

# T 05 Modul 450: Testfall

Testfälle sind das Rückgrat jeder zuverlässigen Software. Sie stellen sicher, dass eine Anwendung korrekt funktioniert, erwartetes Verhalten zeigt und unter unvorhergesehenen Umständen robust bleibt.

## Inhaltsverzeichnis

1. Aufbau eines Testfalls .....	2
2. Testfälle: Ausprägungen .....	3

# 1. Aufbau eines Testfalls

Ein Testfall besteht aus mehreren klar definierten Abschnitten. Diese gewährleisten, dass die Tests wiederholbar, nachvollziehbar und unabhängig voneinander sind.

## 1.1. Struktur eines Testfalls

Testfälle sollten folgende Elemente enthalten:

- **Vorbedingungen:** Was muss erfüllt sein, bevor der Test startet?
- **Schritte:** Welche Aktionen müssen durchgeführt werden?
- **Erwartungen:** Was ist das erwartete Ergebnis?
- **Nachbedingungen:** Wie sollte der Zustand nach dem Test aussehen?
- **Unabhängigkeit:** Testfälle sollten voneinander unabhängig sein.

## 1.2. Vorbedingungen

**Definition:** Die Voraussetzungen, die erfüllt sein müssen, damit der Test durchgeführt werden kann. **Beispiele:**

- Eine Datenbankverbindung muss bestehen.
- Der Benutzer muss sich angemeldet haben.
- Testdaten müssen vorbereitet sein.

## 1.3. Schritte

**Definition:** Die exakten Aktionen, die im Rahmen des Tests durchgeführt werden. **Beispiele:**

1. Öffnen Sie die Applikation.
2. Geben Sie "xyz" in das Suchfeld ein.
3. Klicken Sie auf "Suchen".

## 1.4. Erwartungen

**Definition:** Das erwartete Verhalten oder Ergebnis nach Ausführung der Testschritte. **Beispiele:**

- Eine Erfolgsmeldung wird angezeigt.
- Der Benutzer wird zur Startseite weitergeleitet.
- Ein Fehler wird korrekt abgefangen und gemeldet.

## 1.5. Nachbedingungen

**Definition:** Der Zustand, der nach Abschluss des Tests gegeben sein sollte. **Beispiele:**

- Testdaten werden bereinigt.
- Temporäre Dateien werden gelöscht.
- Der Benutzer wird abgemeldet.

## 2. Testfälle: Ausprägungen

### 2.1. Positivtest: "Happy Flow"

- **Definition:** Testfälle, die den idealen oder erwarteten Ablauf prüfen.
- **Beispiele:**
  - Eine korrekte Anmeldung im System.
  - Eine Suchfunktion liefert erwartete Ergebnisse.
- **Nutzen:** Sicherstellen, dass die Hauptfunktionen wie vorgesehen funktionieren.

### 2.2. Negativtest: "Fehlerprovokation"

- **Definition:** Testfälle, die prüfen, wie das System mit ungültigen oder unerwarteten Eingaben umgeht.
- **Beispiele:**
  - Eingabe von leeren Feldern oder ungültigen Daten.
  - Versuch, auf geschützte Ressourcen ohne Berechtigung zuzugreifen.
- **Nutzen:**
  - Fehleingaben werden korrekt abgefangen.
  - Das System bleibt robust und sicher.