

T 05 Modul 450: Wiederholbare Tests

Tests spielen eine entscheidende Rolle in der Softwareentwicklung. Sie helfen nicht nur, Fehler zu vermeiden, sondern auch sicherzustellen, dass neue Funktionen korrekt implementiert sind und bestehende Funktionalität intakt bleibt. Darüber hinaus dienen sie als Dokumentation und als Werkzeug zum Kennenlernen neuer Bibliotheken oder Schnittstellen.

Inhaltsverzeichnis

1. Gründe zum Testen	2
2. Wiederholbare Tests	4
3. Automatische vs. Manuelle Tests	4
4. Prinzipien wiederholbarer Tests	5

1. Gründe zum Testen

- Fehler vermeiden
- Neue Features testen
- Bestehendes bleibt korrekt
- Gemeldete Fehler nachvollziehen
- Bibliotheken/Schnittstellen kennenlernen

1.1. Fehler vermeiden

Ziele:

- Sicherstellen, dass neuer Code bestehende Funktionalität nicht beeinträchtigt.
- Anforderungen direkt in Tests abbilden, um sie zu validieren.

1.2. Neue Features testen

Ziele:

- Verifizieren, dass neu erstellter Code korrekt funktioniert.
- Nutzen von Methoden wie Test-Driven Development (TDD).

1.3. Bestehendes bleibt korrekt

Thema: Regressionstests

- Ziel: Keine Seiteneffekte durch neue Änderungen.
- **Umfang:**
 - Gesamte Anwendung
 - Bedeutende Module
 - Änderungen, die potenziell betroffen sind
- **Zeitpunkt:**
 - Während der Entwicklung
 - Vor der Auslieferung

1.4. Gemeldete Fehler

Thema: Retests

- Fehler werden durch Kunden gemeldet.
- Ein spezifischer Testfall wird entwickelt, um den Fehler zu reproduzieren und sicherzustellen,

dass er behoben wurde.

- Ziel:
 - Nach Fehlerbehebung testen.
 - Vor der Auslieferung sicherstellen, dass keine Rückfälle auftreten.

1.5. Kennenlerntests

Thema:

- Lernen und Verstehen von neuen Bibliotheken oder Schnittstellen.
- Ziel: Überprüfen, ob und wie die Bibliothek/Schnittstelle in ein Projekt integriert werden kann.

2. Wiederholbare Tests

Wiederholbare Tests zeichnen sich durch ihre Konsistenz und Stabilität aus. Sie liefern bei gleichen Bedingungen immer das gleiche Ergebnis und sind eine Grundvoraussetzung für Automatisierung.

2.1. Wiederholbare Tests

2.1.1. Hauptvorteile

- Geringer Aufwand bei wiederholten Tests.
- Reproduzierbare Ergebnisse.
- Dokumentation und Beweissicherung.
- Basis für Regressionstests und Retests.
- Skalierbarkeit durch Automatisierung.

2.2. Eigenschaften

- **Definierte Testdaten:** Klare und kontrollierte Datenbasis.
- **Isolierung:** Tests hinterlassen keine Seiteneffekte (z. B. Datenleichen).
- **Vorhersagbares Resultat:** Ergebnisse sind deterministisch und stabil.
- **Wartbarkeit:** Leicht an neue Anforderungen anpassbar.
- **Automatisierbarkeit:** Tests können potenziell automatisiert werden.

3. Automatische vs. Manuelle Tests

3.1. Vergleich

Automatisch	Manuell
Toolbasierte Durchführung	Manuelle Durchführung
Schnell und effizient	Langsam und fehleranfällig
Hochgradig wiederholbar	Nicht einfach automatisierbar
Exakt: Penible Ergebnisse	Flüchtigkeitsfehler und potenziell unsauber
Initialkosten hoch	Regelmässige Durchführung teuer

3.2. Automatische Tests

- **Vorteile:**
 - Schnelle und wiederholbare Durchführung.
 - Eignen sich hervorragend für Regressionstests.

- Hohe Exaktheit und Konsistenz.
- **Nachteile:**
 - Hoher Initialaufwand für die Einrichtung.

3.3. Manuelle Tests

- **Vorteile:**
 - Flexible Testfälle, die nicht leicht automatisierbar sind.
 - Entdecken von Fehlern ausserhalb der definierten Anforderungen.
- **Nachteile:**
 - Langsame Durchführung.
 - Höhere Kosten bei regelmässiger Ausführung.
 - Anfälligkeit für Flüchtigkeitsfehler.

4. Prinzipien wiederholbarer Tests

4.1. Grundprinzipien

- **Vollständig:** Tests sollten alle wichtigen Szenarien abdecken.
- **Regression:** Sicherstellen, dass bestehende Funktionalität intakt bleibt.
- **Vergleich:** Ergebnisse sollten mit erwarteten Werten abgeglichen werden.
- **Sicherheit:** Tests sollten robust und zuverlässig sein.
- **Frühzeitigkeit:** Tests früh in der Entwicklung starten.
- **Realismus:** Testfälle sollten reale Szenarien abbilden.
- **Reproduzierbarkeit:** Ergebnisse müssen immer wieder nachvollziehbar sein.

4.2. Best Practices

Die Best Practices bei automatisierten Tests orientiert sich Grundsätzlich an der Softwareentwicklung selbst.

- **FIRST-Prinzipien:**
 - **Fast:** Tests sollten schnell sein.
 - **Independent:** Tests dürfen sich nicht gegenseitig beeinflussen.
 - **Repeatable:** Tests müssen konsistente Ergebnisse liefern.
 - **Self-validating:** Automatische Validierung der Ergebnisse.
 - **Timely:** Tests sollten rechtzeitig erstellt werden.
- **KISS (Keep It Simple, Stupid):**

- Tests sollten einfach und verständlich sein.
- Wartbarkeit und Lesbarkeit sind entscheidend.
- **DRY (Don't Repeat Yourself):**
 - Vermeidung von Duplikationen im Testcode.
 - Wiederverwendbare Komponenten für häufige Szenarien.
- **Red-Green-Refactor:**
 - **Red:** Schreibe einen fehlgeschlagenen Test.
 - **Green:** Implementiere die Funktionalität, um den Test zu bestehen.
 - **Refactor:** Optimierte den Code, ohne Tests zu brechen.