

Grundlagen des Software-Testens – Übung 2

Gruppe 8: Felix Jendrusch, Sascha Gennrich,
Johannes Klick, Irena Kpogbezan

24. August 2012

Theoretische Aufgabe 1 – Herausforderungen bei der Qualitätssicherung

Zunehmender Einsatz von Software Software wird heutzutage in immer mehr Bereichen eingesetzt, in denen Fehler schwere Konsequenzen nach sich ziehen können (z.B. Operationsroboter, Katastrophenwarnsysteme, Produktion). Damit sinkt die Fehlertoleranz.

Steigende Komplexität Software wird immer komplexer. In dieser Komplexität können Fehler leichter entstehen, aber auch leichter übersehen werden.

Höhere Kundenerwartungen Bei der Fülle von Softwareanbietern heutzutage muss ein Softwareprodukt höhere Standards erfüllen, um sich gegen Konkurrenz behaupten zu können.

Outsourcing Durch die Trennung von der Planung und der Entwicklung entstehen natürlich Diskrepanzen, die sich auf die Qualität der Software auswirken können, indem beispielsweise auf Entwicklerseite die Spezifikation falsch verstanden wurde.

Streben nach effizienter Entwicklung Wenn Software möglichst schnell oder preiswert entwickelt werden soll, müssen oft Abstriche bei der Qualität gemacht werden.

Abstraktion Die meisten heutigen software-basierten Systeme haben viele Abstraktionsebenen. Die Qualitätssicherung einer hohen Abstraktionsebene kann durch nur eingeschränkte Sicht auf niedrigere Abstraktionseben sehr anspruchsvoll werden.

Theoretische Aufgabe 2 – interne und externe Softwarequalität

Innere und äußere Qualität bezeichnen die verschiedenen Qualitätssichtweisen auf das zu testende Produkt. Bei der inneren Sichtweise betrachtet man das Produkt aus Entwicklersicht. So spielt es z.B. bei der inneren Qualität eine Rolle, ob das Produkt eine gute Änderbarkeit aufweist. Eine gute Änderbarkeit ermöglicht es Entwicklern – ohne hohen Aufwand – Modifizierungen an der Software vorzunehmen, um diese z.B. an eine neue Kundenumgebung bzw. neue Kundenanforderung anzupassen.

Als Beispiel könnte man hier stark modulare Software aufführen, bei der man ggf. nur ein Modul ändern müsste ohne die ganze Software anpassen zu müssen. Natürlich bedarf es vor einer Modifikation erst einer guten Analysierbarkeit der Software. Schließlich muss zunächst festgestellt werden wo und wie zukünftige Modifikationen vorgenommen werden können/sollen. Nach einer gegebenen Änderung bedarf es Tests, die einen gewissen Aufwand benötigen. Diesen Aufwand bezeichnet man als inneres Merkmal bzw. Testbarkeit. Die Tests können auch Aussagen über die Stabilität eines Programms treffen. Bei den bisher vorgestellten Merkmalen handelte es sich um innere Qualitätsmerkmale, welche vorallem für den Entwickler von Interesse sind bzw. einer inneren Sichtweise entsprechen.

Das äußere Merkmal der Benutzbarkeit stellt eindeutig eine Sicht des Nutzers dar. So ist z.B. für den Nutzer die Analysierbarkeit der Software weit weniger Interessant als die Bedienbarkeit. Denn der Nutzer möchte die Software nicht ändern sondern mit ihr arbeiten. Aus diesem Grund ist die Erlernbarkeit ebenfalls von hoher Bedeutung. Eine Software die Verständlich und leicht erlernbar ist, ist auch einfacher in einem Unternehmen durchzusetzen und daher von höherem Interesse. Es lässt sich zusammenfassend sagen, dass die äußere Qualität sich im Vergleich zur inneren Qualität auf anwendungsbezogene Merkmale bzw. Nutzerkriterien bezieht, während die innere Qualität sich auf Merkmale der Entwicklersicht, z.B., der Softwarekonstruktion und technisch-spezifischen Eigenschaften, bezieht.