

Fragen, Antworten und Kommentare zur aktuellen Vorlesung

Frage: Sollen in einem Step möglichst viele oder möglichst wenige Informationen enthalten sein.

Antwort: Um flexibel zu sein, ist es sinnvoll möglichst kleine Steps zu haben. Sollte z. B. das Wort „und“ im Text genutzt werden, ist zu prüfen, ob eine Trennung in zwei Steps nicht sinnvoll ist. Da die Steps aber auch lesbar sein sollen, könnte ein „und“ oder eine mit Kommata getrennte Aufzählung trotzdem sinnvoll sein.

Frage: Wie kann ich BDD bei der Entwicklung von graphischen Oberflächen einsetzen?

Antwort: Generell kann mit BDD alle Arten von funktionalen Anforderungen prüfen. Wichtig ist dabei die sinnvolle Umsetzung der Steps. Wir lernen später einige Möglichkeiten GUIs zu testen und dabei auch zu steuern. Dieser Ansatz ist auch für die Umsetzung der Steps für GUIs sehr gut nutzbar.

Einfache nicht-funktionale Anforderungen, wie die Farben und Beschriftungen von Knöpfen sind so auch prüfbar. Generell gilt für BDD wie für andere Ansätze auch, dass sie nicht alleine eingesetzt werden müssen und mit anderen kombinierbar sind.

Sie

Können Sie mir ein Beispiel für BDD geben?

Copilot

das webwird verwendet

Sicher, hier ist ein einfaches Beispiel für Behaviour Driven Development (BDD) mit Cucumber in Java. Angenommen, wir haben eine Anforderung für eine Funktion, die prüft, ob eine Zahl gerade ist.

Zuerst schreiben wir die User Story in einer natürlichen Sprache, die von allen Stakeholdern verstanden wird. Diese wird in einer `.feature`-Datei gespeichert:

```
Feature: Überprüfen, ob eine Zahl gerade ist
  Als Benutzer der "IstGerade"-Funktion
  Möchte ich überprüfen, ob eine Zahl gerade ist
  Damit ich entsprechende Aktionen durchführen kann
```

```
Szenario: Eine gerade Zahl eingeben
  Angenommen die Zahl ist 4
  Wenn ich die "IstGerade"-Funktion aufrufe
  Dann sollte das Ergebnis "true" sein
```

```
Szenario: Eine ungerade Zahl eingeben
  Angenommen die Zahl ist 5
  Wenn ich die "IstGerade"-Funktion aufrufe
  Dann sollte das Ergebnis "false" sein
```

Dann implementieren wir die Schritte in Java mit Cucumber:

Java

```

public class IstGeradeSteps {
    private int zahl;
    private boolean ergebnis;

    @Given("die Zahl ist {int}")
    public void dieZahlIst(int zahl) {
        this.zahl = zahl;
    }

    @When("ich die \"IstGerade\"-Funktion aufrufe")
    public void ichDieIstGeradeFunktionAufrufe() {
        IstGerade istGerade = new IstGerade();
        this.ergebnis = istGerade.pruefen(zahl);
    }

    @Then("sollte das Ergebnis {string} sein")
    public void sollteDasErgebnisSein(String erwartetesErgebnis) {
        assertEquals(Boolean.parseBoolean(erwartetesErgebnis), ergebnis);
    }
}

```

KI-generierter Code. Überprüfen und sorgfältig verwenden. [Weitere Informationen zu häufig gestellten Fragen.](#)

Und schließlich implementieren wir die `IstGerade`-Klasse:

Java

```

public class IstGerade {
    public boolean pruefen(int zahl) {
        return zahl % 2 == 0;
    }
}

```

KI-generierter Code. Überprüfen und sorgfältig verwenden. [Weitere Informationen zu häufig gestellten Fragen.](#)

Dieses Beispiel zeigt, wie BDD verwendet wird, um Anforderungen in Tests zu übersetzen und dann den Code zu implementieren, der diese Tests besteht. Es fördert die Kommunikation zwischen technischen und nicht-technischen Stakeholdern und stellt sicher, dass die entwickelte Software den Geschäftsanforderungen entspricht.