Physikalisch-Technische Bundesanstalt, Berlin (Adlershof)

Vorlesung: Wissenschaftliche Softwareentwicklung

2023/24

Dr. habil. Till Biskup

— Glossar zu Lektion 16: „Tests“ —

Hinweis: Die nachfolgend genannten Begriffe und Definitionen erheben keinen Anspruch auf formale Korrektheit, sondern dienen lediglich dem besseren Verständnis der in der Vorlesung behandelten Themen und sind im jeweiligen Kontext zu sehen. Mehrfache, voneinander abweichende Definitionen in unterschiedlichen Kontexten sind daher möglich. Englische Begriffe werden zwar nach Möglichkeit übersetzt, erscheinen aber ggf. unter ihrem englischen Namen in der Liste. Verweise untereinander sind durch ↑ gekennzeichnet.

Black-Box-Test ↑Test einer zu testenden Einheit ausschließlich über öffentliche Schnittstellen, ohne irgendwelche Annahmen über die Implementierung bzw. interne Abläufe zu treffen. ↑Unittests sind i.d.R. gute Beispiele solcher Black-Box-Tests. Für das Gegenteil vgl. ↑White-Box-Test.

Bug (engl. für „Wanze, Käfer“) Programmfehler oder Softwarefehler, allgemein ein Fehlverhalten von Computerprogrammen.

Debugger Werkzeug zur Diagnose und zum Auffinden von Fehlern (↑Bug) in Programmen. Debugger bringen eine Reihe hilfreicher Funktionalitäten zur Analyse eines in einem Programmablauf aufgetretenen Fehlers mit und sind oft in einer IDE integriert.

Debugging Behebung von Fehlern (↑Bugs) in Software, häufig unter Verwendung eines ↑Debuggers. Sollte immer ein systematisches Vorgehen sein.

Integrationstest ↑Test des Zusammenspiels mehrerer Codeblöcke. Einzelne Codeblöcke wurden vorher über ↑Unittests überprüft. Entsprechend finden Integrationstests nicht mehr (notwendigerweise) in Isolation statt. Vgl. ↑Unittest, ↑Systemtest.

McCabe-Metrik Anzahl linear unabhängiger Pfade auf dem Kontrollflussgraphen eines Moduls und damit ein Maß für die Komplexität. Gleichzeitig Maß für die minimale Zahl von ↑Tests eines Moduls, da i.d.R. für jeden unabhängigen Pfad (mindestens) ein Test benötigt wird. Die reale Zahl von Tests wird meist größer sein, da für einen Pfad ggf. mehrere *charakteristische* Parameter aus dem möglichen ↑Parameterraum überprüft werden sollten.

Parameterraum Summe aller möglichen gültigen Parameter, die einer Funktion bzw. Methode übergeben werden können. Parallele aus der Mathematik: Definitionsbereich bzw. Definitionsmenge. Der vollständige Parameterraum selbst sehr einfacher Funktionen lässt sich aufgrund seiner Größe i.d.R. nie überprüfen. Deshalb sollten für ↑Tests charakteristische Parameterkombinationen aus dem Parameterraum verwendet werden.

Regression erneutes Auftreten eines einmal behobenen Fehlers in einer neuen Version. Lässt sich grundsätzlich dadurch umgehen, dass einmal detektierte Fehler in ↑Tests, meist ↑Unittests, verwandelt werden und jede zukünftige Version einer Software entsprechend (automatisch) auf das Auftreten dieses Fehlers überprüft wird (↑Testsuiten).

Regressionstest ↑Test auf Übereinstimmung mit früherer Funktionalität. Diese Form der Überprüfung ist relevant bei Veränderungen im Co-

de, da sie sicher stellt, dass korrekter Code korrekt bleibt. Die Voraussetzung ist natürlich die Korrektheit der verwendeten Tests. Als Tests können hier sowohl ↑Unittests als auch ↑Integrationstests zum Einsatz kommen. Je höher die Testabdeckung und je kleiner die überprüfte Einheit (↑Unittests!), desto einfacher ist es, bei einer ↑Regression die Ursache zu finden und zu beheben.

Systemtest ↑Test des Gesamtsystems und seines Zusammenspiels. Die einzelnen Komponenten werden auf unterster Ebene über ↑Unittests überprüft, das Zusammenspiel dieser Einheiten über ↑Integrationstests. Inwieweit sich Systemtests immer automatisieren lassen, hängt sehr vom jeweiligen System ab. Vgl. ↑Integrationstest, ↑Unittest

Test hier: strukturiertes Vorgehen, eine Software zu überprüfen. Setzt die Definition klarer Anfangs- und Endbedingungen (Eingabe und Ergebnis) voraus und sollte idealerweise vollständig automatisiert ablaufen können. Vgl. ↑Unittest.

Testsuiten Sammlungen von Testroutinen, die in einem funktionalen Zusammenhang stehen, z.B. einen bestimmten Teil eines Programmes abtesten. Lassen sich meist automatisch ausführen. Bestehen oft aus ↑Unittests.

Unittest ↑Test eines Codeblocks in Isolation. Ein Unittest überprüft von außen, ohne den Quellcode des zu testenden Systems zu kennen oder zu benötigen. Die getesteten Codeblöcke sind i.d.R. klein. Zwingende Voraussetzung ist, dass das (erwünschte) Verhalten des zu testenden Codeblocks eindeutig definierbar (und seinerseits in Form von Quellcode formalisierbar) ist. Unittests sind gewissermaßen die unterste Ebene (automatisierter) ↑Tests. Vgl. ↑Integrationstest, ↑Systemtest

White-Box-Test ↑Test einer zu testenden Einheit unter Kenntnis ihrer Implementierung bzw. internen Abläufe. White-Box-Tests haben häufig das Problem, dass sie sehr eng mit dem zu testenden Code verknüpft sind und zu viele Annahmen über dessen interne Abläufe treffen. Das führt oft dazu, dass solche Tests bei Veränderung des überprüften Codes ebenfalls mit verändert werden müssen. Vgl. als Alternative ↑Black-Box-Tests.