LU01e - Teststufen

Auf einen Blick
Software wird nicht auf einmal getestet, sondern in Stufen: zuerst die einzelnen Bausteine, dann ihr Zusammenspiel, dann das ganze System und zum Schluss durch den Kunden. Jede Stufe findet andere Fehler.

Lernziele

Sie können

- die vier Teststufen nennen und unterscheiden,
- für jede Stufe angeben, wer testet und worauf die Testfälle beruhen,
- die Teststufen den Phasen der Entwicklung zuordnen (V-Modell).

Übersicht

Teststufe	Was wird getestet?	Wer testet?	Grundlage der Testfälle	Testart
Komponententest	einzelne Funktionen, Klassen, Module	Entwicklerinnen und Entwickler	Quellcode, Feinentwurf	Glassbox
Integrationstest	Zusammenspiel der Komponenten	Entwicklungsteam	Architektur, Schnittstellen	Glass- und Blackbox
Systemtest	das ganze System in seiner Umgebung	unabhängiges Testteam	Anforderungen, Spezifikation	Blackbox
Abnahmetest	Erfüllt das System die Kundenwünsche?	Kundin, Auftraggeber	Vertrag, Anforderungen	Blackbox

Das V-Modell

Das V-Modell zeigt, dass jede Phase der Entwicklung eine passende Teststufe hat. Links wird geplant und gebaut, rechts wird geprüft.

Entwicklungsphase (links)	↔	Teststufe (rechts)	prüft, ob …
Anforderungen des Kunden	↔	Abnahmetest	… das System die Kundenwünsche erfüllt
Spezifikation des Systems	↔	Systemtest	… das System die Spezifikation erfüllt
Architektur, Schnittstellen	↔	Integrationstest	… die Komponenten richtig zusammenarbeiten
Feinentwurf der Komponenten	↔	Komponententest	… jede Komponente für sich richtig arbeitet

Die Testfälle einer Stufe können geschrieben werden, sobald die passende Entwicklungsphase abgeschlossen ist. Die Abnahmetests lassen sich also schon zu Projektbeginn aus den Anforderungen ableiten.

Komponententest (Unit-Test)

Metapher Armbanduhr: Jedes Zahnrad wird einzeln geprüft.

- Stimmen Durchmesser, Anzahl und Winkel der Zähne?
- Hat das Material Fehler?
- Wie stark kann das Zahnrad belastet werden, bevor es bricht?

Beim Komponententest prüfen die Entwickler einzelne Bausteine der Software, zum Beispiel eine Funktion oder eine Klasse. Die Komponente wird dabei vom Rest des Programms getrennt getestet. Jede Anweisung, Schleife und Verzweigung soll mindestens einmal ausgeführt werden (siehe LU02c - Test Coverage).

Komponententests werden meistens **automatisiert**, zum Beispiel mit PyTest. So können sie nach jeder Änderung in Sekunden wiederholt werden.

Integrationstest

Metapher Armbanduhr: Zahnräder, Federn und Zeiger werden zum Uhrwerk zusammengesetzt.

- Läuft das Uhrwerk gleichmässig und mit der richtigen Geschwindigkeit?
- Bewegen sich die Zeiger ohne Widerstand über das Zifferblatt?

Nachdem die Komponenten einzeln getestet wurden, prüft der Integrationstest ihr Zusammenspiel. Im Vordergrund stehen die **Schnittstellen**: Werden die richtigen Daten im richtigen Format übergeben? Typische Fehler sind vertauschte Parameter, unterschiedliche Einheiten (Franken und Rappen) oder falsche Annahmen über Rückgabewerte.

Systemtest

Metapher Armbanduhr: Die fertige Uhr wird geprüft.

- Läuft sie genau und ohne übermässiges Ticken?
- Passt das Armband zum Gehäuse?
- Lässt sich die Uhr angenehm tragen?

Im Systemtest wird das ganze System in einer Umgebung getestet, die dem späteren Betrieb

möglichst ähnlich ist. Ein unabhängiges Testteam leitet die Testfälle aus den Anforderungen ab, ohne den Code anzuschauen (Blackbox).

Neben den Funktionen werden hier auch **nicht-funktionale** Eigenschaften geprüft:

- Leistung: Wie schnell antwortet das System bei 500 gleichzeitigen Benutzern?
- Benutzbarkeit: Finden sich neue Benutzer ohne Anleitung zurecht?
- Sicherheit: Sind die Daten vor unberechtigtem Zugriff geschützt?
- Kompatibilität: Läuft die Anwendung in allen vorgesehenen Browsern?

Abnahmetest (Akzeptanztest)

Metapher Armbanduhr: Die Kundin probiert die Uhr an. Entspricht sie ihren Wünschen, kauft sie die Uhr.

Beim Abnahmetest prüft der **Kunde** oder die Auftraggeberin, ob die Software die vereinbarten Anforderungen erfüllt und im Alltag brauchbar ist. Der erfolgreiche Abnahmetest ist meistens die Voraussetzung für die Freigabe und die Bezahlung der Software.

Beispiel: Webseite British Inheritance Tax

Teststufe	**Beispiel für einen Test**
Komponententest	Die Funktion, die den Steuersatz bestimmt, liefert für 4.5 Jahre den Wert 24 %.
Integrationstest	Der Klick auf **Berechnen** übergibt die Werte aus dem Formular korrekt an die Berechnung.
Systemtest	Die Webseite funktioniert in Firefox, Chrome und Safari und auf dem Smartphone.
Abnahmetest	Eine Steuerberaterin prüft mit echten Fällen, ob die Resultate stimmen.

Regressionstest

Der Regressionstest ist keine eigene Stufe, sondern kommt auf **jeder** Stufe vor. Nach jeder Änderung am Programm werden die bestehenden Testfälle erneut ausgeführt, um sicherzustellen, dass die Änderung keine neuen Fehler verursacht hat. Weil das sehr oft passiert, lohnt sich hier die Automatisierung besonders.

Kontrollfragen

1. Warum testet man nicht einfach nur das fertige System?
2. Auf welcher Teststufe findet man am ehesten den Fehler «Funktion erwartet Rappen, erhält aber Franken»?
3. Wer führt den Abnahmetest durch und warum nicht das Entwicklungsteam?
4. Ordnen Sie zu: «Die Seite lädt in weniger als 2 Sekunden.» Welche Teststufe, welche Art von Anforderung?

Marcel Suter

From:
https://wiki.bzz.ch/ - **BZZ - Modulwiki**

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu01/teststufen?rev=1794390758

Last update: **2026/11/11 10:52**