquellen, wie eine vorhandene Referenzarchitektur oder Datenbankschemata. Bei der Bewertung beschränkt sich das Modell nicht auf ausgewählte Qualitätsaspekte, sondern bietet durch einen schicht- und technologieübergreifenden Ansatz eine generelle Beurteilung der Codequalität. Diese wird entlang der Business Criteria „Performance“, „Robustness“, „Security“, „Changeability“ und „Transferability“ aus dem CISQ-Modell [6] dargestellt. Zusätzlich wird eine zusammenfassende technische Bewertung angeboten, der so genannte Technical Quality Index (TQI).

Im Modell ist jedem Business Criterion eine Reihe von *Technical Criteria* zugeordnet, die selbst wiederum in eine Reihe von konkreten *Metriken und Regeln* verfeinert werden (vgl. Abb. 2). Beispielsweise trägt der Technical Factor „Secure Coding – Input“ zur Beurteilung des Business Criterion „Security“ bei und wird selbst unter anderem durch die Einhaltung der Regel „Always validate user input with Request variables“ geprüft.

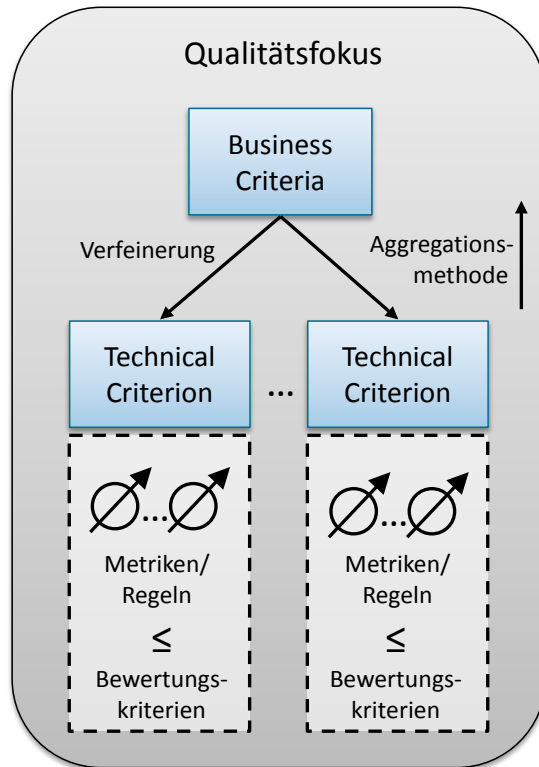


Abb. 2: Qualitätsmodell

4 Das IESE-Bewertungsverfahren

Die Beurteilung von CAST AIP erfolgte in der Studie nicht in einer direkten Gegenüberstellung mit Mitbewerbern, sondern als Einzeluntersuchung im Vergleich zum allgemeinen Stand der Praxis im Bereich der statischen Codeanalyse und zu aktuellen Forschungsergebnissen bezüglich Softwarequalität. Für die Beurteilung wird eine 5-stufige Skala von -2 bis $+2$ genutzt. Dabei repräsentiert 0 als Baseline den Stand der derzeit in der Praxis typischerweise anzutreffenden Lösungen; Werte kleiner als 0 zeigen eine negative Abweichung von diesem Stand der Praxis (das heißt, weitergehende Limitierungen oder Probleme) und Werte größer als 0 eine positive Abweichung in Richtung des (im Sinne des aktuellen Forschungsstands) derzeit Wünschenswerten.

Die Beurteilung strukturiert sich anhand von sieben Leitfragen, die sich jeder Anwender vor der Einführung eines Qualitätsmodells stellen sollte:

- A. Sind die Modellinhalte und -konzepte für die vorgesehene Anwendung ausreichend?
- B. Ist die Verfeinerung im Modell bis hin zu den Metrik- und Regeldefinitionen plausibel?
- C. Ist der Rückweg von den Einzelmessungen bis hin zur Gesamtbewertung plausibel?
- D. Gibt es eine geeignete und nachvollziehbare Darstellung der Bewertung?
- E. Wie robust ist die Bewertung hinsichtlich typischer Störfaktoren?
- F. Wie anwendbar ist das Modell in der Praxis?
- G. Kann ich das Modell an meine konkreten Bedürfnisse anpassen?

Diese Leitfragen dienen als Bewertungsdimensionen (Abb. 3), welche durch insgesamt 47 Kriterien weiter verfeinert werden. Die Kriterien basieren dabei auf bestehenden wissenschaftlichen Arbeiten und dem umfangreichen praktischen Erfahrungsfundus des Autorenteam.

Die Bewertung selbst erfolgte anschließend anhand der einzelnen Kriterien expertenbasiert durch die Autorengruppe nach der Sichtung vorhandener Dokumente und Werkzeuge sowie der Durchführung mehrerer Gespräche mit CAST-Mitarbeitern.

Die abschließende Gesamtbewertung stellt ein Profil über die Bewertungsdimensionen dar, in dem sich das Bewertungsergebnis der einzelnen Dimension aus dem Mittel der Beurteilungen der jeweils zugrunde liegenden detaillierten Kriterien ergibt.

5 Unsere Ergebnisse

Abb. 3 fasst die Bewertungsergebnisse entlang der sieben Leitfragen zusammen. In den folgenden Abschnitten diskutieren wir für jede Dimension die konkreten Aspekte, die wir bei der Bewertung berücksichtigt haben, und stellen die wichtigsten Ergebnisse vor. Dabei fassen wir jeweils die Aspekte zusammen, die dem derzeitigen Stand der Praxis entsprechen (0), eine negative (-) oder eine positive Abweichung (+) darstellen.

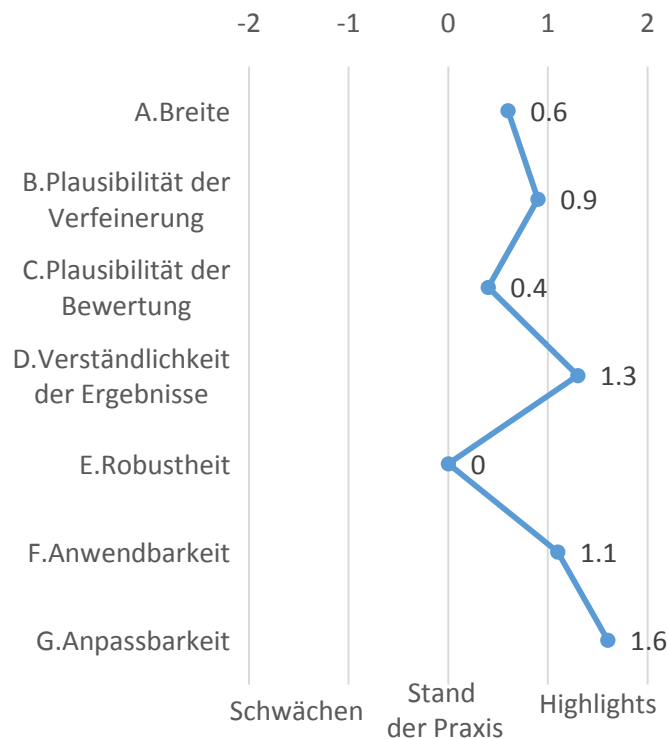


Abb. 3: Bewertungsprofil

A. Breite beschäftigt sich mit dem Umfang, den das Qualitätsmodell abdeckt. Wichtige Aspekte hierbei sind die Artefakte, Programmierkonzepte und Sprachen, die mit dem QM bewertet werden könnten, aber auch Qualitätsaspekte und Metriken, die das QM umfasst. Darüber hinaus haben wir geprüft, welche Anwendungszwecke (zum Beispiel Qualitätsüberwachung oder -vorhersage) das QM konzeptionell unterstützt, welche relevanten Standards dabei berücksichtigt sind und welche Sichten (bzw. Rollen und Interessengruppen) adressiert werden.

(+) CAST AIP umfasst viele verbreitete Programmierkonzepte und mehr als 30 Sprachen. Auch deckt es die meisten gängigen Qualitätsaspekte ab, die durch circa 1.000 Regeln und Metriken vermessen werden. Dies geht weit über den Stand der Praxis hinaus, der derzeit noch stark durch sprachspezifische Insellösungen geprägt ist, die nicht wie bei AIP einen schicht- und technologieübergreifenden Ansatz bieten.

(0) CAST AIP fokussiert dabei auf Softwarecode, inklusive Framework-Konfigurationen, lässt aber Softwareartefakte wie die Anforderungsspezifikation bei Qualitätsanalysen außen vor, was aber dem derzeitigen Stand der Praxis entspricht. Das Qualitätsmetamodell beinhaltet alle Konstrukte, die für die versprochenen Anwendungszwecke notwendig sind, jedoch keine für weitergehende Anwendungszwecke wie Qualitätsvorhersage.

(-) Das QM orientiert sich am Vorschlag der CISQ-Initiative [6], der sich derzeit in der Standardisierung durch OMG befindet. Das QM nennt dabei Qualitätsaspekte ähnlich wie im ISO/IEC 25000 Standard [3], allerdings nutzt es abweichende Definitionen und Hierarchien, was bei Nutzern der ISO/IEC-Norm zu Verwirrung führen kann.

B. Plausibilität der Verfeinerung beschäftigt sich mit der Glaubwürdigkeit der QM-Inhalte. Dabei haben wir berücksichtigt, wie gut einzelne Business Criteria, Technical Criteria und Metriken (bzw. Regeln) beschrieben werden und wie plausibel die Begründungen für bestimmte QM-Inhalte sind. Darüber hinaus haben wir die Vergleichbarkeit der mithilfe des QM erhobenen Messwerte, zum Beispiel über Software verschiedener Größe, berücksichtigt.

(+) Die Metriken/Regeln im QM sind detailliert beschrieben und ihre Anwendung wird klar begründet (für circa 2/3 der Regeln und Metriken werden zusätzlich externe Referenzen angegeben), womit sich das Modell positiv vom Stand der Praxis abhebt. Um Vergleichbarkeit der Messergebnisse zu gewährleisten, werden die entdeckten Regelverstöße hinsichtlich der Anzahl möglicher Verletzungen normalisiert, was einen sinnvolleren Ansatz darstellt als die praxisübliche Normalisierung über die Größe der Software.

(0) Technical Criteria sind im QM nur knapp beschrieben; darüber hinaus ist die Abbildung zwischen Technical und Business Criteria nicht explizit begründet. Auch sind die Business Criteria teilweise nicht oder nur sehr knapp beschrieben. Damit befindet sich die CAST-Lösung auf gleicher Höhe mit dem Stand der Praxis.

C. Plausibilität der Bewertung beschäftigt sich mit der Glaubwürdigkeit der Qualitätsbewertung auf Basis der Struktur des QM und den Bewertungsvorschriften. Hierbei haben wir untersucht, wie individuelle Regelverletzungen bestimmt werden, wie Grenzwerte und Gewichte zu den einzelnen Qualitätsas-

pekten im QM assoziiert werden und inwiefern sich das ausgewählte Aggregationsverfahren unter Berücksichtigung der QM-Struktur eignet. Des Weiteren haben wir das Vorhandensein relevanter Benchmarking-Daten und Evidenzen (zum Beispiel empirische Studien) überprüft.

(+) Das QM erlaubt das Ausschließen nicht relevanter Regelverletzungen (so genannter False Positives) aus der aktuellen und nachfolgenden Analyse, was bei vielen derzeit im Einsatz befindlichen Lösungen nicht oder nur umständlich möglich ist. Die Aggregationsmethode unterstützt die gegenseitige Kompensation positiver und negativer Bewertungen ebenso wie den Fall, dass ein kritisches Ergebnis nicht kompensiert werden darf („Veto“-Element). Dies geht deutlich über die zumeist anzutreffenden rein summativen Aggregationsverfahren hinaus. Vorhandene Benchmarking-Daten sind mit Größe und Hauptsprache der Applikation sowie Marktsegment attribuiert, was eine erste, wenn auch grobgranulare Einschränkung auf vergleichbare Benchmark-Systeme ermöglicht.

(0) Die vom QM vorgegebenen, erfahrungsbasierten Grenzwerte (kritische versus unkritische Regeln) sind nachvollziehbar, jedoch wird die Zuordnung von bestimmten Grenzwerten zu einzelnen Regeln nicht begründet. Außerdem sind der Prozess und die Methodik, mit denen die CAST-Experten einzelne Grenzwerte, Einstufungen und Gewichte bestimmt haben, nicht transparent. Leider stellt dies aber immer noch ein typisches Vorgehen in der Praxis dar. Der Zusammenhang zwischen gemessener und erlebter Qualität wird durch singuläre industrielle Erfahrungsberichte untermauert; aufwändigere wissenschaftliche Studien fehlen jedoch, was allerdings ebenfalls dem Stand der Praxis entspricht.

(-) Bewertungen werden mittels gewichteter Summen aggregiert. Dieser Ansatz nimmt unter anderem an, dass (1) Technical Criteria und Regeln/Metriken unabhängig sind und (2) das QM ausbalanciert ist (alle Knoten auf der gleichen Ebene des QM haben ungefähr die gleiche Anzahl von Unterknoten). Im Falle des CAST AIP werden jedoch beide Annahmen an einigen Stellen verletzt.

D. Verständlichkeit der Ergebnisse beschäftigt sich mit der Nachvollziehbarkeit und Nützlichkeit der Qualitätsbewertungsergebnisse. Hierbei haben wir untersucht, wie gut die Berechnungsvorschriften beschrieben sind und in welcher Form die Bewertungsergebnisse dem Nutzer vorgestellt werden (zum Beispiel mittels geeigneter Visualisierungen). Dabei haben wir insbesondere betrachtet, wie einfach ein potenzieller Nutzer basierend auf der Bewertungsdokumentation entscheiden kann, wo mit einer Softwarequalitätsverbesserung begonnen werden sollte.

(+) Im Werkzeug sind sowohl Kritikalitäten/Gewichte als auch Bewertungsergebnisse über alle QM-Hierarchieebenen einsehbar. Berechnungsvorschriften

sind sehr gut nachzuvollziehen. Bestehende Detailübersichten ermöglichen Nutzern die Verfolgung der Änderungen an einzelnen Bewertungen über die Zeit wie auch der Beiträge einzelner Bewertungen zur Gesamtbewertung (über zahlreiche Sichten mit „Drill-down“-Möglichkeiten). Hier kann sich die Lösung vom Stand der Praxis deutlich abheben.

(0) Die Untersuchung deckte eine unausgeglichene Struktur des QM auf, was die Verständlichkeit sowie eine konsistente Gewichtung einzelner Elemente im QM erschwert. Damit steht das Modell aber nicht allein, denn unausgeglichene Strukturen sind in der Praxis bei größeren Modellen häufig anzutreffen.

E. Robustheit beschäftigt sich mit der Stabilität der Qualitätsbewertungen. Dabei haben wir untersucht, wie empfindlich die Qualitätsbewertungsergebnisse gegenüber unvollständigen und fehlerhaften Messdaten sowie kleinen Änderungen am QM (zum Beispiel neue Maße, geänderte Gewichtung der Bewertungselemente) reagieren. Zusätzlich haben wir geprüft, ob das QM die Unsicherheit in einem Bewertungsergebnis sichtbar macht.

(0) Bestehende Detailübersichten und Log-Dateien ermöglichen einem Nutzer mit entsprechendem Sachverstand die Identifikation unvollständiger und ungewöhnlicher (gegebenenfalls falscher) Messergebnisse. Jedoch werden solche Messungen bei der Qualitätsbewertung nicht explizit als Unsicherheiten berücksichtigt. Das QM ist robust gegenüber der Anpassung von Gewichtungen der Metriken/Regeln und Technical Criteria, indem es auf solche Änderungen mit überschaubaren Auswirkungen in den Bewertungsergebnissen reagiert. Änderungen an der Kritikalität einzelner Metriken/Regeln haben jedoch teilweise unerwartet starke Auswirkungen auf das Bewertungsergebnis. Ursächlich hierfür ist insbesondere die Unterstützung auch nicht kompensatorischer Bewertung; viele in der Praxis gängige Lösungen entziehen sich diesem Problem, indem einfach keine nicht kompensatorischen Bewertungen angeboten werden. Insofern ist dieser Effekt mit Blick auf die höhere Mächtigkeit des Bewertungsverfahrens beim CAST-QM als akzeptabel und als Teil des Konzepts anzusehen.

F. Anwendbarkeit beschäftigt sich mit der Nützlichkeit des QM in der Praxis. Hierbei haben wir untersucht, wie aufwändig die Durchführung einer Qualitätsbewertung ist, inwiefern das QM die Verbesserung des Softwareprodukts unterstützt und wie gut die Anwendung des QM durch CAST unterstützt wird. Dabei haben wir die Funktionalität der CAST-Software, die Integration in Entwicklungsumgebungen und die Verfügbarkeit von Support (Ansprechpartner, Dokumentation, Schulungsangebote etc.) betrachtet.

(+) Nach einer systemspezifischen Konfiguration erfolgt die Vermessung und Bewertung automatisiert, wobei das Modell sich nicht nur auf monolytische Software, sondern auch auf komplexe Systeme anwenden lässt, die gleichzeitig

mehrere der unterstützten Sprachen und Paradigmen nutzen. Die Werkzeugumgebung ermöglicht dabei die Beobachtung von Trends auf allen Ebenen der Bewertung (insbesondere die Veränderungen gegenüber der letzten Prüfung), was insbesondere bei der Überwachung von Verbesserungsmaßnahmen wichtig ist. Das CAST AIP bietet im Gegensatz zu den meisten anderen Lösungen zudem eine Benchmarking-Datenbank mit mehr als 1.500 realen Applikationen aus über 200 Organisationen. Bei externen Datenpunkten ist dem Nutzer aber unabhängig von der Quelle bei Vergleichen immer zu einer gewissen Vorsicht zu raten. Für nahezu alle Regelverletzungen bietet das Werkzeug zudem Empfehlungen, wie die Qualitätsdefizite beseitigt werden können. In einigen Fällen zeigen sich bestimmte Qualitätsverbesserungen aber nicht direkt in den aggregierten Bewertungen, zum Beispiel wenn die Anzahl der Verletzungen einer kritischen Regel die Maximalgrenze weit übersteigt. Dies kann aber bei Bedarf in entsprechend definierbaren Detailreports überwacht werden.

(-) Die Integration in Nightly-Build-Werkzeuge, die den derzeitigen Stand der Technik darstellt, ist zwar möglich, jedoch nicht Bestandteil der angebotenen Lösung.

G. Anpassbarkeit beschäftigt sich mit der Adaptierbarkeit des QM und der Qualitätsbewertungsmethode an spezifische Anwendungsumgebungen und die Bedürfnisse bestimmter Nutzer [8]. Hierbei haben wir untersucht, inwiefern sich einzelne Elemente und Parameter des QM ändern lassen und ob resultierende Auswirkungen möglichst lokal gehalten werden. Zusätzlich haben wir analysiert, ob ein entsprechender Änderungsleitfaden vorhanden ist und inwiefern Änderungsaktivitäten durch die CAST-Software unterstützt werden.

(+) Das Werkzeug unterstützt das Einbinden, Ändern und Entfernen von Regeln und Metriken (auch für bisher nicht unterstützte Sprachen oder Entwicklungsartefakte) sowie die Anpassung von Gewichten, Grenzwerten und Kritikalitäten für bestehende Regeln/Metriken und Faktoren. Durch das Konzept der schrittweisen lokalen Bewertung und Aggregation in einer baumartigen Struktur können Anpassungen lokal ausgeführt werden. Das Hilfesystem gibt ausführliche technische Hinweise zur Anpassung von QMs. Was jedoch noch fehlt, ist eine Schritt-für-Schritt-Anleitung für typische Anpassungsszenarien mit weiterführenden Hinweisen [10].

6 Fazit

Eine der größten Stärken des CAST-AIP-Qualitätsmodells stellt seine Unterstützung für ein sehr breites Spektrum von Programmiersprachen und Paradigmen dar. Insbesondere ist die zur Verhinderung von Systemausfällen in der Praxis wichtige systemumfassende Analyse über Programmiersprachengrenzen hinweg [18] sowie die Prüfung in Bezug auf Architekturvorgaben eingeschlossen. Im QM werden dabei aber implizit mehrere Annahmen hinsichtlich der QM-Struktur und der Qualität der Eingabedaten gemacht, die nicht vollständig sichergestellt sind. Auch ist die angewandte Bewertungsmethode in manchen Fällen zu empfindlich gegenüber Änderungen.

Eine weitere Stärke der CAST-AIP-Lösung ist die sehr gute Unterstützung der IT-Entscheider als Hauptzielgruppe, wobei auch Funktionen für Entwicklungsverantwortliche vorhanden sind. Insbesondere wird eine Fehleranalyse auf Systemebene und die Priorisierung der gefundenen Fehler ermöglicht. Neben individualisierbaren Dashboards bietet CAST AIP Detailübersichten, die es dem Nutzer erlauben, anhand ausführlicher Beschreibungen die Verletzung einzelner Regeln bis zu den Bewertungen bestimmter Business Criteria nachzuvollziehen. Änderungen in der Softwarequalität lassen sich auf allen Ebenen überwachen und somit kann gezielt gegengesteuert werden. Eine Integration in Nightly-Build-Werkzeuge oder Entwicklungsumgebungen (zum Beispiel Eclipse oder MS Visual Studio), wie man sie beim Stand der Praxis antrifft, ist jedoch, obwohl prinzipiell möglich, nicht Bestandteil der angebotenen Lösung.

Eine Schwäche ist der geringe Bezug zu marktüblichen Softwarequalitätsstandards wie der ISO/IEC 25000 Serie [3]. Das QM orientiert sich weitgehend an einem OMG-Standard [6], der sich derzeit noch in Bearbeitung befindet. Da gleichzeitig jedoch eine ISO/IEC-ähnliche Terminologie mit abweichender Bedeutung verwendet wird, kann dies zu Missverständnissen führen. Für Nutzer, die auf die Verwendung der ISO-Qualitätscharakteristiken angewiesen sind, bietet sich als Behelfslösung die Definition eigener Business Criteria entsprechend der ISO-Norm an, wobei sie hierbei von einer weiteren Stärke der CAST-Lösung profitieren, nämlich der Möglichkeit, auch weitergehende Anpassungen am QM vorzunehmen. So können Nutzer neben neuen Business und Technical Criteria auch neue Metriken und Regeln einbinden, ändern und entfernen sowie Gewichte, Grenzwerte und Kritikalitäten an eigene Gegebenheiten und Bedürfnisse anpassen.

Insgesamt kann man feststellen, dass CAST AIP seine Stärken in der Anpassbarkeit, der Verständlichkeit der Ergebnisse und der praktischen Anwendbarkeit

ausspielen kann. Bezüglich dieser Bewertungsdimensionen setzt es sich deutlich vom heutigen Stand der Praxis ab.

Weiterführende Literatur

1. CAST Product Documentation Portal
(<http://doc.castsoftware.com/help/index.jsp>)
2. Heidrich, J., Rombach, D., Kläs, M., „Model-based Quality Management of Software Development Projects“, in: Software Project Management in a Changing World (Eds. Ruhe, G., Wohlin, C.), Springer, pp. 125–156, 2014
3. ISO/IEC 25000:2014, „Systems and software engineering -- Systems and software Quality Requirements and Evaluation (SQuaRE) -- Guide to SQuaRE“
4. Schubert, C., Kläs, M., Wagner, S., „Softwarequalität erfassen und vergleichen: Von der Messung bis zum abstrakten Qualitätsattribut“, OBJEKTSpektrum, SIGS DATACOM, no. 1, pp. 50–55, 2014
5. Lampasona, C., Kläs, M., Mayr, A., Göb, A., Saft, M., „Early Validation of Software Quality Models with respect to Minimality and Completeness: An Empirical Analysis“, Tagungsband des DASMA Software Metrik Kongresses (MetriKon 2013), Shaker, pp. 123–132, 2013
6. CISQ Specifications for Automated Quality Characteristic Measures (CISQ–TR–2012–01) v2.1, Consortium for IT Software Quality, 2012
(<http://it-cisq.org/wp-content/uploads/2012/09/CISQ-Specification-for-Automated-Quality-Characteristic-Measures.pdf>)
7. Mayr, A., Plösch, R., Kläs, M., Lampasona, C., Saft, M., „A Comprehensive Code-based Quality Model for Embedded Systems – Systematic Development and Validation by Industrial Projects“, Proceedings of the 23rd IEEE International Symposium on Software Reliability Engineering (ISSRE 2012), Dallas, USA, November 27–30th, IEEE, 2012
8. Lampasona, C., Kläs, M., „Application Scenarios and Guidelines for Quality Model Adaptation“, Tagungsband des DASMA Software Metrik Kongresses (MetriKon 2012), Shaker, pp. 3–18, 2012
9. Wagner, S., Lochmann, K., Heinemann, L., Kläs, M., Trendowicz, A., Plösch, R., Seidl, A., Goeb, A., Streit, J., „The Quamoco Product Quality Modelling and Assessment Approach“, Proceedings of the 34th International Conference on Software Engineering (ICSE 2012), Zurich, Switzerland, June 2–9th, ACM/IEEE, pp. 1133–1142, 2012
10. Kläs, M., Lampasona, C., Münch, J., „Adapting Software Quality Models: Practical Challenges, Approach, and First Empirical Results“, Proceedings

- of the 37th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2011), August 30 to September 2, Oulu, Finland, IEEE Computer Society, 2011
11. Wagner, S., Broy, M., Deißböck, F., Kläs, M., Liggesmeyer, P., Münch, J., Streit, J., „Softwarequalitätsmodelle. Praxisempfehlungen und Forschungsagenda“, Informatik Spektrum, vo. 33, no. 1, Springer, pp. 37–44, 2010
 12. Kläs, M., Heidrich, J., Münch, J., Trendowicz, A., „CQML Scheme: A Classification Scheme for Comprehensive Quality Model Landscapes“, Proceedings of the 35th EUROMICRO Conference Software Engineering and Advanced Applications (SEAA 2009), IEEE Computer Society, pp. 243–250, 2009
 13. Hamdan, A., Heidrich, J., Stauner, T., Wickenkamp, A., „Erfahrungen mit der Messung der Wartbarkeit von Steuergeräte-Software“, INFORMATIK 2005 – Informatik LIVE! Band 2, Beiträge der 35. Jahrestagung der Gesellschaft für Informatik e. V. (GI), Bonn, Germany: Springer-Verlag, pp. 171–175, 2005
 14. Lampasona, C., Heidrich, J., Basili, V., Ocampo, A., „Software Quality Modeling Experiences at an Oil Company“, Proceedings of the 6th International Symposium on Empirical Software Engineering and Measurement (ESEM), Lund, Sweden, pp. 243–246, 2012
 15. Figueira, J. R., Greco, S., and Ehrgott, M., „Multiple Criteria Decision Analysis: State of the Art Surveys“, Vol. 78. International Series in Operations Research & Management Science. Springer Verlag, 2005
 16. Trendowicz, A., Kopczyńska, S., „Adapting Multi-Criteria Decision Analysis for Assessing the Quality of Software Products. Current Approaches and Future Perspectives“, Advances in Computers (A. Mamon, ed.), Elsevier, vol. 93, 2014 (accepted for publication)
 17. Liggesmeyer, P., Barthel, H., Ebert, A., Heidrich, J., Keller, P., Yang, Y., Wickenkamp, A., „Quality Improvement Through Visualization of Software and Systems“, in: Mehmet Savsar, „Quality Assurance and Management“, ISBN 978-953-51-0378-3, Chapter 17, 2012
 18. Soley, R., „How to Deliver Resilient, Secure, Efficient, and Easily Changed IT Systems in Line with CISQ Recommendations“, OMG Working Paper, 2012, http://www.omg.org/CISQ_compliant_IT_Systemsv.4-3.pdf

Mögliche Interessenkonflikte

Diese Studie wurde ergebnisunabhängig durch die CAST GmbH beim Fraunhofer-Institut für Experimentelles Software Engineering IESE beauftragt. Die präsentierten Ergebnisse und Schlussfolgerungen sind in Beobachtungen, Recherche- und Gesprächsergebnissen sowie Meinungen der Fachautoren begründet. Trotz sorgfältiger Prüfung kann die Korrektheit der gemachten Aussagen nicht garantiert werden. Eine Haftung für die Richtigkeit, Fehlerfreiheit, Freiheit von Schutzrechten Dritter, Vollständigkeit und/oder Verwendbarkeit der Aussagen wird ausgeschlossen.

Über Fraunhofer IESE

Die Fraunhofer-Gesellschaft betreibt anwendungsorientierte Forschung zum Nutzen der Wirtschaft und zum Vorteil der Gesellschaft. Sie ist die größte Forschungsorganisation für anwendungsorientierte Forschung in Europa. 17.000 Mitarbeiterinnen und Mitarbeiter, überwiegend mit natur- und ingenieurwissenschaftlicher Ausbildung, bearbeiten das jährliche Forschungsvolumen von 1,6 Milliarden Euro.

Als eines von 67 Instituten und Forschungseinrichtungen steht das Fraunhofer-Institut für Experimentelles Software Engineering IESE weltweit für empirisch untermauerte Methoden und Prozesse für die industrielle Software- und Systementwicklung. Wichtige Services sind die Erstellung maßgeschneiderter Qualitätsmodelle und die herstellerunabhängige Eruierung und Einbindung geeigneter Werkzeuge zu deren Umsetzung im Unternehmen. Die Grundlage bildet hierbei eine Historie aus zahlreichen Industrie- und Forschungsprojekten.