BDD - Brauchst Du Das?

@KatrinRabow

BDD - Behaviour Driven Development

@KatrinRabow

@KatrinRabow

Im letzten Jahrtausend	Ausbildung zur Reiseverkehrs-Kauffrau
2001 – 2016	b2bServices, Griesheim CONSULTING
2015 – 2021	Technische Universität Darmstadt M.Sc. WIRTSCHAFTSINFORMATIK
Seit 2021	msg systems ag, Frankfurt IT CONSULTANT

Wir sind eine international agierende Unternehmensgruppe

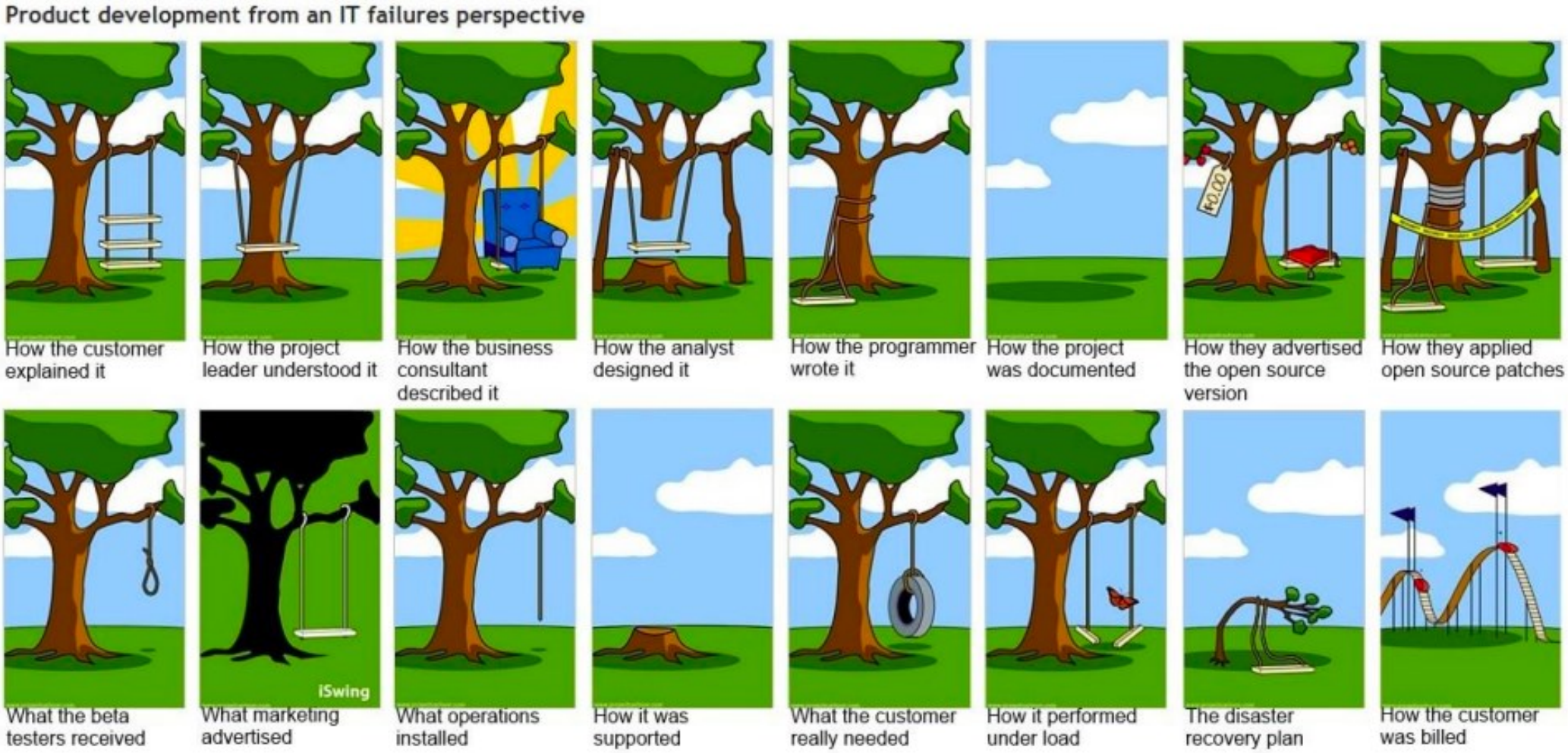
- Gründungsjahr: 1980
- Unternehmenssitz: Ismaning/München
- Umsatz: 1,1 Mrd. € (2020)
- Über 9.000 Mitarbeitende
- Standorte in 28 Ländern
- Branchen: Automotive, Banking, Consumer Products, Food, Healthcare, Insurance, Life Science & Chemicals, Manufacturing, Public Sector, Telecommunications, Travel & Logistics, Utilities
- Starke Unternehmensgruppe aus über 20 starken Marken
- Platz 6 unter den umsatzstärksten IT-Beratungs- und Systemintegrationsunternehmen in Deutschland

Eine Unternehmensgruppe mit starken Marken unter einem Dach vereint

BDD - Behaviour Driven Development

@KatrinRabow

Software Development in a nutshell



Quelle: How Systems Development goes : ProgrammerHumor (reddit.com)

BDD in a nutshell

- Auch ATDD (Acceptance Test Driven Development)
- Entwicklung wird getrieben durch die Frage, wie sich die Software für die Nutzenden verhalten soll
- Stakeholder werden sehr früh in den Entwicklungsprozess miteinbezogen
- Anhand konkreter Beispiele wird das gewünschte Verhalten der Software beschrieben
- Aus der Spezifikation dieser Beispiele ergeben sich die Akzeptanzkriterien für die Umsetzung.

Product Owner:in oder Business Analyst:in

Welches Problem versuchen wir zu lösen?

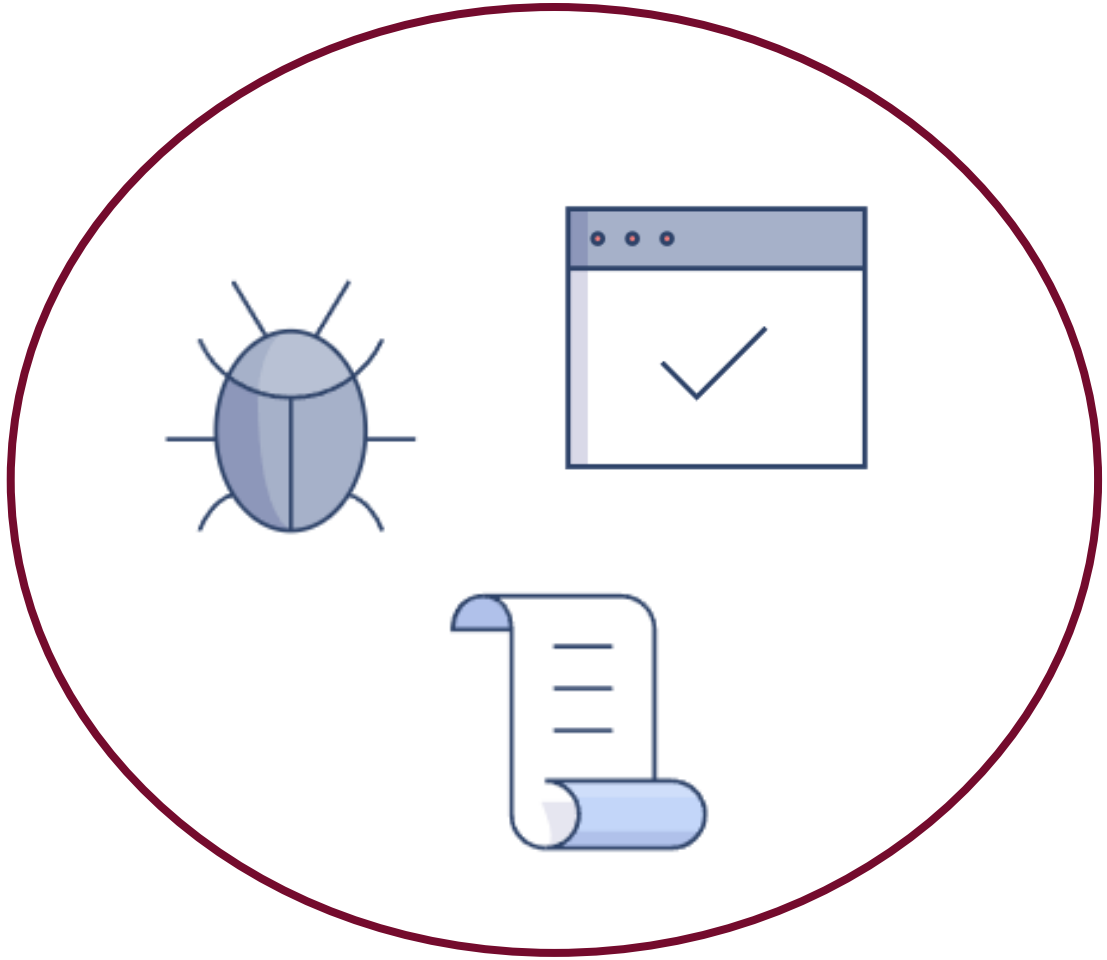
Coder:in

Wie könnte eine Lösung zu diesem Problem aussehen?

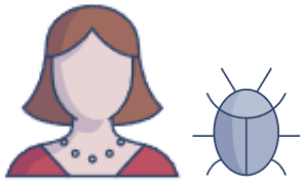
Tester:in

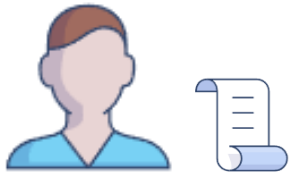
Was würde passieren, wenn…?

Three Amigos Prinzip

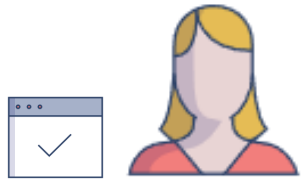


Three Amigos Prinzip

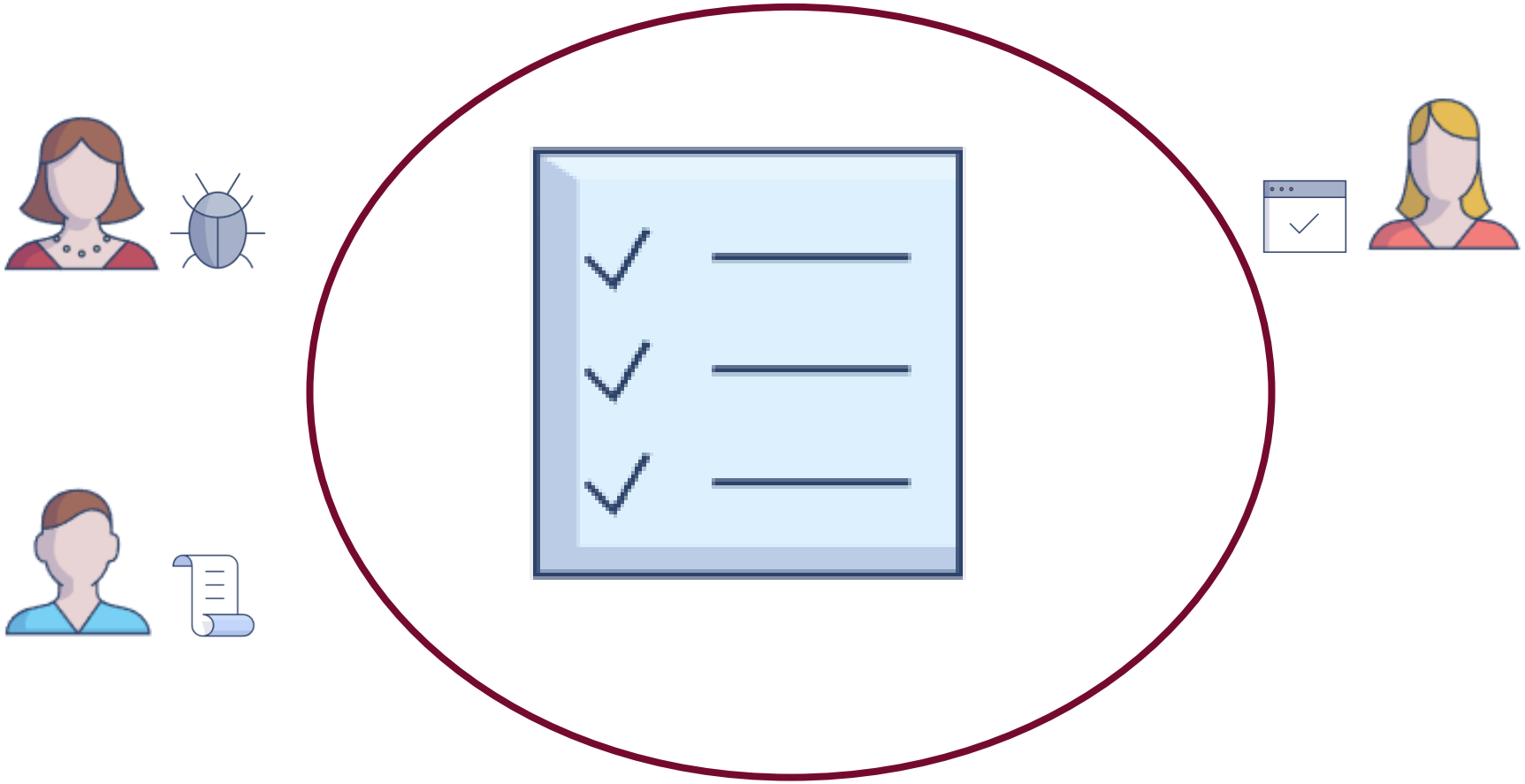




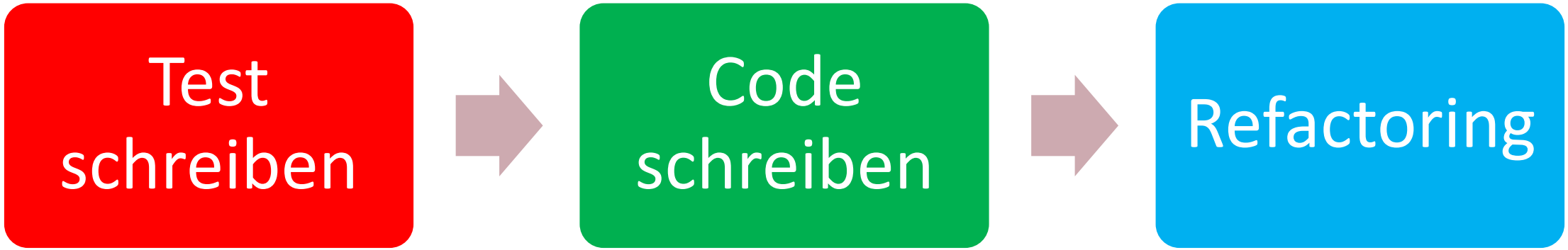
Feature: Formular ausfüllen
Scenario: Ausfüllen
Given Person hat Formular ausgefüllt
When Person das Formular einreicht
Then erhält die Person eine Mail
Scenario: Keine Telefonnummer
Given Person erfasst ihre Daten
And die Person gibt keine TelNr an
When die Person die Daten speichert
Then kommt ein Hinweis
…



Three Amigos Prinzip



Test Driven Development



Beispiel

Gherkin Scenario

Feature: Detaillierte Tests zu den Daten einer Person

Scenario: Name der Person wird erfolgreich eingetragen
Given eine Person wird angelegt
When der Name "Rabow" eingetragen wird
Then wird der Name "Rabow" gespeichert

Scenario: Name der Person wird nicht eingetragen erzeugt Fehlermeldung
Given eine Person wird angelegt
When der Name nicht eingetragen wird
Then wird die Meldung "Bitte geben Sie Ihren Namen an." ausgegeben

Gherkin Scenario

Feature: Detaillierte Tests zu den Daten einer Person

Background:
Given eine Person wird angelegt

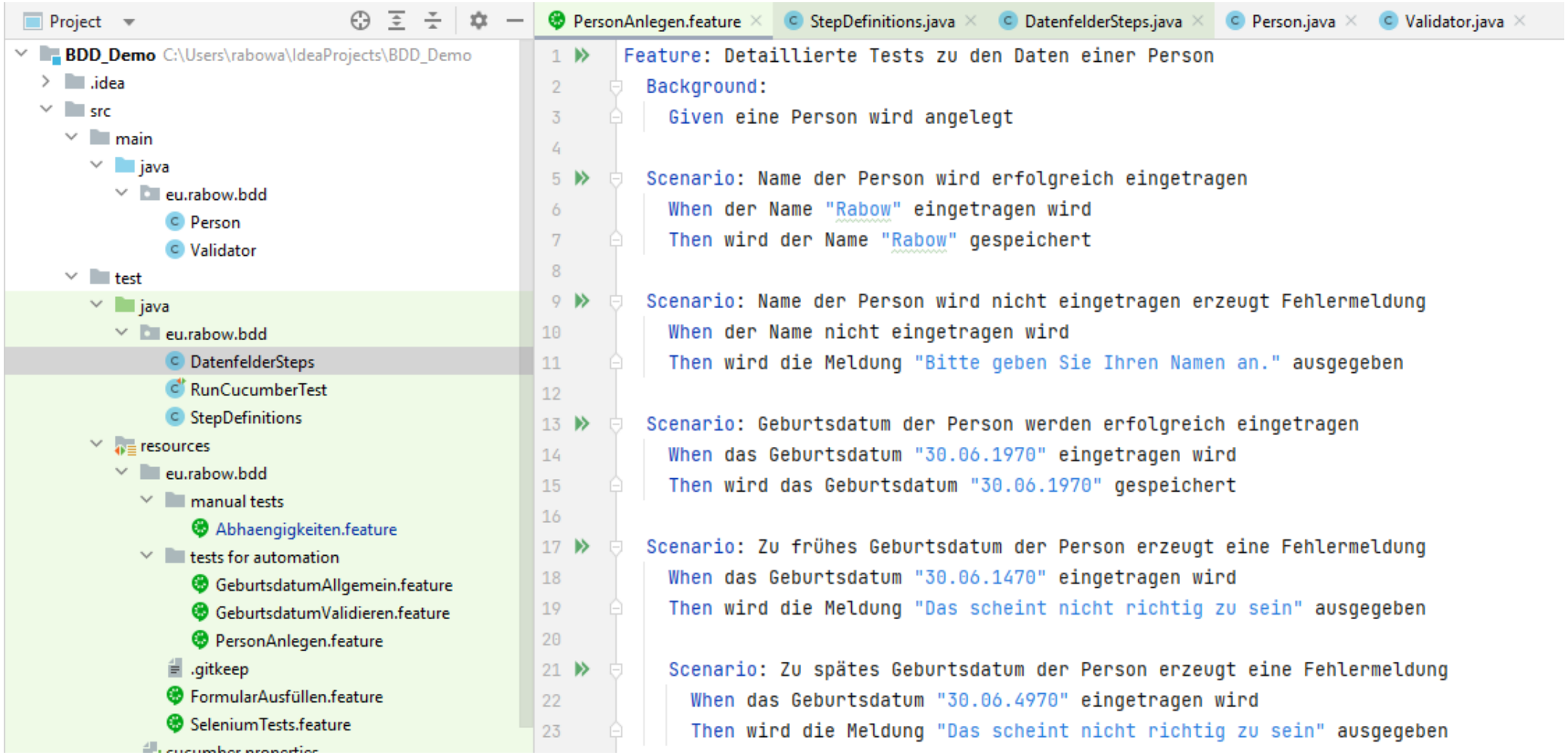
Scenario: Geburtsdatum der Person werden erfolgreich eingetragen
When das Geburtsdatum "30.06.1970" eingetragen wird
Then wird das Geburtsdatum "30.06.1970" gespeichert

Scenario: Zu frühes Geburtsdatum der Person erzeugt eine Fehlermeldung
When das Geburtsdatum "30.06.1470" eingetragen wird
Then wird die Meldung "Das scheint nicht richtig zu sein" ausgegeben

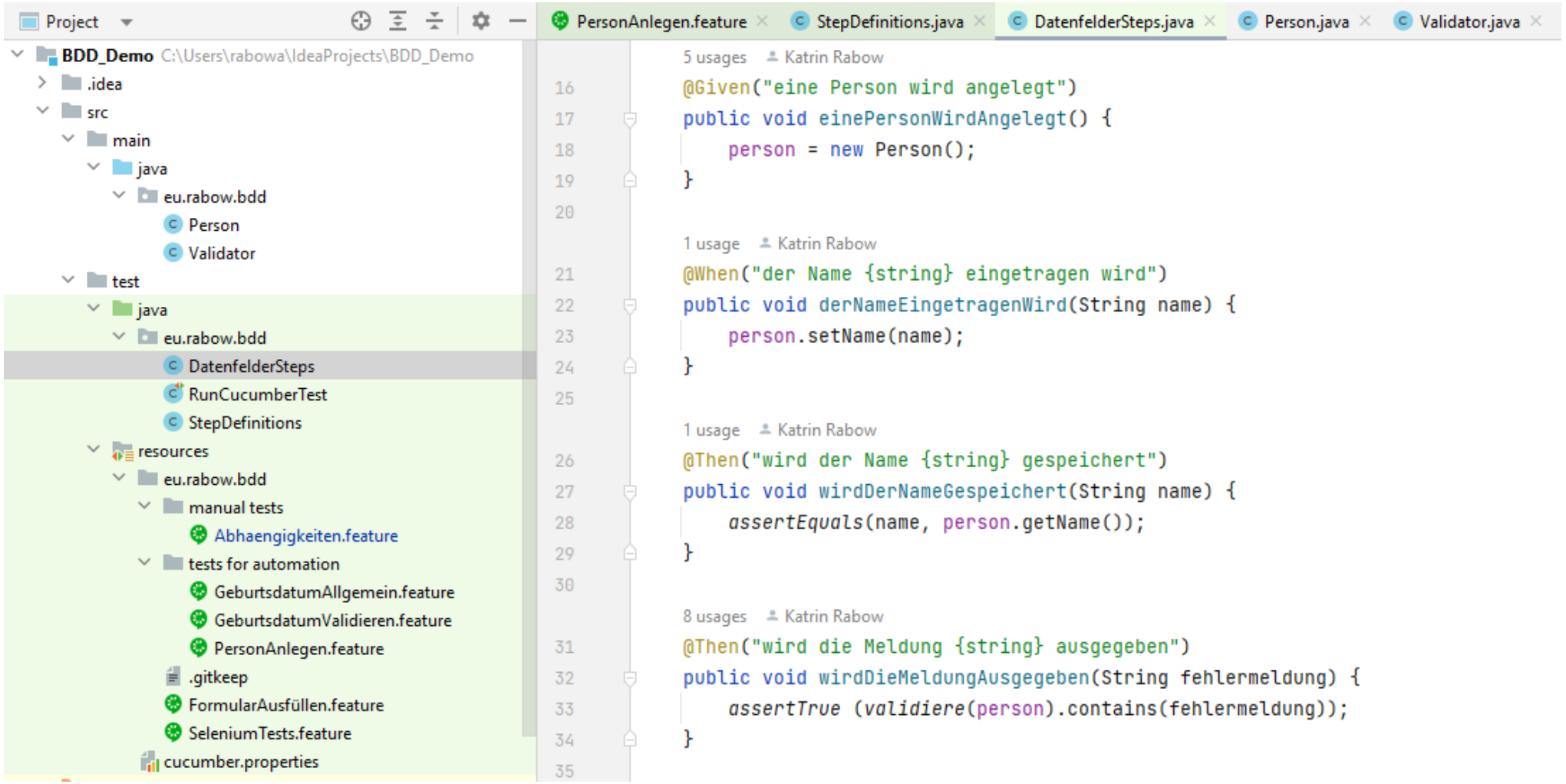
.msg

Demo

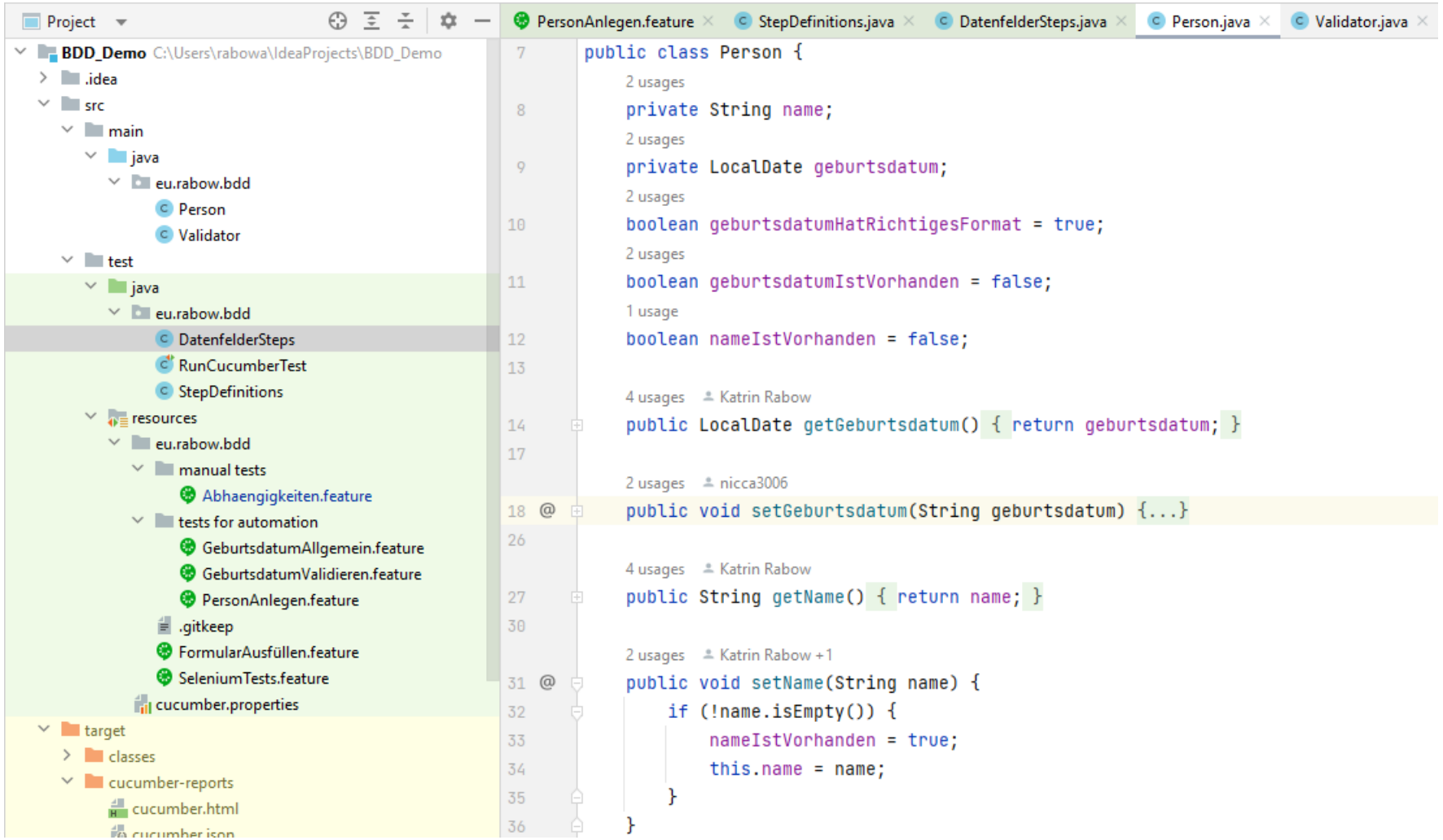
Umsetzung im Code



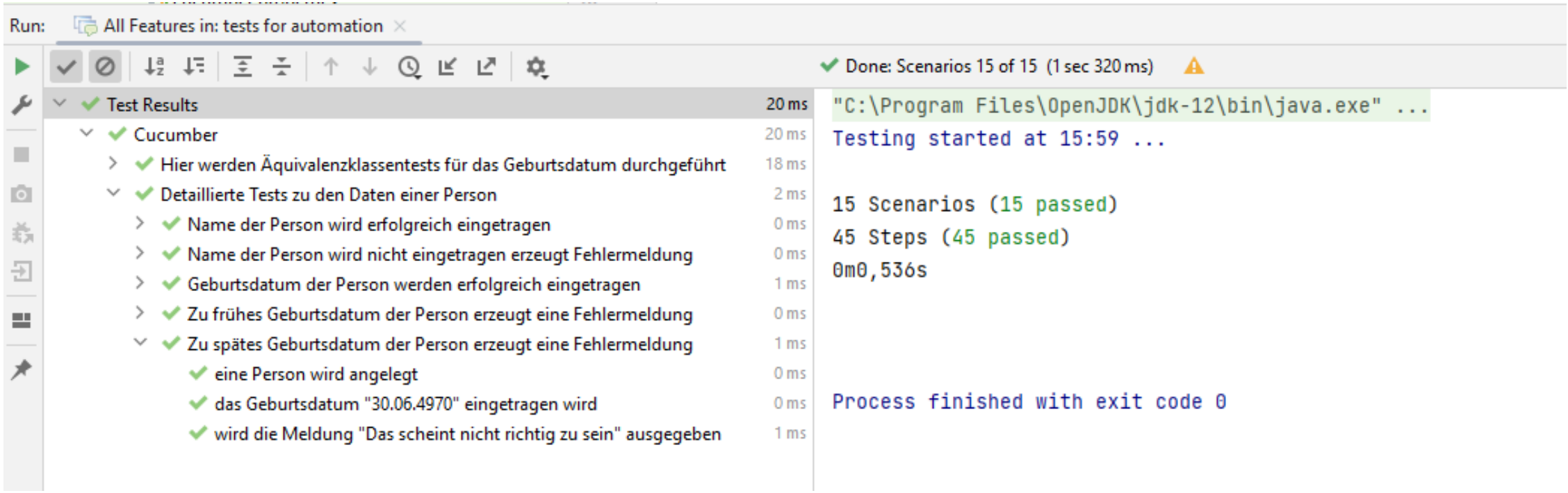
Umsetzung im Code



Umsetzung im Code



Umsetzung im Code



Testergebnis

Feature: Detaillierte Tests zu den Daten einer Person

Background:

- **Given** eine Person wird angelegt

Scenario: Name der Person wird erfolgreich eingetragen

- **When** der Name *"Rabow"* eingetragen wird
- **Then** wird der Name *"Rabow"* gespeichert

Scenario: Name der Person wird nicht eingetragen erzeugt Fehlermeldung

- **When** der Name nicht eingetragen wird
- **Then** wird die Meldung *"Bitte geben Sie Ihren Namen an."* ausgegeben

Scenario: Geburtsdatum der Person werden erfolgreich eingetragen

- **When** das Geburtsdatum *"30.06.1970"* eingetragen wird
- **Then** wird das Geburtsdatum *"30.06.1970"* gespeichert

Scenario: Zu frühes Geburtsdatum der Person erzeugt eine Fehlermeldung

- **When** das Geburtsdatum *"30.06.1470"* eingetragen wird
- **Then** wird die Meldung *"Das scheint nicht richtig zu sein"* ausgegeben

Scenario: Zu spätes Geburtsdatum der Person erzeugt eine Fehlermeldung

- **When** das Geburtsdatum *"30.06.4970"* eingetragen wird
- **Then** wird die Meldung *"Das scheint nicht richtig zu sein"* ausgegeben

Cucumber Report Features Tags Steps

Project	Number	Date
BDD-Demo	1	19 Sep 2022, 16:38

Feature Report

	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Detaillierte Tests zu den Daten einer Person	15	0	0	0	0	15	5	0	5	0.002	Passed

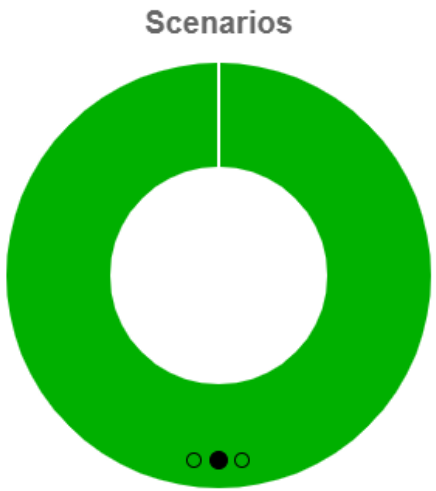
Feature Detaillierte Tests zu den Daten einer Person

- **Background** 0.000
  - **Steps**
    - **Given** eine Person wird angelegt 0.000
- **Scenario** Name der Person wird erfolgreich eingetragen 0.000
  - **Steps**
    - **When** der Name **"Rabow"** eingetragen wird 0.000
    - **Then** wird der Name **"Rabow"** gespeichert 0.000
- **Background** 0.000
  - **Steps**
    - **Given** eine Person wird angelegt 0.000
- **Scenario** Name der Person wird nicht eingetragen erzeugt Fehlermeldung 0.000
  - **Steps**
    - **When** der Name nicht eingetragen wird 0.000
    - **Then** wird die Meldung **"Bitte geben Sie Ihren Namen an."** ausgegeben 0.000

Project	Number	Date
BDD-Demo	1	19 Sep 2022, 16:38

Features Statistics

The following graphs show passing and failing statistics for features



	Steps						Scenarios			Features	
Feature	Passed	Failed	Skipped	Pending	Undefined	Total	Passed	Failed	Total	Duration	Status
Hier werden Äquivalenzklassentests für das Geburtsdatum durchgeführt	30	0	0	0	0	30	10	0	10	0.014	Passed
Detaillierte Tests zu den Daten einer Person	15	0	0	0	0	15	5	0	5	0.002	Passed
	45	**0**	**0**	**0**	**0**	**45**	**15**	**0**	**15**	**0.016**	**2**
	100.00%	**0.00%**	**0.00%**	**0.00%**	**0.00%**		**100.00%**	**0.00%**			**100.00%**

Scenario Outline

```
@Validierung
Feature: Hier werden Äquivalenzklassentests für das Geburtsdatum durchgeführt
Scenario Outline:
  Given eine Person wird angelegt
  When das Geburtsdatum <Wert> eingetragen wird
  Then wird die Meldung <Meldung> ausgegeben
  Examples:
   | Wert | Meldung |
   |"31.12.1849"  | "Das scheint nicht richtig zu sein" |
   | "01.01.2081"  | "Das scheint nicht richtig zu sein" |
   | "15.12.2023" | "Das scheint nicht richtig zu sein" |
   |"00.00.0000"  |"Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "12345678901" |"Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "ABC4567EFG" | "Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "01.--.2000" | "Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "--.01.2000" | "Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "00.12.2000" | "Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
   | "32.01.2000" | "Bitte geben Sie das Datum im Format tt.mm.yyyy an." |
```

Scenario Datenfelder Definitionen

```
#language: de
@nichtTesten
Funktionalität: Allgemeine Regeln zum Data Dictionary - Feld Geburtsdatum

  Szenario: Geburtsdatum der Person ohne exakten Tag der Geburt ist korrekt
    Angenommen eine Person gibt ihr Geburtsdatum ein
    Wenn der Tag der Geburt unbekannt ist
    Dann kann der Tag leer bleiben

  Szenario: Geburtsdatum der Person liegt zwischen 1850 und 2080
    Gegeben sei eine Person gibt ihr Geburtsdatum ein
    Wenn das Datum nach 1850 liegt
    Und das Datum vor 2080 liegt
    Dann wird das Geburtsdatum gespeichert

  Szenario: Geburtsdatum der Person liegt vor heute
    Angenommen eine Person gibt ihr Geburtsdatum ein
    Wenn das Datum vor dem heutigen Tag liegt
    Dann wird das Geburtsdatum gespeichert

  Szenario: Geburtsdatum der Person wird nicht eingetragen
    Angenommen eine Person wird angelegt
    Wenn das Geburtsdatum nicht eingetragen wird
    Dann wird die Meldung "Bitte geben Sie Ihr Geburtsdatum an." ausgegeben
```

Beispiel manueller Test

Die mit * gekennzeichneten Felder sind Pflichtfelder

Step 5

Von den Optionen 1.b. bis 1.d. muss mindestens eine Option ausgewählt sein. Die Felder 1.e. bis 1.h. sind bzgl. Pflichtfeld/optionales Feld von der Auswahl der Option 1.b. bis 1.d. abhängig. Entsprechend dieser Abhängigkeit sind die Felder als Pflichtfelder (mit Sternchen) bzw. optionale Felder (ohne Sternchen) zu kennzeichnen:

- Auswahl "Verein in Gründung":
  - Feld 1.e ist Pflichtfeld
  - In den Feldern 1.f. bis 1.h. sind keine Angaben erforderlich, die Felder 1.f. bis 1.h. sind demzufolge für die Eingabe zu sperren bzw. auszublenden
- Auswahl "Verein, dessen Eintragung ins Register nicht länger als 3 Monate zurückliegt":
  - die Felder 1.e. bis 1.h. zur Erfassung der VR-Daten sind Pflichtfelder
- Auswahl "Verein, dessen Eintragung ins Register vor mehr als 3 Monate erfolgt ist":
  - die Felder 1.e. bis 1.g. sind Pflichtfelder
  - das Feld 1.h. zur Erfassung des Datums der Eintragung ist optional

befindet, die Satzung)"

10. Buttonleiste mit folgenden Buttons:
    a. Abbrechen
    b. Zurück

m. Eingabetextfeld "E-Mail"*
n. Eingabetextfeld "Telefon"
o. Eingabetextfeld "Mobil"

Hinweis f. Devs

Die Felder ~~3. bis 6.~~ 5. bis 8. können für die Eingabe gesperrt bzw. ausgeblendet werden, wenn die Checkbox ~~1.b. in Step 5~~ 3. ausgewählt ist (da in diesem Fall diese Dokumente nicht notwendig sind). Wenn die Checkbox 3. nicht ausgewählt ist, kann das Feld 4. für die Eingabe gesperrt bzw. ausgeblendet werden.

Ist im Step 5 die Option 1.b. ausgewählt, können die Felder 5. bis 8. für die Eingabe gesperrt bzw. ausgeblendet werden. Das Hochladen eines Dokuments in Feld 10. ist optional, d. h. es braucht bei der Prüfung, ob in allen Dokument-Upload-Feldern mindestens ein Dokument hochgeladen ist, nicht berücksichtigt zu werden.

Ist im Step 5 die Option 1.c. ausgewählt, können die Felder 5. bis 8. für die Eingabe gesperrt bzw. ausgeblendet werden.

a. Überschrift "**Gesetzliche Vertretung (1)**"
b. Eingabetextfeld "Familienname"*
c. Eingabetextfeld "Vorname"*
d. Eingabetextfeld "Geburtsname"
e. Eingabe-Datumsfeld "Geburtsdatum"*
f. Eingabetextfeld "Geburtsort"*
g. Eingabetextfeld "Geburtsland"
h. DropDown-Liste "Staatsangehörigkeit(en)"* mit folgenden Elementen
    i. Werte aus Tabelle Staatsangehörigkeit
        (bei erstmaligen Betreten der Maske ist der Wert "deutsch" vorbelegt)
    ii. Eine Mehrfachauswahl von bis zu 5 Elementen ist möglich

i. Infotext "**Anschrift Hauptwohnsitz gesetzliche Vertretung**"
j. Eingabetextfeld "Straße und Hausnummer"*
k. Eingabetextfeld "Postleitzahl"*
l. Eingabetextfeld "Ort"*

iii. Löschen

2. Panel mit
    u. Hinzufügen-Element "Weitere gesetzliche Vertretung hinzufügen" (Damit Antragssteller weitere Vertretende hinzufügen kann)

3. Buttonleiste mit folgenden Buttons:
    u. Abbrechen
    v. Zurück
    w. Weiter

Hinweis f. Devs
Die mit * gekennzeichneten Felder sind Pflichtfelder

Ist im Step 5 die Option 1.d. ausgewählt, können die Felder 1.z. und 1.aa. für die Eingabe gesperrt bzw. ausgeblendet werden.

Gherkin Scenario

Feature: Wenn unser Kunde ein Verein ist, dann wollen wir nicht nur den Namen wissen, sondern auch die Angaben zum Vereinsregistereintrag abfragen. Ist der Verein innerhalb der letzten 3 Monate gegründet worden, dann wollen wir verpflichtend auch noch das Eintragungsdatum ins Vereinsregister kennen.
In Abhängigkeit von diesen Angaben wollen wir bestimmte Unterlagen erhalten.

Background:
Given Nutzende Person befindet sich auf der Seite Verein
And Nutzende Person loggt sich ein
And nutzende Person bearbeitet einen Vorgang

Gherkin Scenario

Scenario: Es handelt sich um einen Verein in Gründung
When Radiobutton VereinInGründung ist aktiviert
Then Eingabetextfeld "Im Vereinsregister eingetragener Name"* ist Pflichtfeld
[…]
And Infotext "Kontaktperson für Rückfragen" wird angezeigt
And Eingabetextfeld “Name"* ist Pflichtfeld
[…]
And Eingabetextfeld “Telefon” ist optionales Feld
And Eingabetextfeld “Mobil" ist optionales Feld
And keine weiteren Eingabefelder sind vorhanden
And Buttonleiste ist vorhanden

Gherkin Scenario

Scenario: Die Eintragung des Vereins war vor weniger als drei Monaten ☑
When Radiobutton VereinMitEintragungWenigerAls3Monate ist aktiviert ☑
Then Eingabetextfeld "Im Vereinsregister eingetragener Name"* ist Pflichtfeld ☑
And Eingabetextfeld "Vereinsregistergericht"* ist Pflichtfeld ☑
And Eingabetextfeld "Vereinsregister-Nummer"* ist Pflichtfeld ☑
And Eingabetextfeld "Datum der Eintragung ins Vereinsregister"* ist Pflichtfeld ☑
[…]
And Infotext "Kontaktperson für Rückfragen" wird angezeigt ☑
[…]
And keine weiteren Eingabefelder sind vorhanden ☑
And Buttonleiste ist vorhanden ☑

.msg

Gherkin & Cucumber

Testfälle schreiben mit Gherkin

Die Beschreibung des gewünschten Verhaltens (Funktionalität und Szenarien) erfolgt in natürlicher Sprache unter Anwendung der jeweiligen Fachsprache (der sogenannten ‚ubiquitären Sprache'). Dabei liegt der Fokus darauf zu formulieren, WAS erreicht werden soll und nicht, WIE es erreicht werden soll. Die Szenarien sollten daher relativ abstrakt beschrieben werden - so präzise wie nötig, nicht so präzise wie möglich**.**

Beispiel (noch nicht formalisiert)

*„Damit die nutzende Person unseren Vorgang ausführen kann, werden ihre persönlichen Daten benötig*t."

Daraus können sich unterschiedliche Szenarien ergeben (Daten liegen bereits vor, Daten müssen noch erfasst werden, …). Um den Übergang zu automatisierten Tests zu erleichtern, werden die Beispiele nun in eine bestimmte Form gebracht:

Testfälle schreiben mit Gherkin

Beispiel (formalisiert)
Funktionalität: Die persönlichen Daten der nutzenden Person werden benötigt. Diese können bereits in einem Konto gespeichert sein oder müssen erst noch erfasst werden.
Als nutzende Person möchte ich meinen Namen und meine Anschrift in einem Konto speichern, um sie immer wieder verwenden zu können.
Szenario: Name und Anschrift hinzufügen
Angenommen die Antragstellung wurde gestartet
Wenn die nutzende Person Name und Adresse eingibt
Dann soll diese dauerhaft gespeichert werden

Szenario: Name und Anschrift aus Konto abrufen
…

Zusammenspiel mit Cucumber

Die natürlichsprachigen Beispiele, die auf diese Weise gemeinsam erarbeitet wurden, können als Textdateien im Projekt anschließend mit Hilfe von Cucumber in den Code übernommen werden.
Cucumber ist ein spezielles Adapter-Framework, welches die einzelnen Szenario-Schritte der Gherkin Feature Files mit der zugehörigen Implementierung der Testautomatisierung verknüpft und ausführt.
Dabei basiert der erste Code auf Klassen und Methoden, die es noch gar nicht gibt, und die z.B. mit Mocks oder Stubs imitiert werden. Im Laufe der Implementierung wird jedes „Angenommen“, jedes Ereignis aus einem „Wenn“ und jedes Ergebnis aus „Dann“ durch eine Klasse repräsentiert, so dass letztlich ein Ende-zu-Ende-Test daraus resultiert.

Zusammenspiel mit Cucumber

Die BDD Beschreibungen können direkt verarbeiten werden, indem jedem Schritt wie „Wenn die nutzende Person Name und Adresse eingibt“ konkrete Benutzeraktionen und Verifikationen zugewiesen werden. Jede Step-Definition-Methode hat einen regulären Ausdruck, der in der zugehörigen Annotation spezifiziert ist. Das Pattern wird benutzt, um die Step-Definition mit allen passenden Schritten zu verknüpfen. Auf diese Weise werden die gemeinsam verfassten Beschreibungen zu einer ‚ausführbaren Spezifikation‘.
Cucumber-Tests lassen sich mit einem speziellen TestRunner für JUnit ausführen. Hierfür legt man eine leere Testklasse an und versieht diese mit der Annotation @RunWith(Cucumber.class). Jedes Szenario kann bei Bedarf auch einzeln getestet werden. Natürlich lassen sich Cucumber-Tests auch im Buildprozess mit Ant oder Maven anstoßen.

.msg

Change Requests

Change Requests

- Geänderte Anforderungen resultieren in geänderten Beispielen / Scenarios
- Geänderte Scenarios resultieren in geänderten Tests
- Geänderte Tests resultieren in geändertem Code
- Dokumentation ist automatisch aktualisiert

Takeaways

Takeaways

- Scenarios müssen lesbar und verständlich sein
- Richtige Granularität finden
- Scenarios gemeinsam schreiben oder mindestens ein Review durchführen
- Nicht alle Scenarios müssen zu (automatisierten) Tests werden
- WICHTIG: Es muss eine gemeinsame Sprache (ubiquitous language) vorhanden sein / gefunden werden

Takeaways

- Fokus liegt auf dem Verhalten der Software im Zusammenspiel zwischen Business und Technik
- Steht am Anfang an der Testgedanke zu stark im Vordergrund, fällt es viel schwerer die Beispiele zu formulieren - Gherkin soll die Beispiele unterstützen, nicht einschränken!
- BDD ist eine Kollaborationsmethode – KEINE Testmethode!

Takeaways

Das Ziel von Behavior Driven Development ist die Zusammenarbeit und Kommunikation zwischen allen Stakeholdern.
Als wertvolles Beiprodukt entsteht eine Testautomatisierung und lebende Dokumentation, die immer den aktuellen Stand der Erwartungen der nutzenden Personen abbildet.

@KatrinRabow

LET'S CONNECT

Quellen

Tools für Acceptance Test-Driven Development (ATDD) | Informatik Aktuell (informatik-aktuell.de)

BDD: Behavior-Driven-Development - Beispiele, Expertenwissen, Best Practices (testing-board.com)

Gurken zum Kaffee (entwickler.de)

GitHub - TNG/JGiven: Behavior-Driven Development in plain Java

Start Testing With BDD and Spock Framework | Pluralsight | Pluralsight

An Introduction to Behavior-Driven Development (BDD) with Cucumber for Java – YouTube