LU02b - Testziele

Auf einen Blick
Ein Testziel ist eine **Frage**, die wir mit Testfällen beantworten wollen, zum Beispiel: «Lehnt das Programm negative Beträge ab?» Klare Testziele bestimmen, welche Testfälle wir brauchen und wann wir mit dem Testen fertig sind.

Lernziele

Sie können

- erklären, was ein Testziel ist,
- prüfbare Testziele formulieren,
- Verifikation und Validierung unterscheiden,
- die wichtigsten Arten von Testzielen nennen und mit Beispielen belegen.

Gute Testziele formulieren

Ein gutes Testziel ist so formuliert, dass man nach dem Test eindeutig mit **Ja** oder **Nein** antworten kann.

Unklar	**Prüfbar**
Das Programm soll gut funktionieren.	Berechnet das Programm für alle Zeilen der Steuertabelle den richtigen Steuersatz?
Die Seite soll schnell sein.	Wird das Resultat nach dem Klick auf **Berechnen** in weniger als 1 Sekunde angezeigt?
Eingaben sollen geprüft werden.	Erscheint bei jeder Eingabe, die keine Zahl ist, die Meldung «Keine gültige Zahl»?

Aus jedem Testziel entstehen ein oder mehrere Testfälle.

Verifikation und Validierung

	Verifikation	**Validierung**
Leitfrage	Bauen wir das Produkt **richtig**?	Bauen wir das **richtige** Produkt?
Vergleich mit	Spezifikation und Entwurf	Erwartungen und Bedürfnissen der Benutzer
Beispiel	Der Steuersatz für 4.5 Jahre ist wie spezifiziert 24 %.	Die Steuerberaterin kann die Seite für ihre echten Fälle brauchen.
Typische Stufe	Komponenten-, Integrations-, Systemtest	Abnahmetest

Ein Programm kann perfekt verifiziert sein und trotzdem niemandem nützen, wenn die Spezifikation an den Bedürfnissen vorbeigeht.

Arten von Testzielen

Testziel	Worum geht es?	Beispielfrage
Fehler finden	Fehler in Logik und Technik aufdecken, bevor die Benutzer sie finden	Stürzt das Programm bei der Eingabe 0 ab?
Funktionalität sicherstellen	Alle funktionalen Anforderungen sind korrekt umgesetzt.	Wird bei Schenkungen unter 325'000 £ keine Steuer berechnet?
Nicht-funktionale Anforderungen	Leistung, Sicherheit, Benutzbarkeit, Zuverlässigkeit	Verstehen neue Benutzer die Fehlermeldungen ohne Hilfe?
Kompatibilität	Die Software läuft auf allen vorgesehenen Plattformen.	Funktioniert die Seite in Safari auf dem iPhone?
Risiken minimieren	Teure oder gefährliche Fehler vermeiden, z. B. Datenverlust, Sicherheitslücken	Kann über das Eingabefeld Schadcode eingeschleust werden?
Regression vermeiden	Änderungen dürfen Bestehendes nicht kaputt machen.	Funktionieren nach dem Bugfix alle bisherigen Testfälle noch?
Qualität messen	Kennzahlen liefern, um über die Freigabe zu entscheiden	Wie viele Hauptfehler sind noch offen? Wie hoch ist die Abdeckung?
Entwicklung verbessern	Aus den Fehlern lernen	In welchem Modul treten die meisten Fehler auf und warum?

Funktionale und nicht-funktionale Anforderungen

Funktional: *Was* macht das System?

- Berechnung der Steuer
- Prüfung der Eingaben
- Zurücksetzen des Formulars

Nicht-funktional: *Wie gut* macht es das?

- Antwortzeit
- Bedienbarkeit, Barrierefreiheit
- Sicherheit, Stabilität

Beide Arten brauchen eigene Testziele. Nicht-funktionale Anforderungen werden oft vergessen, weil sie selten ausdrücklich verlangt werden.

Zusammengefasst: Testziele machen sichtbar, wozu wir testen. Sie lenken die Arbeit auf das Wichtige und sind die Grundlage für Testfälle, Testintensität und Testende (siehe LU02a - Prüfstrategie).

Kontrollfragen

1. Formulieren Sie die Anforderung «Die Eingabe muss benutzerfreundlich sein» als prüfbares

Testziel.
2. Ein Programm erfüllt die Spezifikation vollständig, die Kundin ist aber unzufrieden. Welche Art von Prüfung wurde vernachlässigt?
3. Nennen Sie für ein Login-Formular je ein funktionales und ein nicht-funktionales Testziel.
4. Warum ist «Regression vermeiden» bei Software, die oft geändert wird, besonders wichtig?

M450-LU02

Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu02/testziele?rev=1794390868

Last update: **2026/11/11 10:54**