

BDD – den Test von Anfang an mitdenken

sepp.med
Qualität sichert Erfolg

Beatrix Forster



Wer sind wir?

- **Beatrix Forster**

- Tel: +49 9195 931-116
- E-Mail: beatrix.forster@seppmed.de

- **Nguyen Dang**

- E-Mail: quang-nguyen.dang@seppmed.de



Wer sind wir?

- Über 35 Jahre IT-Expertise

- Prozessberatung
- SW-Entwicklung
- QS & Test
- Schulung

- Selbstverständnis

- IT-Dienstleister
- Werkzeughersteller
- Methodenprovider
- Trainingsprovider
- Professional IT-Services



Das Projekt

- System-Level-Test für ein Cloud-basiertes Befundungssystem für medizinische Bilder
- Lösungsansatz BDD

Behavior Driven Development – was ist das?

Product Owner
Entwickler
Tester

BDD konkret

Technisches Setup

- VS Code
- VS Code
- JUnit
- JUnit
- JUnit

- C#
- Visual Studio
- MS Testmanager

Fazit

Automatisierte Testfälle: 168

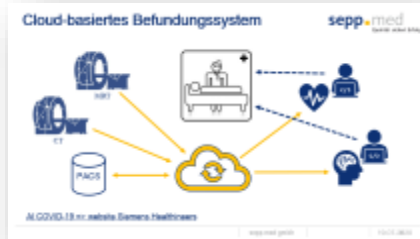
- Fast alle Feature-Tests wurden automatisiert
- Umfassende Testbibliothek
- Automatisches Setup für verschiedene Konfigurationen

Fragen?

- Kontaktieren Sie uns hier auf der Konferenz oder über die Kontaktdaten auf Folie 2.

Das Projekt

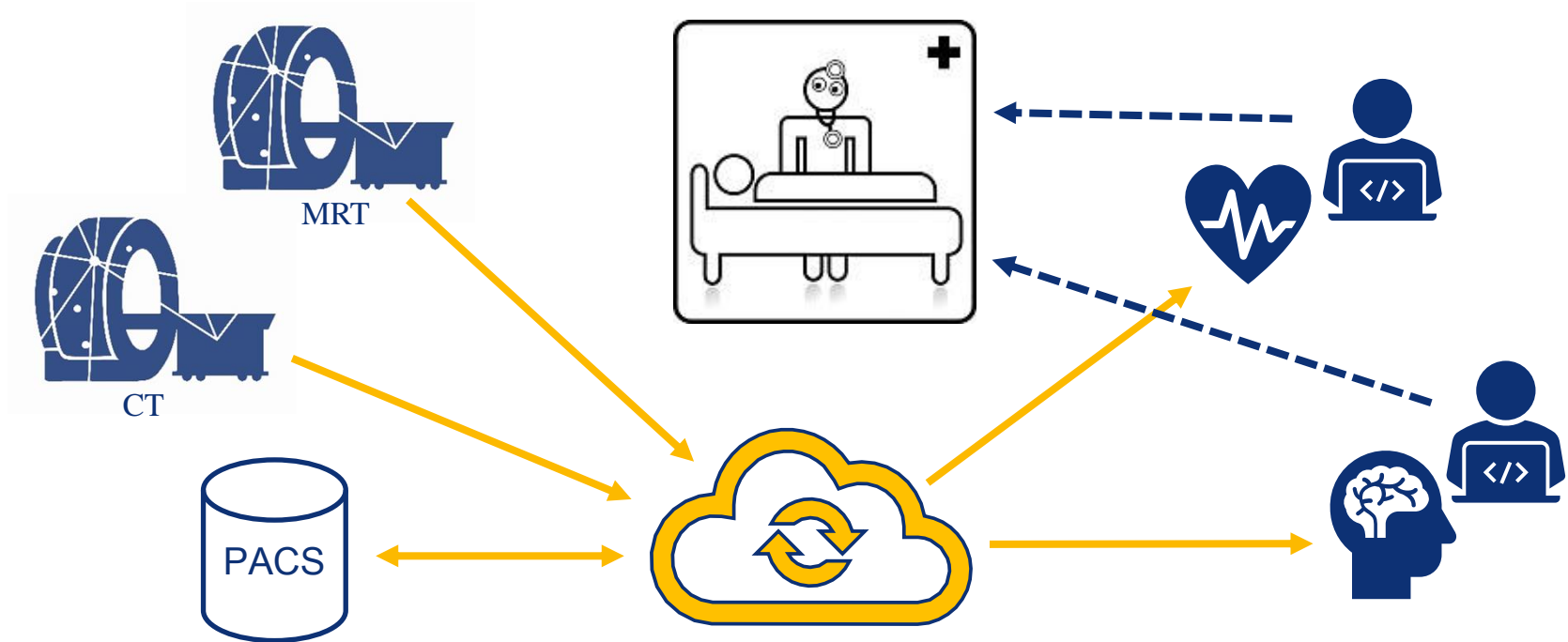
- System-Level-Test für ein Cloud-basiertes Befundungssystem für medizinische Bilder



- Lösungsansatz BDD



Cloud-basiertes Befundungssystem



[AI COVID-19 => website Siemens Healthineers](#)

Ziele

- Kurze Release-Zyklen bei hoher Testabdeckung
 - Hoher Anteil an automatisierten Tests
 - Testfallerstellung parallel zur Codierung
- Umfassender Test verschiedener Konfigurationen
 - Automatisiertes Testsetup für verschiedene Konfigurationen
- Schnelle Reaktion auf Feature-Änderungen
 - Review durch Product Owner
 - Einfache Anpassung von Testszenarien

Behavior Driven Development – was ist das?



Product Owner

Entwickler

Tester

Kollaboration & Automatisierung

- Systemverhalten wird heruntergebrochen auf
 - Einzelne „Features“
 - Abgebildet in konkreten Beispielen „Scenarios“
- Das selbe File ist
 - User Story für den Product Owner
 - Akzeptanzkriterium für den Entwickler
 - Testfall für den Tester
- natürliche Sprache & wohldefinierte Syntax
 - Ermöglicht Automatisierung

Gherkin Syntax

Feature: *feature name*
user story

Scenario: *example 1*
Given *an initial state*
When *an action is taken*
Then *verify an outcome*

Scenario: *example 2*
Given *initial state*
When *another action is taken*
Then *verify another outcome*

Scenario:

- einfaches Englisch
- 1 Feature = 1 User Story
- Scenario
 - Konkretes Beispiel
 - 1 Feature => n Scenarios
- Step
 - Ausführbare Step Definition
 - wiederverwendbar

Kollaboration & Automatisierung

Scenario: *example 1*

Given *an initial state*

When *an action is taken*

Then *verify an outcome*

Ein Szenario ist

- Konkretes Beispiel für den Product Owner
- Akzeptanzkriterium für den Entwickler
- 1 Testfall für den Tester
- Alle arbeiten mit dem selben File und sind auf dem selben Stand

Kollaboration & Automatisierung

- BDD-Framework
 - Erzeugt die Verknüpfung zwischen Step und Step Definition
 - Erzeugt aus den Scenarios ausführbaren Testcode
 - Verwaltet die Steps
- Steps sind
 - Wiederverwendbar
 - Parametrierbar

Kollaboration & Automatisierung II

- Testscenarios stehen für sich allein
 - Unabhängig in beliebiger Reihenfolge ausführbar
- Hooks – ausführbare Scripte
 - before – testrun / feature / scenario
 - Herstellen des Testsetup / Kontext
 - after - wiederherstellen des Ausgangszustands
- @tags - erlauben gezielte Auswahl von Testszenarios
 - Filtern nach Konfigurationen

Nützliche Links

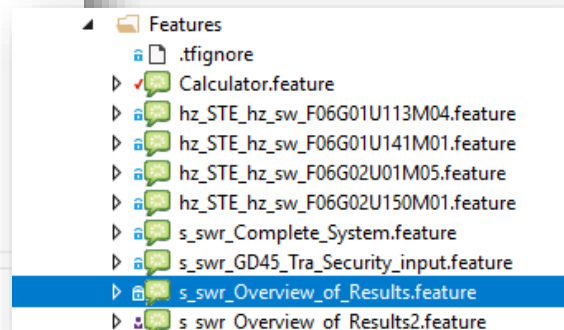
- Einführung in Behavior Driven Development
 - <https://automationpanda.com/bdd/>
- Gherkin Syntax
 - <https://cucumber.io/docs/gherkin/>

BDD konkret

```

1 Feature: s_swir_Overview_of_Results
2 YANSRS_454398
3 Description of the feature ...
4
5 @DefaultCUI
6 @Common
7 Scenario: s_swir_Overview_of_Results_Web_UI
8   Given the configuration parameter are set as follows
9     | conf_param | conf_value |
10    | ALL | DEFAULT |
11    | Lung Lesions: Send for Confirmation | on |
12    | Lung Parenchyma: Send for Confirmation | on |
13    | Lung Lesion: Detection of Lung Lesion | on |
14    | Lung Parenchyma: Detection of Lung Parenchyma | on |
15 When the study of patient AIRC_CS_2089058 [ImagePool-ID: 506] is uploaded through the teamplay Receiver
16 And all results of the study are available in the Confirmation UI
17 And the according study is selected in the Confirmation UI
18 Then the following result series are available in ConfirmationUI
19   | Result |
20   | Lung Parenchyma VRT |
21   | Lung Lesions List |
22   | Lung Lesions VRT |
23   | Sketch |
24 And a sketch image is shown in the Confirmation UI
25
26
27 @DefaultPACS
28 @Common
29 Scenario: s_swir_Overview_of_Results_PACS
30   Given the configuration parameter are set as follows
31     | conf_param | conf_value |
32     | ALL | DEFAULT |
33     | All Algorithms: Send Results to Confirmation | off |
34 When the study of patient ChestAI_Nbg008_S01 [ImagePool-ID: 61] is uploaded through the teamplay Receiver
35 And all results are evaluated and sent to PACS
36 Then the following result series are available in PACS
37   | series |
38   | Overview result image |
39

```



```
1 Feature: s_swr_Overview_of_Results
2     YANSRS_454398
3     Description of the feature ...
```

```
4 @DefaultCUI
```

```
5 @Common
```

```
6 Scenario: s_swr_Overview_of_Results_Web_UI
```

```
7 Given the configuration parameter are set as follows
```

conf_param	conf_value
ALL	DEFAULT
Lung Lesions: Send for Confirmation	on
Lung Parenchyma: Send for Confirmation	on
Lung Lesion: Detection of Lung Lesion	on
Lung Parenchyma: Detection of Lung Parenchyma	on

```
8 When the study of patient AIRC_CS_2089058 [ImagePool-ID: 506] is uploaded through the teamply Receiver
```

```
9 And all results of the study are available in the Confirmation UI
```

```
10 And the according study is selected in the Confirmation UI
```

```
11 Then the following result series are available in ConfirmationUI
```

Result
Lung Parenchyma VRT
Lung Lesions List
Lung Lesions VRT
Sketch

```
12 And a sketch image is shown in the Confirmation UI
```

```
13 @DefaultPACS
```

```
14 @Common
```

```
15 Scenario: s_swr_Overview_of_Results_PACS
```

```
16 Given the configuration parameter are set as follows
```

conf_param	conf_value
ALL	DEFAULT
All Algorithms: Send Results to Confirmation	off

```
17 When the study of patient ChestAI_Nbg008_S01 [ImagePool-ID: 61] is uploaded through the teamply Receiver
```

```
18 And all results are evaluated and sent to PACS
```

```
19 Then the following result series are available in PACS
```

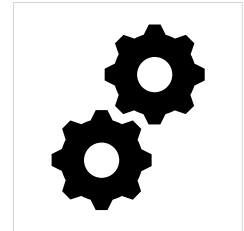
series
Overview result image

Szenario

```

7 | Scenario: s_swr_Overview_of_Results_Web_UI
8 |   Given the configuration parameter are set as follows
9 |       | conf_param | conf_value |
0 |       | ALL | DEFAULT |
1 |       | Lung Lesions: Send for Confirmation | on |
2 |       | Lung Parenchyma: Send for Confirmation | on |
3 |       | Lung Lesion: Detection of Lung Lesion | on |
4 |       | Lung Parenchyma: Detection of Lung Parenchyma | on |
5 |   When the study of patient AIRC_CS_2089058 [ImagePool-ID: 506] is uploaded through the teamplay Receiver
6 |   And all results of the study are available in the Confirmation UI
7 |   And the according study is selected in the Confirmation UI
8 |   Then the following result series are available in ConfirmationUI
9 |       | Result |
0 |       | Lung Parenchyma VRT |
1 |       | Lung Lesions List |
2 |       | Lung Lesions VRT |
3 |       | Sketch |
4 |   And a sketch image is shown in the Confirmation UI

```



```
[When(@"the study of patient (.*) \[ImagePool-ID: (.*)\] is uploaded through the teamplay Receiver")]
```

```
[SuppressMessage("ReSharper", "InconsistentNaming")]
```

2 references | Dang, Nguyen (ext) (SHS DI CT ..., 41 days ago | 2 authors, 3 changes | 2 bugs | 1 work item | 0 exceptions)

```
public void WhenTheStudyOfPatientChestAI_NbgImagePool_IDIsUploadedThroughTheTeamplayReceiver(string patientName, int imagePoolId)
```

```
{  
    Debug.WriteLine(message: "Original Patient name: " + patientName);
```

```
    myScenarioContext.Set(imagePoolId, key: "ImagePool-ID");
```

```
    string reusedPatient = "";
```

```
    string reusedCaseId = "";
```

```
    string configIdentifier = GetConfigIdentifierFromTag();
```

```
    if (myScenarioContext.ContainsKey("reuseExistingCase") && myScenarioContext.Get<bool>(key: "reuseExistingCase"))
```

```
    {
```

```
        return;
```

```
    }
```

```
    //find to reuse the existing case which used the same original patient and has the same configuration
```

```
    if (myScenarioContext.ScenarioInfo.Tags.Contains("DefaultPACS"))
```

```
    {
```

```
        var reusedCases = myCasesRequest.GetCasesContainsNameFromList(patientName, CasesRequest.ListName.InPacs);
```

```
        if (myScenarioContext.ScenarioInfo.Tags.Contains("NoClonedStudy")) reusedCases = reusedCases.Where(x => x.Item2 == patientName
```

```
        else
```


```
        {
```


```
            reusedCases = reusedCases.Where(x => x.Item2.Contains(configIdentifier)).ToList();
```


Technisches Setup

 **MSTest.TestAdapter** by Microsoft
The adapter to discover and execute MSTest Framework based tests.

 **MSTest.TestFramework** by Microsoft
This is MSTest V2, the evolution of Microsoft's Test Framework.

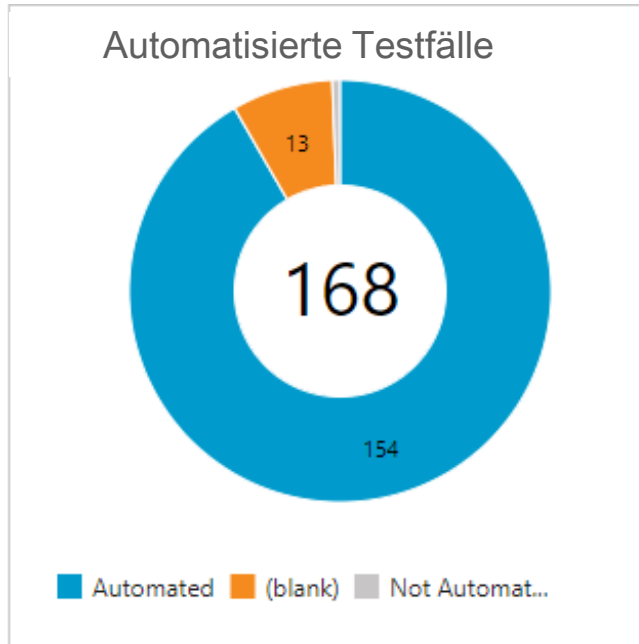
 **SpecFlow** by TechTalk
SpecFlow aims at bridging the communication gap between domain experts and developers by binding business readable behavior specifications to the underlying implementation. Our mission is to provide a pragmatic and frictionless approach to Acceptance Test Driven Development and Behavior Driv...

 **SpecFlow.Tools.MsBuild.Generation** by TechTalk
Package to enable the code-behind file generation during build time <http://specflow.org/documentation/Generate-Tests-from-MsBuild/>

 **SpecSync.AzureDevOps.SpecFlow.2-4** by Spec Solutions
SpecFlow plugin for SpecSync - enables synchronizing automated Azure DevOps test cases

- C#
- Visual Studio
- MS Testmanager

Fazit



- Fast alle Feature-Tests wurden automatisiert
- Umfassende Testbibliothek
- Automatisches Setup für verschiedene Konfigurationen

Ziele

- Kurze Release-Zyklen bei hoher Testabdeckung
 - ✓ Hoher Anteil an automatisierten Tests
 - ✓ Testfallerstellung parallel zur Codierung
- Umfassender Test verschiedener Konfigurationen
 - ✓ Automatisiertes Testsetup für verschiedene Konfigurationen
- Schnelle Reaktion auf Feature-Änderungen
 - ✓ Review der Szenarien durch Product Owner
 - ✓ Einfache Anpassung von Testszenarien

Fazit II

Schwierigkeiten

- Synchronisierung der Arbeit in größerer Gruppe
- Es erfordert viel Disziplin Ordnung und Übersicht zu behalten
- Es dauert, bis eine geeignete Testbibliothek entsteht

Tipps

- Gruppe von 3-4 Personen
- Klein anfangen
- Konzept welche Steps benötigt werden
- Regelmäßig aufräumen

Fazit III

- Release nach jedem Sprint
 - 4 Wochen statt 6 Monate bis zur Freigabe von neuen Features
- Am Ende eines Sprints
 - Sind alle Tests durchgeführt
 - Alle Tests sind vom Product Owner geprüft & freigegeben
 - Existiert ein Test zu jeder Anforderung

Fragen?



- Kontaktieren Sie uns hier auf der Konferenz oder über die Kontaktdaten auf Folie 2.



sepp.med
Qualität sichert Erfolg



Alexander Pocsai



Nguyen Dang

Vielen Dank!

...an meine Kollegen
für ihre Unterstützung

...an Sie
für Ihre Aufmerksamkeit