LU11a - Grundlagen

Quelle: https://www.browserstack.com/guide/what-is-test-driven-development

Was ist TDD?

Test Driven Development (TDD) ist ein Entwicklungsvorgehen, dass aus der agilen Entwicklung und dem Extreme programming hervorgegangen ist. TDD ist eine agile Softwareentwicklungsmethode, bei der Tests vor der eigentlichen Implementierung des Codes geschrieben werden. Das Ziel ist, die Entwicklung zu strukturieren, Fehler frühzeitig zu erkennen und eine hohe Codequalität zu gewährleisten.

Dabei werden in einem iterativen Prozess …

- … Unit Tests erstellt
- … Funktionen/Methoden programmiert
- … die Funktionen/Methoden verbessert

bis alle Tests erfolgreich durchgeführt werden.

Grundprinzipien von TDD

Red-Green-Refactor-Zyklus

- Red: Schreibe einen Test für die neue Funktionalität. Der Test schlägt fehl (da es noch keine Implementierung gibt).
- Green: Implementiere den minimalen Code, der erforderlich ist, damit der Test besteht.
- Refactor: Überarbeite den Code, um ihn zu optimieren und die Qualität zu verbessern, während alle Tests weiterhin bestehen.

Kleine Schritte

Entwickle und teste kleine, isolierte Einheiten von Funktionalität nacheinander. Reduziert die Komplexität und erleichtert das Debuggen.

Test-First-Ansatz

Tests werden vor der Implementierung geschrieben, um die Anforderungen zu klären und den Fokus auf das Ziel zu behalten.

Automatisierte Tests

Die Tests sind automatisiert, sodass sie bei jedem Build oder Deployment wiederholt ausgeführt werden können.

Planung

Natürlich musst du sowohl für das Erstellen der Tests als auch für das Programmieren der Funktionen/Methoden ein gutes Verständnis der Anforderungen haben. Ergeben sich im Verlauf der Entwicklung neue Erkenntnisse zu den Anforderungen, so wird der Prozess wiederholt. Die Tests werden ergänzt oder korrigiert und erneut ausgeführt.

Beispiele

Für einen Taschenrechner benötigen wir eine Funktion `add` um Zahlen zu addieren. Zunächst schreiben wir alle relevanten Unit Tests:

- Gültige Addition
- Fehlende / falsche Zahlen
- Extremwerte
- ...

Anschliessend programmieren wir die Funktion `add` und führen alle Unit Tests aus. Solange nicht alle Tests erfolgreich sind, wird die Funktion `add` überarbeitet.

Unit Tests

Jeder Unit Test beim TDD soll nur einen Aspekt prüfen. Wir wollen möglichst kleine Codestücke testen und nicht das Zusammenspiel mehrere Codeteile. Bei Bedarf setzen wir Mockups von Funktionen ein, die nicht Teil des aktuellen Tests sind.

Häufig werden zunächst die Unit Tests für die erfolgreiche Verarbeitung erstellt. Im Verlauf der Entwicklung werden dann weitere Tests für Fehlersituationen oder Extremwerte ergänzt.

M450-LU11, M450-C2I

Marcel Suter

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu11/grundlagen?rev=1794935164

Last update: **2026/11/17 18:06**