

## Softwarequalität? Ja, bitte!

**Rückrufaktionen der Automobilhersteller wegen Problemen mit der Elektronik, Geschirrspülmaschinen, die sich irgendwie merkwürdig verhalten, oder der PC, der sich regelmäßig in einem undefinierbaren Zustand befindet. Häufig ist die Software das Problem, die nicht adäquat entwickelt wurde. Doch was macht gute Softwarequalität aus, und wie kann man sie erreichen?**

### Was ist Softwarequalität?

- *Qualitativ hochwertige Software arbeitet zuverlässig:* Zuverlässigkeit spielt insbesondere für sicherheitskritische Systeme die wesentliche Rolle. Ein Airbag muß bei einem Unfall rechtzeitig zünden und darf auf keinesfalls versehentlich zünden. Auch PC-Anwendungen müssen zuverlässig sein. Fehlerhafte Software verärgert den Anwender und vergrault ihn. Ein zuverlässiges Programm muß auch mit Fehlbedienungen umgehen können.
- *Qualitativ hochwertige Software ist anwenderfreundlich:* Die Software muß sich nach den Bedürfnissen der Anwender richten, nicht andersherum. Im Idealfall ist eine Benutzerschnittstelle so intuitiv gestaltet, daß auf Gebrauchsanweisung oder Online-Hilfe fast verzichtet werden kann.
- *Qualitativ hochwertige Software besteht aus verständlichem Sourcecode:* Programmierrichtlinien und ein durchdachter Softwareentwurf sind der beste Weg zu mehr Verständlichkeit und zur Vermeidung von Fehlern. In den Programmierrichtlinien wird u.a. festgelegt, in welcher einheitlichen Form Variablen und Funktionen benannt werden. Kommentare sorgen für Durchblick beim Programmierer, wenn er sein Werk nach längerer Zeit wieder zur Hand nimmt, und bei neuen Kollegen.
- *Qualitativ hochwertige Software läßt sich testen:* Softwaretests müssen schon in der Entwurfsphase eingeplant werden und stellen eigentlich bereits vom Umfang her ein eigenes Projekt dar. Dabei müssen Entwickler und Testteam eng zusammenarbeiten. Nur über getestete Software kann man sich ein positives Urteil bilden.
- *Qualitativ hochwertige Software läßt sich verändern:* Es muß immer möglich sein, Software geänderten Verhältnissen anzupassen. Häufig erkennt der Auftraggeber neue Anforderungen erst während des Projekts, oder technische neue Entwicklungen machen Anpassungen erforderlich.
- *Qualitativ hochwertige Software ist portierbar:* Mit Portabilität ist die Fähigkeit gemeint, Software auf anderen Zielplattformen laufen zu lassen als die, für die sie ursprünglich entwickelt wurde. Ein mo-

dularer Aufbau und eine möglichst plattformunabhängige Entwicklung sind ein probates Mittel.

- *Qualitativ hochwertige Software ist effizient:* Hiermit ist vornehmlich der ressourcenschonende Umgang mit Speicherplatz und Rechnerleistung gemeint. Beides steht in eingebetteten Systemen in der Regel nur in begrenztem Maß zur Verfügung.

### Software-Metriken

Zur Bewertung der Softwarequalität bedient man sich sogenannter Metriken. Man unterscheidet Produktmetriken, die die Software selbst bewerten, und Prozeßmetriken zur Bewertung des Entwicklungsprozesses.

Bekannteste Produktmetrik und Grundlage für weitere Metriken ist der Softwareumfang, gemessen in *Lines of Code* (LoC). Dieser kann zur Messung der Produktivität des Teams während eines Projekts herangezogen werden. Dabei wird im Quelltext jede Anweisung, die mit einem Begrenzer – in „C“ ist das z.B. das Semikolon – als eine Codezeile gezählt. Als Alternative kommt auch die aufwendigere *Function Point*-Analyse als Meßmethode für den Softwareumfang in Betracht. Dazu wird auf Basis des Softwareentwurfs die Anzahl der Eingaben, der Ausgabeoperationen, der Dateizugriffe, der externen Schnittstellen und die Anzahl der Anfragen, die von außerhalb eines Moduls eintreffen können, ermittelt, gewichtet, aufaddiert und mit einem Korrekturfaktor versehen.

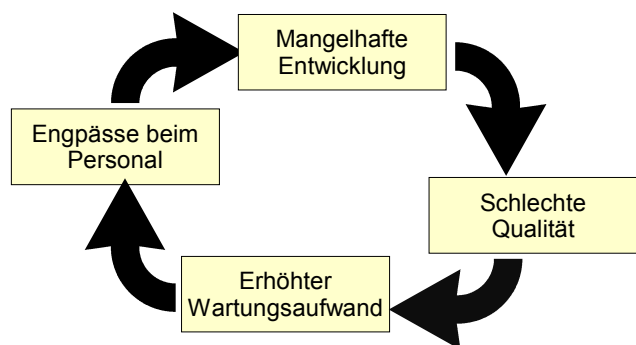


Abb.: Teufelskreis des Software-Engineerings

Eine weitere Produktmetrik ist die Fehlerrate. Mit ihr werden während der Entwicklung die gesammelten und protokollierten Softwarefehler einzelner Module und des Gesamtsystems mit den *Lines of Code* bzw. *Function Points* in Relation gesetzt. Fehlerraten sollten vom Management vertraulich behandelt werden!

Die Restfehlerrate ist die Fehlerrate für bereits ausgelieferte Software; also Fehler, die vom Kunden zurückgemeldet werden. Die Restfehlerrate wird auch dazu herangezogen, um verschiedene, Projekte, die gleichartig sind, miteinander zu vergleichen.

Die Fehler je Entwicklungsphase sind eine Metrik zur Bewertung des Entwicklungsprozesses, mit der die

einzelnen Entwicklungsphasen eines Projekts bewertet werden. Dazu wird ermittelt, wie viele Fehler in jeder Stufe des verwendeten Vorgehensmodells aufgetreten sind. Zum Beispiel: Anzahl der Fehler in der Spezifikation, im Systementwurf, in der Implementierungsphase oder nach Übergabe an den Kunden. Die Ergebnisse unterschiedlicher Projekte können somit verglichen werden.

## Qualitätsmaßnahmen

Was konkret kann getan werden, um die Softwarequalität zu steigern? Ein paar Möglichkeiten seien hier kurz vorgestellt.

- *Dokumentation der Anforderungen:* Gemeinsam mit dem Auftraggeber wird die Spezifikation entwickelt. Diese muß frei von Widersprüchen sein und alle Anforderungen enthalten. Die Spezifikation bildet die Grundlage für alle weiteren Projektschritte.
- *Schaffung eines Fehlerberichtswesens* zur Erfassung und Bewertung von aufgetretenen Softwarefehlern, die vor oder nach Auslieferung der Software aufgetreten sind. Ein Änderungsmanagement verwaltet und verarbeitet die Fehlerberichte und trägt dafür Sorge, daß entdeckte Fehler beseitigt werden. Auch Änderungswünsche des Auftraggebers werden hier erfaßt, bewertet und umgesetzt.
- Eine *Aufstellung von Programmierrichtlinien*, die für alle Entwickler verbindlich ist, muß definiert werden. Dazu zählen u.a. einheitliche *Header* zur eindeutigen Identifizierung von Quelltexten und deren Entwickler, Vorgaben zur Benennung von Modulen, Funktionen, Variablen und Konstanten. Und: Vorgaben zur Kommentierung und Dokumentation des Quelltextes, was beides häufig vernachlässigt wird.
- *Einrichten eines Konfigurationsmanagements* zur eindeutigen Identifikation aller Softwarekomponenten und zur Versionskontrolle. Dazu zählen auch Dokumente wie die Spezifikation, Testresultate und die Dokumentation. Insbesondere müssen Änderungen zurückverfolgbar sein. Das Konfigurationsmanagement hat stets die aktuellsten Versionen parat. Sämtliche Auslieferungen von Softwareversionen an den Kunden werden hierüber abgewickelt.
- *Überprüfung des Softwareentwurfs:* Insbesondere Schnittstellen, Daten- und Kontrollfluß der zu entwickelnden Software werden überprüft und ermittelt, ob sie mit den Vorgaben der Spezifikation vollständig übereinstimmen. Es sollte auch untersucht werden, ob Testmöglichkeiten im Entwurf ausreichend berücksichtigt worden sind.
- Regelmäßige Durchführung von *Code Walk-throughs* und *Reviews*, bei denen geprüft wird, ob

entwickelte Module den Anforderungen entsprechen. Entdeckte Mängel werden protokolliert und müssen von den Entwicklern innerhalb einer gegebenen Frist abgearbeitet werden. In einem Folge-Review wird das Softwaremodul erneut überprüft.

- *Planung und Durchführung von Softwaretests* zur Verifikation der Software. Mit *White Box Tests* wird die Programmlogik überprüft, jeder Programmpfad muß abgedeckt werden. Mit *Black Box Tests* wird das funktionale Verhalten eines Softwaremoduls getestet. Dafür werden bestimmte vorbereitete Testfälle durchgespielt. Die Testergebnisse werden protokolliert. Die eingesetzte Testumgebung, Hardware und Software, wird dokumentiert, um entdeckte Fehler reproduzierbar zu machen. Es ist sinnvoll, daß der Entwickler eines Moduls nicht auch der Tester ist.
- *Validierung der Software* zwecks Überprüfung, ob die Software den Anforderungen der Kunden gerecht wird. Dazu wird je nach Art der Software z.B. überprüft, ob das System hohen Belastungen standhält, alle geforderten Funktionen implementiert worden sind, die Software anwenderfreundlich ist, Sicherheitsanforderungen erfüllt sind, das System wartungsfreundlich ist usw. Kurz: Stimmt das Endprodukt mit den Anforderungen in der Spezifikation überein?

**Fazit:** Softwarequalität ist machbar, trotz zunehmender Komplexität der Anwendungen. Zunächst muß sich jeder Projektbeteiligte darüber bewußt werden, was mit Softwarequalität gemeint ist und welche Ziele verfolgt werden. Die Anforderungen des Auftraggebers sind ein weiterer Punkt. Nur auf Basis von vollständigen und in sich konsistenten Vorgaben kann Qualität überhaupt erst entstehen. Der Entwicklungsprozeß muß von qualitätssichernden Maßnahmen flankiert werden. Hier ist das Projektmanagement gefordert. Und schließlich: Durch die Verwendung von Metriken kann die Qualität des Produkts und des Entwicklungsprozesses quantifiziert werden. (bar)

## Weitere Informationen

Bei weiteren Fragen stehen wir Ihnen gerne zur Verfügung.

Ingenieurbüro Barheine  
Albstraße 47  
76275 Ettlingen

Tel.: 0 72 43 / 52 37-67

Fax.: 0 72 43 / 52 37-68

E-Mail: [kontakt@barheine.de](mailto:kontakt@barheine.de)

Web: <http://www.barheine.de>