

Qualitätsbewertung beim objektorientierten Entwurf

Ralf Reißing

reissing@informatik.uni-stuttgart.de

Abteilung Software Engineering

Institut für Informatik, Universität Stuttgart

Betreuer der Arbeit: Prof. Dr. Jochen Ludewig

Art der Arbeit: Dissertation

Fachbereich: 2 (2.1.9)

Zusammenfassung

Der Entwurf spielt eine zentrale Rolle in der Softwareentwicklung, weil er die Struktur der Software bestimmt. Sobald er implementiert wurde, sind Änderungen schwierig und teuer. Also ist hohe Entwurfsqualität wichtig, um Softwarekosten zu reduzieren; Qualitätssicherung in der Entwurfsphase lohnt sich also. Leider bleibt dabei oft unklar, was Entwurfsqualität eigentlich ist. Diese Arbeit will zur Klärung des Begriffs beitragen. Es wird dazu ein Qualitätsmodell für den objektorientierten Entwurf aufgestellt, das ausgewählte Kriterien der Entwurfsqualität enthält. Zu jedem Kriterium werden objektive und subjektive Metriken eingeführt. Das Qualitätsmodell kann zum Vergleich von Entwurfsalternativen und zur Schwachstellenanalyse verwendet werden.

1. Problem

Der Entwurf ist ein wichtiger Kostenfaktor in der Softwareentwicklung: Es fallen nicht nur die Kosten für die Erstellung an, sondern er beeinflusst auch die Kosten der nachfolgenden Phasen Implementierung und Wartung in hohem Maße. Bell et al. (1987) haben z. B. festgestellt, dass nur 5-10% des Aufwands über den gesamten Lebenszyklus in die Entwurfsphase fließt, aber bis zu 80% des Aufwands zur Korrektur von schlechten Entwurfsentscheidungen aufgewendet werden muss. Wenn ein schlechter Entwurf nicht in der Entwurfsphase korrigiert wird, sind die Kosten zur Korrektur nach der Auslieferung um den Faktor 5 bis 100 höher (Boehm, Basili, 2001).

Auch wenn der erste Entwurf zunächst genug ist, kann es später Schwierigkeiten geben, wenn in der Wartung Änderungen und Erweiterungen durchgeführt werden sollen. Die Eigenschaften des Entwurfs, z. B. Änderbarkeit, bestimmen, wie leicht solche Erweiterungen vorgenommen werden können. Weil mindestens die Hälfte des Gesamtaufwands in die Wartung geht (Boehm, 1976), lohnt es sich in hohem Maße, so früh wie möglich eine hohe Entwurfsqualität zu haben – und diese über den gesamten Lebenszyklus aufrecht zu erhalten. Dazu muss natürlich klar sein, was Entwurfsqualität bedeutet, d. h. welche Qualitätskriterien in welchem Maße relevant sind.

2. Ziele

Um hohe Entwurfsqualität zu erreichen ist Qualitätssicherung notwendig, z. B. durch Entwurfsbewertung und Reviews. Es ist wichtig und nützlich, Entwurfsqualität so früh wie möglich messen zu können. Starke Korrelationen zwischen Entwurfsmetriken (z. B. Modularitätsmetriken) und der Wartbarkeit wurden z. B. von Rombach (1990) festgestellt, so dass Metriken für die Qualitätssicherung nützlich sind. Natürlich bedeutet frühes Messen beim Entwurf, dass man nicht detaillierte Entwurfsinformation oder sogar Code voraussetzen darf.

Entwurfsmetriken sollten nicht nur eine Messvorschrift, sondern auch eine Anleitung zur Interpretation haben, denn sonst haben die Messungen keinen Sinn. Daher ist es am besten, die Metriken mit einem Qualitätsmodell zu verknüpfen. Das Qualitätsmodell bestimmt die Interpretation der Messwerte in Hinblick auf Qualität – und bestimmt damit die Bedeutung des Qualitätsbegriffs. Da das Qualitätsmodell in Kriterien gegliedert ist, sind die Metriken den einzelnen Kriterien zugeordnet.

Entwurfsbewertung kann von Experten mit Hilfe von Checklisten durchgeführt werden, die auf dem Qualitätsmodell und seinen Metriken aufbauen. Allerdings sind typische Entwürfe so groß, dass die Qualitätssicherung sehr zeitaufwendig wird. Daher ist es effizienter, ein Werkzeug zur Entwurfsbewertung zu verwenden. Auch wenn ein solches Werkzeug einen menschlichen Experten niemals ersetzen kann, kann es ihm die Erhebung der Metriken abnehmen und dabei helfen, Teile des Entwurfs zu identifizieren, die (potentielle) Schwachstellen sind. Außerdem können Entwurfsalternativen verglichen werden, indem die Alternativen bewertet werden.

3. Vorgehensweise

Die Vorgehensweise zur Entwurfsbewertung wird in Abbildung 1 gezeigt. Es wird davon ausgegangen, dass der Entwurf als UML-Modell dargestellt ist. Das UML-Modell ist natürlich eine Instanz des UML-Metamodells (OMG, 2001a). Hier wird die UML als Entwurfsrepräsentation verwendet, weil es sich um die Standardnotation für objektorientierte Entwürfe handelt. UML-Artefakte sind bereits früh in der Entwurfsphase verfügbar und können daher zur Entwurfsbewertung verwendet werden.

Aus dem UML-Modell werden diejenigen Entwurfsartefakte, die für die Entwurfsbewertung relevant sind, herausgezogen. Das Ergebnis ist ein reduziertes Entwurfsmodell, das eine Instanz des Metamodells ODEM (Object-Oriented Design Model) ist. ODEM enthält eine Untermenge des UML-Metamodells, aber auch zusätzliche Mengen und Relationen, welche die Komplexität des UML-Metamodells verbergen und damit die Definition von Metriken erleichtern (Reißing, 2001).

Von dem reduzierten Entwurfsmodell wird ein allgemeines Qualitätsprofil abgeleitet. Dazu wird das Qualitätsmodell QOOD (Quality of Object-Oriented Design) verwendet. Da sich die Qualitätsanforderungen von Projekt zu Projekt unterscheiden, muss das Qualitätsmodell anpassbar sein. Daher ist QOOD ein allgemeines, generisches Qualitätsmodell, das als Ausgangspunkt für spezifische Qualitätsmodelle („MyQOOD“) dient.

Ein spezifisches Qualitätsmodell erlaubt die Ableitung eines spezifischen Qualitätsprofils aus dem allgemeinen Qualitätsprofil. Das spezifische Qualitätsprofil ist die Grundlage für die abschließende Bewertung durch einen Reviewer. Es beruht auf den konkreten Qualitätsanforderungen und einer individuellen Qualitätssicht, zum Beispiel der des Kunden oder des Wartungsentwicklers. Diese Parameter entscheiden, welche Aspekte des allgemeinen Modells relevant und wie wichtig die Aspekte im Verhältnis zueinander sind (wodurch sich eine Gewichtung ergibt).

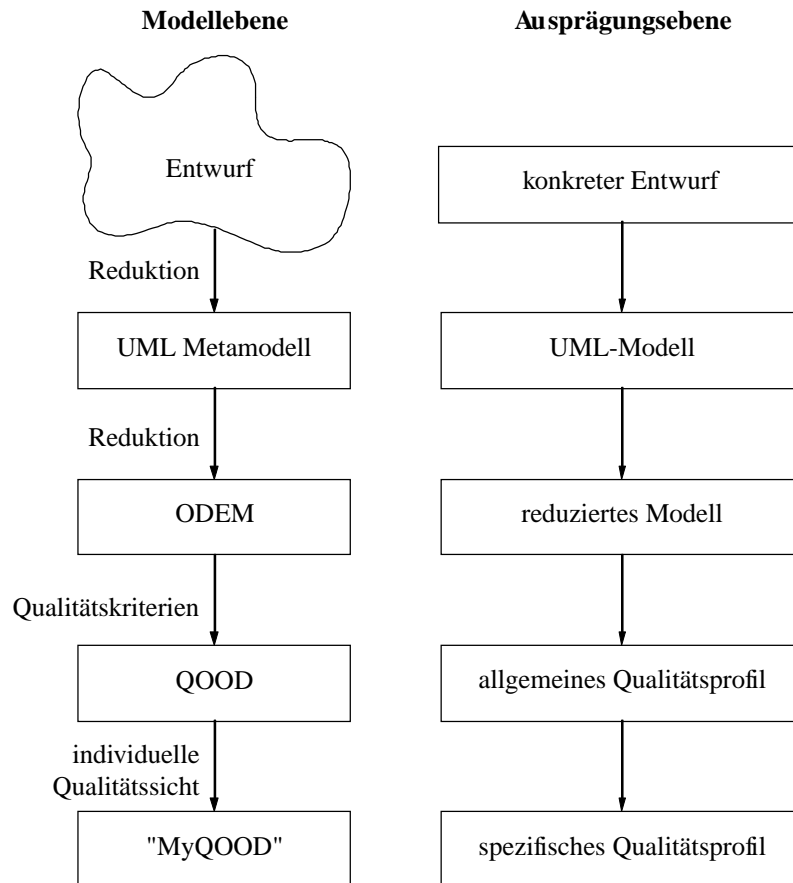


Abbildung 1: Ableitung eines spezifischen Qualitätsprofils

4. Das allgemeine Qualitätsmodell

Um das allgemeine Qualitätsmodell QOOD zu erstellen, wurden zunächst die wichtigsten Aspekte der Softwarequalität ausgewählt, z. B. Wartbarkeit, Wiederverwendbarkeit und Brauchbarkeit. Ein Entwurf hat nämlich dann eine hohe Qualität, wenn er zu hoher Softwarequalität führt. Anschließend wurden für die Aspekte der Softwarequalität Entwurfskriterien identifiziert, die auf die Aspekte positiv wirken. Dazu wurden vor allem die im Bereich der Software- und Entwurfsqualität bereits veröffentlichten Bücher, Artikel und Standards analysiert. Anschließend wurden die Kriterien mit (messbaren) Attributen des Entwurfs verknüpft und Metriken für diese Attribute definiert. Um das häufig vorkommende Problem der unpräzisen Definition von Metriken zu vermeiden, wurden die Metriken formal auf der Basis von ODEM definiert, das hier als formales Referenzmodell des objektorientierten Entwurfs dient.

Einige Kriterien, z. B. Verständlichkeit, lassen sich nicht objektiv messen. Daher braucht man zusätzlich auch subjektive Metriken. Um es dem Entwerfer einfacher zu machen, die subjektiven Metriken zu erheben, wird zu jedem Kriterium ein Fragebogen (ähnlich einer Checkliste) angegeben. Der Fragebogen enthält Fragen nach bestimmten Eigenschaften, die ein Entwurf haben (oder nicht haben) sollte. Eine durch Fragebögen geführte subjektive Bewertung verbessert auch Reproduzierbarkeit und Wiederholbarkeit der subjektiven Messung. Die Fragen können z. B. von bekannten Prinzipien und Heuristiken des objektorientierten Entwurfs abgeleitet werden.

Auch wenn objektive Metriken für ein Kriterium verfügbar sind, erfassen sie oft nicht alle möglichen Aspekte. Daher ist die abschließende Qualitätsmetrik für ein Kriterium eine subjektive Metrik, die von einem Bewerter anhand der vorliegenden Messungen und des Fragebogens erhoben wird. Natürlich ist es möglich, hier anstelle einer subjektiven eine objektive Metrik zu verwenden, die nur aus den objektiven Metriken des Kriteriums berechnet wird. Ein solches Vorgehen erleichtert zwar die Automatisierung der Bewertung, reduziert aber die Breite der Bewertung.

Am Beispiel des Aspekts Wartbarkeit lässt sich zeigen, wie das resultierende allgemeine Qualitätsmodell QOOD aussieht. Abbildung 2 zeigt die Entwurfskriterien zum Aspekt Wartbarkeit. Die Kriterien wurden so gewählt, dass sie sich positiv auf die Entwurfsqualität auswirken.

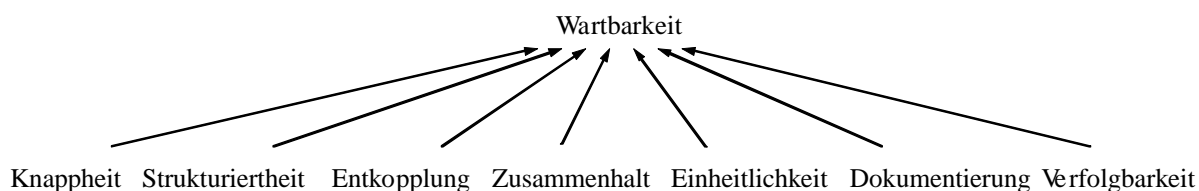


Abbildung 2: Kriterien des Aspekts Wartbarkeit

Knappheit bedeutet, im Entwurf mit möglichst wenig Konstrukten auszukommen, also z. B. mit wenig Klassen und Paketen. Strukturiertheit fokussiert hier auf die Vererbungs- und Schachtelungshierarchie, z. B. geringe Breite und Tiefe. Entkopplung bedeutet die Vermeidung von Abhängigkeiten der Entwurfskomponenten untereinander, z. B. eine geringe Zahl von Assoziationen oder Benutzt-Beziehungen. Zusammenhalt ist eine hohe Zusammengehörigkeit der Bestandteile einer Entwurfskomponente. Einheitlichkeit ist die Einhaltung der Entwurfsstandards und ein einheitlicher Stil. Dokumentierung ist die Güte der Dokumentation der Entwurfskomponenten. Verfolgbarkeit (traceability) ist die Rückführbarkeit der Entwurfskomponenten auf die Anforderungen.

Für einige Kriterien ist es leicht, objektive Metriken auf dem UML-Modell zu erheben, z. B. bei Knappheit und Entkopplung. Hier einige Beispiele für Knappheitsmetriken:

Akronym	Bedeutung	Bereich	Akronym	Bedeutung	Bereich
NAC	number of attributes of a class	Klasse	NAS	number of attributes in the system	System
NOC	number of operations of a class	Klasse	NCS	number of classes in the system	System
NCP	number of classes in a package	Paket	NIS	number of interfaces in the system	System
NIP	number of interfaces in a package	Paket	NOS	number of operations in the system	System
NPP	number of packages in a package	Paket	NPS	number of packages in the system	System

Bei anderen Kriterien ist man vor allem auf die subjektive Einschätzung des Bewerter angewiesen, z. B. bei Zusammenhalt und Einheitlichkeit. Dann spielen die Fragebögen eine umso wichtigere Rolle. Hier ein Auszug aus dem Fragebogen für den Zusammenhalt:

Bereich	Frage
System/ Paket	Bildet das Paket eine abgeschlossene Einheit? (+) (Kriterien: eigenständiger Themenbereich, einheitliche Abstraktionsebene, enthaltene Klassen gehören zusammen)
	Liegen Vererbungsstrukturen vollständig in dem Paket? (+)
Klasse/ Interface	Realisiert die Klasse nur eine (oder sehr wenige) Verantwortlichkeiten? (+) (Wenn viele Operationen vorhanden sind, deutet das auf das Gegenteil hin)
	Ist die Klasse eine abgegrenzte Abstraktion eines Begriffs aus dem Problem- oder Lösungsbereich? (+)
	Verfügt die Klasse über viele get-/set-Operationen? (-) (Das deutet auf eine schlechte Aufteilung hin, da offensichtlich Funktion außerhalb der Klasse ist)

5. Ausblick

Um die Anwendung des Modells zu vereinfachen, wird der Ansatz von einem Werkzeug unterstützt. Das Werkzeug erhebt alle automatisch erhebbaren Metriken auf dem reduzierten UML-Modell. Das Standard-Dateiformat für UML-Modelle ist XMI (XML, Metadata Interchange, siehe OMG, 2001b), so dass die Eingabe des Werkzeugs eine XMI-Datei ist. Zur Zeit wird ein Prototyp des Werkzeugs in einer Diplomarbeit entwickelt.

Das Werkzeug berechnet die Metriken und bereitet die Messungen so auf, dass sie in eine Tabellenkalkulation importiert werden können. Dort können dann auch die subjektiven Metriken erfasst werden, so dass zur Berechnung des spezifischen Qualitätsprofils alle notwendigen Informationen vorliegen. Die Berechnungsvorschriften des spezifischen Qualitätsmodells werden ebenfalls im Spreadsheet realisiert. Zusätzlich kann die Diagrammfunktionalität der Tabellenkalkulation zur graphischen Darstellung der Messungen verwendet werden.

Das Qualitätsmodell wird anhand einer Fallstudie validiert, indem mehrere Entwurfsalternativen eines Projekts bewertet und dann die Ergebnisse mit Expertenmeinungen verglichen werden. Das validierte Modell soll dann in weiteren Fallstudien dazu eingesetzt werden, die Auswirkungen der Anwendung verschiedener Refactorings und Entwurfsmuster auf die Entwurfsqualität zu bewerten. Auf diese Weise kann man Hinweise darauf gewinnen, ob – und wann – die versprochenen Vorteile von Mustern und Refactorings eintreten.

Literatur

Bell, G.; Morrey, I.; Pugh, J. (1987): Software Engineering: A Programming Approach. Prentice Hall, New York, 1987.

Boehm, B. (1976): Software Engineering. IEEE Transactions on Computers 25, 12, 1976, 1226-1241.

Boehm, B.; Basili, V. (2001): Software Defect Reduction Top 10 List. IEEE Computer 34, 1, 2001, 135-137.

Object Management Group (2001a): Unified Modeling Language (UML) Specification. http://www.omg.org/technology/documents/formal/unified_modeling_language.htm

Object Management Group (2001b): XML Metadata Interchange (XMI) Specification. http://www.omg.org/technology/documents/formal/xml_metadata_interchange.htm

Reißing, R. (2001): Towards a Model for Object-Oriented Design Measurement. In: Brito e Abreu, F. et al. (eds.): Proceedings of the 5th International ECOOP Workshop on Quantitative Approaches in Object-Oriented Software Engineering (Budapest, June 2001), 71-84.

Rombach, H. (1990): Design Measurement: Some Lessons Learned. IEEE Software 7, 2, 1990, 17-25.