

1 Qualität

Nach dem Durcharbeiten dieses Kapitels sollten Sie

- die Qualität für ein Softwaresystem definieren können,
- typische Qualitätskriterien kennen,
- Qualitätskriterien messbar festlegen können.

Die Qualität einer Software lässt sich nur schwer beschreiben und trotzdem ist es in der Praxis häufig wichtig, sie genau zu definieren. Wenn ich beispielsweise als Kunde die Entwicklung einer Individualsoftware beauftrage, möchte ich eine gewisse Mindestqualität vertraglich vereinbaren, damit ich im Falle einer schlechten Qualität Nachbesserungen verlangen kann. Als Software-Entwickler möchte ich genau wissen, was dieser Kunde unter hoher Qualität versteht, damit ich seine Wünsche zufriedenstellen kann und Folgeaufträge erhalte. Bei einer Software für den anonymen Markt weiß ich als Software-Entwickler, dass ich ein Produkt immer weiter verbessern kann. Aber wann ist die Qualität hoch genug, damit ich das Produkt ausliefern kann?

Die Qualität einer Software ist entscheidend für ihren Erfolg. Nur wenn eine Software „richtig funktioniert“, ist sie für eine Aufgabe geeignet und kann von Kunden eingesetzt werden. Das bedeutet nicht nur, dass an entscheidenden Stellen keine Fehler auftreten dürfen, sondern auch, dass nicht-funktionale Anforderungen erfüllt werden, z.B. dass bei Steuerungsaufgaben definierte Antwortzeiten eingehalten werden oder dass bei einer Massendatenverarbeitung ein bestimmter Datendurchsatz erreicht wird. Diese Anforderungen werden auch als Qualitätsforderungen bezeichnet (s. Kurseinheit „Grundlagen der Softwaretechnik und Requirements Engineering“, Kapitel 4.5.2).

Qualität ist entscheidend

Es ist Aufgabe der Qualitätssicherung sicherzustellen, dass die Qualitätsforderungen eingehalten werden. Ganz konkret bedeutet dies, dass für jede Qualitätsforderung mindestens eine Maßnahme vorhanden sein muss, die die Erfüllung dieser Qualitätsforderung gewährleistet. Da jede Maßnahme Geld kostet, kostet auch jede Qualitätsforderung Geld. Da die möglichst kostengünstige Produktion ein wesentliches Ziel des Software Engineerings ist (s. Kurseinheit „Grundlagen der Softwaretechnik und Requirements Engineering“), ist es wichtig, nur die absolut notwendigen Qualitätsforderungen sicherzustellen und nur effiziente und effektive Qualitätsmaßnahmen zu ergreifen.

Qualitätssicherung

Qualitätssystem	Erfahrungsgemäß haben Softwareentwickler und Kunden häufig unterschiedliche Vorstellungen, was unter einer Qualitätsforderung genau zu verstehen ist. Aus diesem Grund hat es sich als zweckmäßig erwiesen, Qualitätsforderungen genau zu definieren. Als wichtiges Werkzeug wurden dazu von verschiedenen Seiten so genannte Qualitätssysteme entwickelt. In verschiedenen Standardisierungsgremien und Unternehmen wurden ganz unterschiedliche Qualitätssysteme eingeführt. Die IEEE ¹ definierte beispielsweise den Standard IEEE 1061-1998, den Sie in der Kurseinheit „Grundlagen der Softwaretechnik und Requirements Engineering“ kennen gelernt haben. In Europa wurde der verbreitete Standard ISO ² 9126 entwickelt, der die Grundlage des heute gültigen Standards ISO/IEC 25000:2005 (SQuaRE) darstellt [ISO 25000].
Beispiel Qualitätsforderungen	Das zugrunde liegende Prinzip von Qualitätssystemen soll an einem Beispiel erläutert werden: Ein Kunde fordert eine hohe Effizienz einer Software, insbesondere eine hohe Performance bei einem vorgegebenen Server, auf dem die Software später laufen soll. Im weiteren Gespräch stellt sich heraus, dass er darunter eine geringe Antwortzeit des Systems im Onlinebetrieb und einen hohen Datendurchsatz beim Massendatenimport versteht. Die Antwortzeit des Systems soll auch aus Gründen der Benutzerfreundlichkeit unter Vollast des Systems, d.h. bei 100 angemeldeten, gleichzeitig arbeitenden Nutzern, nicht mehr als 1 Sekunde betragen. Beim Import von angelieferten Massendaten sollen im laufenden Betrieb mindestens 50 Datensätze pro Minute verarbeitet werden können. Der anfängliche Begriff „Effizienz“ wird also durch den Begriff „Performance“ und die Vorschrift einer Ressource („Verbrauchsverhalten“) definiert. Performance wird dabei durch die Forderungen nach „Antwortzeit“ und „Datendurchsatz“ konkretisiert. Diese schrittweise Konkretisierung ist typisch.
Systematik von Qualitätssystemen	Qualitätssysteme sind aus diesem Grund hierarchisch aufgebaut (Abb. 1.1): Die Qualität wird durch so genannte Qualitätsmerkmale definiert, im Beispiel „Effizienz“. Qualitätsmerkmale werden durch Qualitätsteilmerkmale konkretisiert, im Beispiel „Performance“ und „Verbrauchsverhalten“. Qualitätsteilmerkmale wiederum werden anhand von Qualitätsindikatoren genau definiert. Ein Qualitätsindikator beschreibt die einem Artefakt innewohnende Eigenschaft ³ , die objektiv durch Menschen oder – besser – automatisiert

-
- 1 Institute of Electrical and Electronics Engineers, ein Berufsverband von Ingenieuren aus den Bereichen Elektrotechnik und Informatik
 - 2 International Organization for Standardization, die internationale Vereinigung von Normungsorganisationen
 - 3 Eine Eigenschaft, die nicht innewohnend ist, wäre beispielsweise der Preis. Der Preis unterliegt aber nicht der Qualitätssicherung, sondern dem Marketing und der Projektleitung.

bestimmt werden kann, im Beispiel „Antwortzeit“ und „Datendurchsatz“. Dazu werden häufig Metriken eingesetzt. Metriken sind Vorschriften, anhand derer Zahlen berechnet werden können, die ein Maß für die Güte einer Software darstellen. Mehrere Teilmerkmale können sich auch auf einen Qualitätsindikator beziehen. Aus diesem Grund sind in Abb. 1.1 die Qualitätsindikatoren als breiter Balken dargestellt.

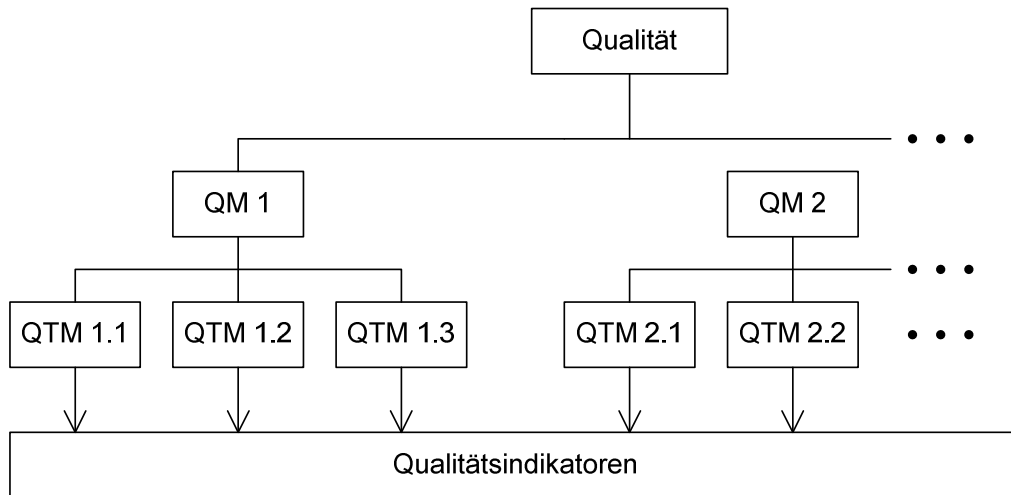


Abb. 1.1: Systematischer Aufbau von Qualitätssystemen. Qualität wird durch Qualitätsmerkmale (QM) definiert, die durch Qualitätsteilmerkmale (QTM) konkretisiert werden, die sich wiederum auf Metriken abstützen.

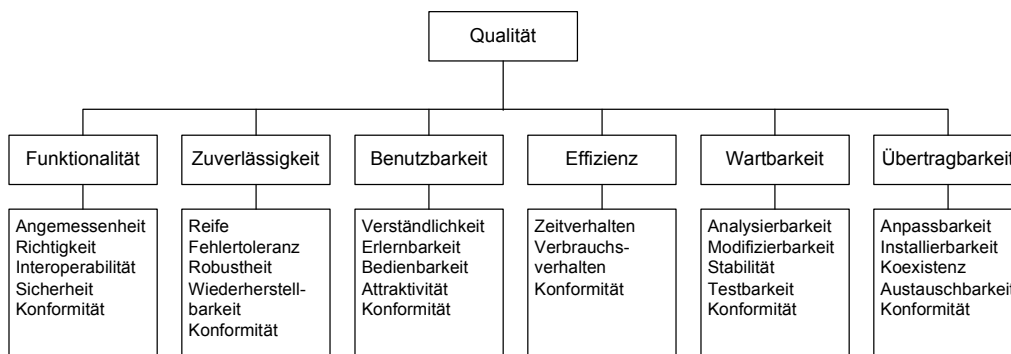


Abb. 1.2: Qualitätsmerkmale und Teilmerkmale nach ISO 9216. Die genauere Definition der Teilmerkmale sowie Qualitätsindikatoren findet sich in öffentlichen Quellen wie den Standards oder auch dem Internet.

Da viele Begriffe wie Benutzerfreundlichkeit gar nicht so einfach zu definieren sind, bietet es sich an, vorgegebene Qualitätsmodelle als Vorlage zu nutzen (Abb. 1.2). Gemeinsam mit Kunden werden aus einer Liste von Quali-

Vorgehen

tätsmerkmalen und Qualitätsteilmerkmalen diejenigen ausgewählt, die für das konkrete Projekt notwendig sind. Nicht immer sind die vorgeschlagenen Qualitätsmerkmale ausreichend oder passend. In diesem Fall müssen die für das Projekt passenden Merkmale selbst definiert werden. Für mögliche Messvorschriften für Qualitätsindikatoren gibt es ebenfalls Vorschläge in den Standards. Damit die definierten Qualitätsforderungen auch tatsächlich erreicht werden, ist es erforderlich, Maßnahmen zu planen, die dies sicherstellen (s. Kapitel 6). Die Erfahrung zeigt, dass Qualitätsforderungen, die nicht überprüft werden, in der Regel auch nicht erfüllt werden.

Genauere Definition durch Szenarios

Die eindeutige Definition von Qualitätsindikatoren ist manchmal schwierig. Dies führt dazu, dass die Definition häufig zu abstrakt bleibt und für die Qualitätssicherung unklar bleibt, was genau zu überprüfen ist. Um die Beschreibung exakter und für die Praxis nutzbar zu machen, können Qualitätsforderungen mittels Szenarien definiert werden. Ein Qualitäts-Szenario besteht aus folgenden Teilen (Abb. 1.3):

- Quelle des Stimulus. Ein Akteur, z.B. ein Mensch, ein Computersystem oder eine Hardware, generiert den Stimulus.
- Stimulus. Ein Vorgang, der etwas im betroffenen System auslöst.
- Umgebung. Der Stimulus tritt in einer bestimmten Umgebung auf, z.B. wird hier die Anzahl der gleichzeitig arbeitenden Benutzer definiert.
- Artefakt. Das können z.B. das gesamte System oder nur Teile davon sein, auch z.B. die Dokumentation.
- Antwort. Die Aktivität, die auf den Stimulus erwartet wird.
- Messvorschrift. Die Messung sollte so genau definiert sein, dass sie objektiv messbar ist.

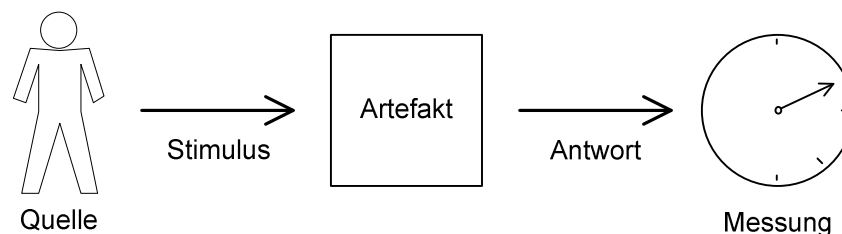


Abb. 1.3: Skizze eines Szenarios zur Beschreibung von Qualitätsforderungen

Als Beispiel soll die Antwortzeit des Beispiels von Seite 2 mittels eines Szenarios definiert werden:

Beispiel für Szenario

- Quelle des Stimulus: Ein normaler Anwender bzw. eine Anwenderin des Systems.
- Stimulus: Der Anwender bzw. die Anwenderin löst in einem Dialog eine Aktion aus.
- Umgebung: Gleichzeitig sind am System 100 Nutzer angemeldet und arbeiten. Die Datenbank des Systems ist mit Daten gefüllt.
- Artefakt: Das System.
- Antwort. Das System antwortet mit einer Bestätigungsanzeige.
- Messvorschrift. Gemessen wird die Zeit zwischen Auslösung und Anzeige der Bestätigung. Sollte auf das System gewartet werden müssen um weiterarbeiten zu können, ist diese Zeit hinzuzuzählen. Wird bei der Funktionalität das Legacy-System genutzt, ist die Antwortzeit des Legacy-Systems getrennt zu messen und von der gesamten Antwortzeit abzuziehen.

Auf diese Art und Weise können viele Qualitätsforderungen definiert werden. Viele Beispiele finden Sie dazu im Buch „Software Architecture in Practice“ [Bas03].

In der Praxis ergibt sich manchmal der Fall, dass sich unterschiedliche Qualitätsforderungen widersprechen. Zum Beispiel können für eine hohe Benutzerfreundlichkeit viele Daten aus einer Datenbank zur Anzeige und Auswahl abgerufen werden und ein ständiger Vergleich von Eingaben mit Daten aus einer Datenbank notwendig sein. Dies könnte sich u. U. mit der Qualitätsforderung eines niedrigen Verbrauchs- und Antwortverhaltens nicht vertragen, was beispielsweise bei mobilen Endgeräten mit geringer Leistung und Netzanbindung zu Problemen führen könnte. In diesem Fall ist es zwingend notwendig, zwischen den verschiedenen Qualitätsforderungen abzuwägen. Die verschiedenen Qualitätsforderungen können auch unterschiedlich große Auswirkungen auf den Erfolg des Produkts haben. Aus diesem Grund ist es wichtig zu wissen, wie wichtig die einzelnen Qualitätsforderungen sind. Zu diesem Zweck sollten Qualitätsforderungen priorisiert werden. Dies kann geschehen, indem z.B. die einzelnen Forderungen mit „sehr wichtig“, „wichtig“ oder „auch wichtig“ oder aber mit Zahlen von 1 (wichtig) bis 10 (eher unwichtig) priorisiert werden.

Priorisierung

Eine so definierte Qualität gibt eine gewisse Sicherheit über ein gemeinsames Verständnis aller Beteiligten hinsichtlich der Qualitätsforderungen in einem Projekt. Die Qualitätsforderungen definieren für die Qualitätssicherung klare Aufgaben. Auf der anderen Seite ist der Begriff Qualität kaum vollständig in

Notwendig aber nicht hinreichend für den Erfolg

ein solches Raster zu fassen. Eine hohe Qualität liegt dann vor, wenn Kunden ein Produkt erhalten, das genau ihren Bedürfnissen entspricht, nicht dann, wenn Qualitätsforderungen erfüllt sind. Dies bedeutet, dass unabhängig von definierten Qualitätsforderungen alle an einer Softwareentwicklung Beteiligten immer wieder ihren gesunden Menschenverstand einsetzen müssen, um zu überprüfen, ob das Produkt tatsächlich später erfolgreich sein kann oder nicht. Formale Qualitätsforderungen sind ein Hilfsmittel, bieten aber keine Garantie.

Zusammenfassung

Qualität kann mit Hilfe von Qualitätssystemen definiert werden. Dazu werden Qualitätsmerkmale durch Qualitätsteilmerkmale detaillierter beschrieben. Die Qualitätsteilmerkmale werden durch Qualitätsindikatoren genau definiert. Dies sind Messvorschriften, auch Metriken genannt, die die Bestimmung von Kennzahlen und deren Sollwerte beschreiben. Ein praktisches Hilfsmittel zur Beschreibung von Qualitätsteilmerkmalen sind auch Szenarien. Eine gute Vorlage für Qualitätssysteme findet sich z.B. in der Norm ISO/EC 25000. Normalerweise werden aus einem Qualitätssystem bestimmte Teilmerkmale ausgesucht, geeignete Metriken gesucht und die relative Wichtigkeit der Teilmerkmale definiert.



Übungsaufgaben

- 1.1 Qualitätsforderungen sind nicht-funktionale Anforderungen. Inwieweit ist auch die Funktionalität einer Software Bestandteil von Qualitätsforderungen?
- 1.2 Welche Qualitätsmerkmale und Teilmerkmale aus der ISO 9126 sind in welcher Reihenfolge wichtig für ein System zur Steuerung von Aufzügen?
- 1.3 Wie könnte „Änderbarkeit von Quellcode“ durch ein Szenario messbar definiert werden?