Kosten-basierte Klassifikation von Qualitätsanforderungen

Florian Deissenboeck, Stefan Wagner, Markus Pizka
Institut für Informatik
Technische Universität München
{deissenb,wagnerst,pizka}@in.tum.de

Abstract: Obwohl Qualität bekanntermaßen eine komplexe Thematik ist, wird deren Vielschichtigkeit oft ignoriert und versucht, alle Qualitätsanforderungen an Softwaresysteme einheitlich zu behandeln. Diese Pauschalisierung erschwert einen differenzierten und ökonomisch motivierten Umgang mit den Qualitätsanforderungen. Als Lösungsansatz wird eine Strukturierung auf Basis der entstehenden Kosten vorgeschlagen. Die Stakeholder und ihre Aktivitäten werden identifiziert und Qualitätsanforderungen entsprechend zugeordnet. Dadurch ergibt sich eine klare Aufteilung, die zur Priorisierung, Konfliktlösung und in Qualitätsmodellierung sowie -analyse genutzt werden kann.

1 Kosten und Qualität

Im Kontext von *Qualitätsanforderungen* werden üblicherweise die aus der Standardliteratur bekannten Qualitätsattribute wie *Nutzbarkeit, Zuverlässigkeit, Sicherheit* und *Wartbarkeit* diskutiert. Obwohl seit langem bekannt ist, dass es sich bei Qualität um eine äußerst komplexe und vielfältige Thematik handelt [Gar84], wird hierbei oft davon ausgegangen, dass sich diese Attribute, trotz ihrer sehr unterschiedlichen Natur, auf gleiche Art und Weise behandeln lassen. Wir sind der Meinung, dass diese Vorstellung mitverantwortlich für die heute oft unbefriedigende Beherrschung der Qualitätsanforderungen ist (vgl. bspw. [CdPL04]) und schlagen daher eine Systematik zur Strukturierung von Qualitätsanforderungen vor.

Hierbei gehen wir von der Prämisse aus, dass Qualität kein Selbstzweck sein kann, und Qualitätsanforderungen grundsätzlich im Kontext ihrer wirtschaftlichen Bedeutung diskutiert werden sollten. Eine Kosten/Nutzen-Analyse sollte die Basis jeglicher Qualitätsbetrachtungen sein, da

1. der wirtschaftliche Erfolg eines jeden Projekts letztendlich *der* ausschlaggebende Erfolgsindikator ist, und
2. Kosten/Nutzen-Betrachtungen die einzig universell verwendbare Skala darstellen, die eine Berücksichtigung aller Einflussfaktoren ermöglicht.

Dieses Papier beschreibt unseren Kosten-basierten Ansatz anhand ausgewählter Qualitätsattribute, vergleicht ihn mit bestehenden Ansätzen und zeigt Nutzungsmöglichkeiten auf.

2 Qualitätsanforderungen

Eine Kosten-basierte Handhabung der Qualitätsanforderungen erfordert, dass Qualitätsziele grundsätzlich wirtschaftlich motiviert werden und von Projektbeginn an fest in das Requirements Engineering integriert werden. Hierzu müssen aus den, üblicherweise in der Frühphase eines Projektes definierten, strategischen Zielen *(Business Goals)* entsprechende Qualitätsanforderungen abgeleitet werden [GBB+06]. Auf Basis dieser wirtschaftlich fundierten Qualitätsanforderungen kann identifiziert werden, welche Kosten durch die Einhaltung bzw. Nicht-Einhaltung der Qualitätsanforderungen für welche Stakeholder entstehen. Diese Kosten dienen somit als fundierte Grundlage zur Diskussion von Qualitätspriorisierungen und zur Lösung von Zielkonflikten.

2.1 Klassifikation der Qualitätsanforderungen

Bei einer detaillierten Betrachtung der bekannten Qualitätsattribute wie Nutzbarkeit, Zuverlässigkeit, Wartbarkeit, Portierbarkeit und Performanz [ISO03], wird deutlich, dass jedes dieser Attribute einen Einfluss auf die Produktivität unterschiedlicher Aktivitäten, durchgeführt von unterschiedlichen Stakholdern in unterschiedlichen Prozessphasen, hat. So beeinflusst das Attribut Nutzbarkeit offensichtlicherweise, die Aktivität *Nutzung* des Stakeholders *Benutzer*, das Attribut Wartbarkeit hingegen die Aktivität *Wartung*, die normalerweise vom Stakeholder *Entwickler* durchgeführt wird. Je nach Grad der Erfüllung der Qualitätsanforderungen können die betroffenen Stakeholder ihre Aktivitäten unterschiedlich effizient ausführen und verursachen dementsprechende Kosten.

Hierbei ist zu beachten, dass viele Qualitätsattribute komplexer Natur sind und mehreren Aktivitäten mehrerer Stakeholder zugeordnet sind. So beschreibt z. B. das Attribut Performanz meist sowohl Systemeigenschaften, die eine effiziente *Nutzung* durch den *Benutzer* ermöglichen, als auch Eigenschaften, die einen kostengünstigen Betrieb dienen (z. B. niedriger Speicherverbrauch). Zusätzlich muss darauf geachtet werden, dass tatsächliche *alle* betroffenen Stakeholder und die von ihnen durchgeführten Aktivitäten über den *gesamten* Lebenszyklus der Software identifiziert werden (z. B. Schulungsaufwände bei mangelnder Nutzbarkeit). Beispielhafte Zusammenhänge zeigt Abbildung 1.

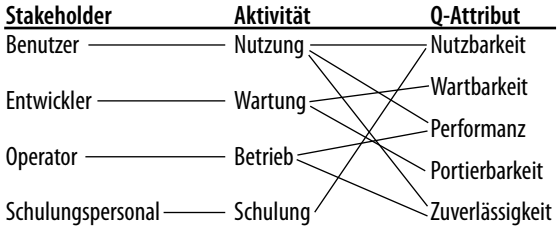

Abbildung 1: Zusammenhang Stakeholder → Aktivitäten → Qualitätsattribute

2.2 Priorisierung und Zielkonflikte

In vielen Fällen muss eine Priorisierung der Qualitätseigenschaften vorgenommen bzw. eine Entscheidungen bei Zielkonflikten bzgl. konkurrierender Qualitätsattribute getroffen werden. Ein Beispiel hierfür sind Modularisierungseigenschaften eines Systems, die sich positiv auf die Wartbarkeit, aber negativ auf die Performanz auswirken können.

Um diese Umstände beschreiben zu können, müssen die im vorherigen Abschnitt erläuterten Zusammenhänge zwischen Qualitätsattributen und Aktivitäten so präzisiert werden, dass ein Attribut-unabhängiger Kostenvergleich ermöglicht wird. Dabei muss die Präzisierung fein-granular genug sein, um aufzeigen zu können, welche einzelnen Systemeigenschaften (z. B. eine Modularisierungseigenschaft) welche Aktivitäten wie beeinflussen.

Im Idealfall würde hierzu eine tatsächlich monetäre Bestimmung des Einflusses der Eigenschaften auf die einzelnen Aktivitäten verwendet und diese entsprechend miteinander verglichen. Da es insbesondere in frühen Prozessphasen sehr aufwändig bis unmöglich ist, jeden dieser Zusammenhänge mit einem tatsächlichen Betrag zu beziffern, bieten sich für diesen Vergleich entsprechende Qualitätsökonomiemodelle an, wie wir sie bereits in [WWD07, BDP06, Wag06] vorgestellt haben.

3 Verwandte Arbeiten

Ein relevanter Standard ist der IEEE Std 830-1998 [IEE98]. Er empfiehlt ein Vorgehen für die Spezifikation von Anforderungen. Stakeholder spielen keine explizite Rolle, nur der Anwender und die Schnittstellen des Systems werden im Detail behandelt. Andere nichtfunktionale Anforderungen finden sich nur in untergeordneten Bereichen. Ebert diskutiert hingegen in [Ebe97] einen Ansatz für den Umgang mit nichtfunktionalen Anforderungen. Er unterteilt die nichtfunktionalen Anforderungen in *benutzerorientiert* und *entwicklungsorientert*, aber basiert diese Unterteilung nicht darauf, bei wem Kosten entstehen.

Boehm und In schlagen in [BI96] ein Vorgehen vor, das dazu dient Konflikte in Qualitätsanforderungen zu entdecken. Dazu haben sie verschiedene Qualitätsanforderungen kategorisiert und Möglichkeiten angegeben, wie mit diesen umgegangen werden sollte. Einen Bezug zu den Stakeholdern stellen sie nicht explizit her. Auch umfassendere Vorgehen, wie z.B. [CdPL04, MCN92, HP06, DKK+05] beziehen die Stakeholder und die entstehenden Kosten nicht oder nur in geringem Umfang ein.

4 Zusammenfassung

Qualität ist nicht klar definiert, sondern setzt sich aus verschiedenen Facetten zusammen [Gar84, ISO03]. Aus diesem Grund ist es auch problematisch, Anforderungen an die Qualität eines Systems gleichartig behandeln zu wollen. Die Performanz einer Software muss mit grundsätzlich anderen Mitteln modelliert und gemessen werden, als die Wartbarkeit.

Dies lässt sich zum Teil auf die involvierten Stakeholder zurückführen. Deshalb schlagen wir eine Klassifizierung der Qualitätsanforderungen vor, die berücksichtigt, bei welchem Stakeholder Kosten im Rahmen welcher Aktivitäten anfallen. Dies erlaubt einen strukturierteren Umgang mit diesen Anforderungen in weiteren Phasen. Wir planen die Klassifizierung in bestehende Requirements Engineering-Vorgehen zu integrieren [GBB+06] und in der Qualitätsmodellierung zu nutzen [WWD07, BDP06, Wag06].

Literatur

[BDP06] Manfred Broy, Florian Deissenboeck und Markus Pizka. Demystifying Maintainability. In *Proc. 4th Workshop on Software Quality*, Seiten 21–26. ACM Press, 2006.

[BI96] Barry Boehm und Hoh In. Identifying Quality-Requirement Conflicts. *IEEE Softw.*, 13(2):25–35, 1996.

[CdPL04] Luiz Marcio Cysneiros und Julio Cesar Sampaio do Prado Leite. Nonfunctional Requirements: From Elicitation to Conceptual Models. *IEEE Trans. Softw. Eng.*, 30(5), 2004.

[DKK+05] Joerg Doerr, Daniel Kerkow, Tom Koenig, Thomas Olsson und Takeshi Suzuki. Non-Functional Requirements in Industry – Three Case Studies Adopting an Experience-based NFR Method. In *Proc. 13th International Conference on Requirements Engineering (RE'05)*, Seiten 373–382. IEEE CS Press, 2005.

[Ebe97] Christof Ebert. Dealing with Nonfunctional Requirements in Large Software Systems. *Ann. Softw. Eng.*, 3:367–395, 1997.

[Gar84] David A. Garvin. What Does »Product Quality« Really Mean? *MIT Sloan Manage. Rev.*, 26(1):25–43, 1984.

[GBB+06] Eva Geisberger, Manfred Broy, Brian Berenbach, Juergen Kazmeier, Daniel Paulish und Arnold Rudorfer. Requirements Engineering Reference Model (REM). Technischer Bericht, Technische Universität München, 2006.

[HP06] Andrea Hermann und Barbara Paech. MOQARE = "Misuse-oriented Quality Requirements Engineering" – Über den Nutzen von Bedrohungsszenarien beim RE von Qualitätsanforderungen. *Softwaretechnik-Trends*, 26(1):13–14, 2006.

[IEE98] IEEE Std 830-1998. *IEEE Recommended Practice for Software Requirements Specifications*, 1998.

[ISO03] ISO 9126: Product Quality – Part 1: Quality Model, 2003.

[MCN92] John Mylopoulos, Lawrence Chung und Brian Nixon. Representing and Using Nonfunctional Requirements: A Process-Oriented Approach. *IEEE Trans. Softw. Eng.*, 18(6):483–497, 1992.

[Wag06] Stefan Wagner. A Model and Sensitivity Analysis of the Quality Economics of Defect-Detection Techniques. In *Proc. International Symposium on Software Testing and Analysis (ISSTA '06)*, Seiten 73–83. ACM Press, 2006.

[WWD07] Sebastian Winter, Stefan Wagner und Florian Deissenboeck. A Comprehensive Model of Usability. In *Proc. Engineering Interactive Systems 2007 (EIS '07)*. Springer, 2007. Zur Veröffentlichung angenommen.