LU16a - Behaviour Driven Development

Behaviour Driven Development (BDD) ist ein agiler Ansatz zur Softwareentwicklung, der die Kommunikation zwischen Entwicklern, Testern und Fachexperten verbessert, indem er den Fokus auf das Verhalten eines Systems aus der Perspektive des Endbenutzers legt. Es handelt sich um eine Erweiterung von Test Driven Development (TDD), die Wert auf eine gemeinsame Sprache und Zusammenarbeit legt.

Kernkonzepte von BDD

Fokus auf Verhalten

Anstatt sich auf die technische Umsetzung zu konzentrieren, beschreibt BDD, wie sich ein System in bestimmten Szenarien verhalten sollte. Dabei steht der Nutzen für den Endbenutzer im Vordergrund.

Gemeinsame Sprache (Ubiquitous Language)

BDD fördert die Verwendung einer für alle Beteiligten verständlichen Sprache, oft basierend auf natürlicher Sprache. Diese Sprache wird typischerweise in Form von **Gherkin-Syntax** verwendet, die leicht lesbare Schlüsselwörter wie **Given**, **When** und **Then** enthält.

Zusammenarbeit

Fachexperten, Entwickler und Tester arbeiten eng zusammen, um Anforderungen in Form von Verhaltensspezifikationen zu definieren. Dadurch wird sichergestellt, dass alle dasselbe Verständnis von der Funktionalität des Systems haben.

Die Struktur von BDD-Szenarien

BDD-Szenarien werden oft in **Feature-Dateien** beschrieben, die ein bestimmtes Verhalten eines Systems definieren. Die Szenarien bestehen aus drei Hauptteilen:

1. **Given** – Ausgangssituation oder Vorbedingungen.

Beispiel: *Given the user is logged into their account.*

1. **When** – Eine Aktion, die der Benutzer ausführt.

Beispiel: *When the user clicks on the „Checkout“ button.*

1. **Then** – Das erwartete Ergebnis der Aktion.

Beispiel: *Then the user sees the order confirmation page.*

Ein vollständiges Beispiel könnte so aussehen:

```
Feature: Benutzeranmeldung

  Scenario: Erfolgreiche Anmeldung
    Given der Benutzer ist auf der Login-Seite
    When der Benutzer gibt gültige Anmeldedaten ein
    Then wird der Benutzer auf die Startseite weitergeleitet
```

Vorteile von BDD

- **Klarheit und Präzision**: BDD hilft, Anforderungen klar und verständlich zu formulieren, sodass Missverständnisse reduziert werden.
- **Frühes Feedback**: Durch die enge Zusammenarbeit und den Fokus auf Verhalten werden Probleme frühzeitig erkannt und behoben.
- **Automatisierung und Nachvollziehbarkeit**: Die Szenarien können in automatisierte Tests integriert werden, die sicherstellen, dass das System wie erwartet funktioniert.
- **Benutzerzentrierung**: Der Fokus auf den Endbenutzer stellt sicher, dass die entwickelten Funktionen tatsächlich Mehrwert liefern.

Fazit

Behaviour Driven Development fördert die Zusammenarbeit und Kommunikation zwischen allen Projektbeteiligten, indem es den Fokus auf das Verhalten eines Systems aus Sicht des Benutzers legt. Mit einer gemeinsamen Sprache und leicht verständlichen Szenarien sorgt BDD dafür, dass Software nicht nur funktional, sondern auch nutzerfreundlich entwickelt wird.

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

Permanent link:
https://wiki.bzz.ch/modul/m450/learningunits/lu16/bdd

Last update: **2025/01/20 10:47**