

# Auf das Zusammenspiel kommt es an

**sepp**.med  
Qualität sichert Erfolg

Beatrix Forster

Product Owner

Entwickler

Tester



## Wer sind wir?

- Über 35 Jahre IT-Expertise
  - Prozessberatung
  - SW-Entwicklung
  - QS & Test
  - Schulung
- Selbstverständnis
  - IT-Dienstleister
  - Werkzeughersteller
  - Methodenprovider
  - Trainingsprovider
  - Professional IT-Services



# Agenda

**Das Projekt**

- System Level Test für ein Cloud-basiertes Befundungssystem für medizinische Bilder
  - AI-Rad Companion von Siemens Healthineers**
  - COVID-19: Postprocessing Applikationen zur Unterstützung der Diagnose von Lungenerkrankungen
- Paradigma: Kurze Release-Zyklen ohne Kompromiss bei der Sicherheit!
- Lösungsansatz: BDD

sepp.med gmbh 08.10.2020 4

**Behavior Driven Development – was ist das?**

Requirement Engineer  
Entwickler  
Tester

sepp.med gmbh 08.10.2020 5

**Zusammenspiel konkret**

Requirement Engineer, Entwickler, Tester

Intro  
Durchführung  
Finale

sepp.med gmbh 08.10.2020 6

**Fazit**

- BDD ist nicht (nur) für Tester
- Requirement Engineer, Entwickler und Tester entwickeln von Anfang an ein gemeinsames Verständnis davon, was entwickelt werden soll, und wie geprüft werden soll, ob das Richtige entwickelt wird.

sepp.med gmbh 08.10.2020 27

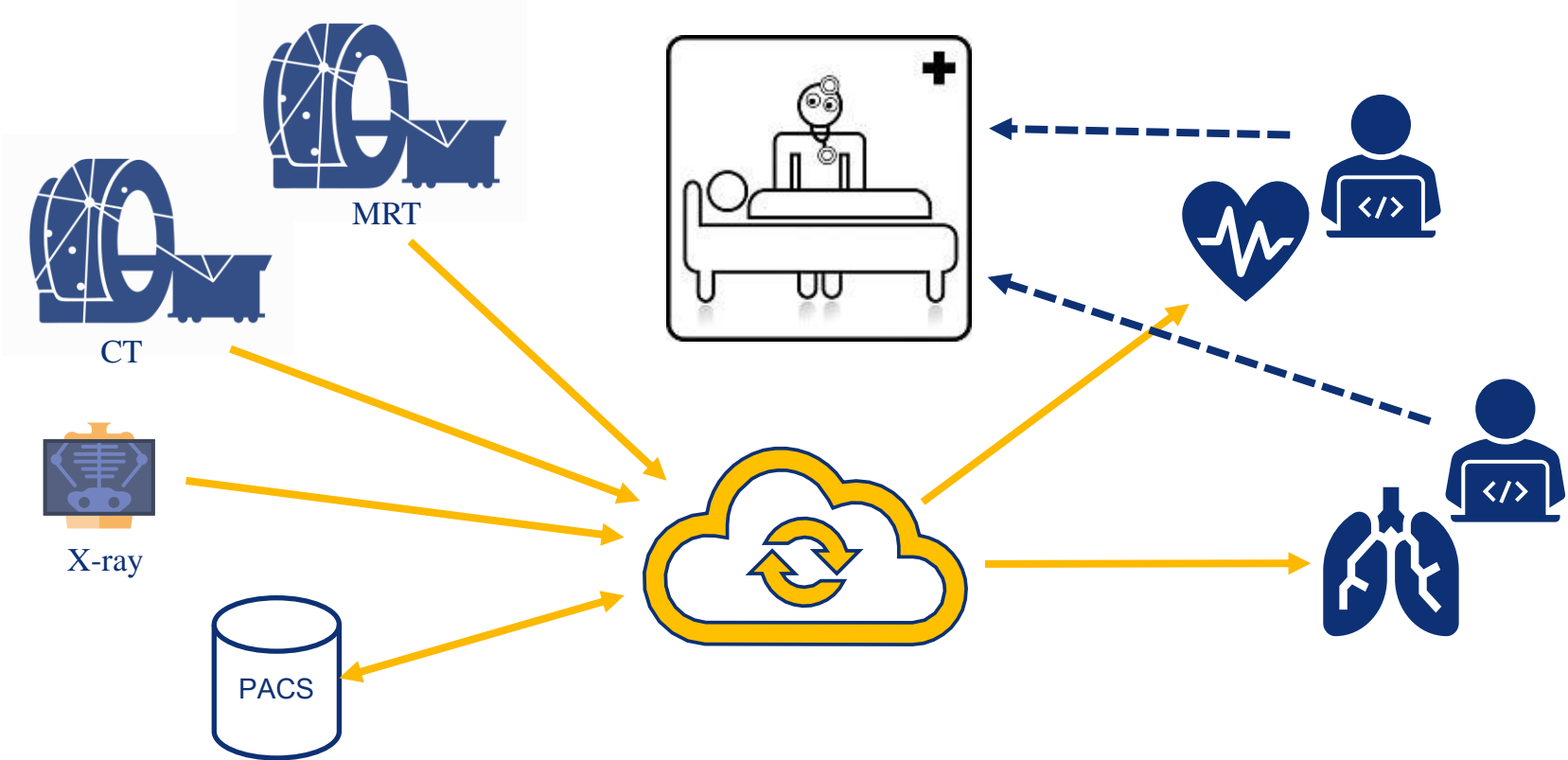
**Fragen?**

sepp.med gmbh 08.10.2020 32

# Das Projekt

- System Level Test für ein Cloud-basiertes Befundungssystem für medizinische Bilder
  - AI-Rad Companion von Siemens Healthineers
  - COVID-19: Postprocessing Applikationen zur Unterstützung der Diagnose von Lungenerkrankungen
- Paradigma:  
Kurze Release-Zyklen ohne Kompromiss bei der Sicherheit!
- Lösungsansatz: BDD

# Cloud-basiertes Befundungssystem



## Ziele

- Frühe Einbindung von Entwicklern und Testern
  - Gemeinsames Verständnis, was entwickelt werden soll
  - Code und Test entstehen gleichzeitig
- hohe Testabdeckung
  - Früh anfangen zu testen
  - Hoher Anteil an automatisierten Tests
- Schnelle Reaktion auf Feature-Änderungen
  - Einfache Anpassung von Testszenarien
  - Review durch den Requirement Engineer

# Behavior Driven Development – was ist das?

A photograph of three violinists in profile, playing their instruments. They are arranged in a line from foreground to background. The background is a solid light blue color. The violinists are wearing dark clothing. The violins are a rich, warm brown color. The bows are light-colored wood.

Requirement Engineer

Entwickler

Tester

## Kollaboration & Automatisierung

- Systemverhalten wird heruntergebrochen auf
  - Einzelne „Features“
  - Abgebildet in konkreten Beispielen „Scenarios“
- Requirement Engineer, Entwickler, Tester
  - Sind von Beginn an bis zur Freigabe eingebunden
  - Sind laufend auf dem selben Stand
- natürliche Sprache & wohldefinierte Syntax
  - Klares Verständnis
  - 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: ....

- Fachliche Sprache
- 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*

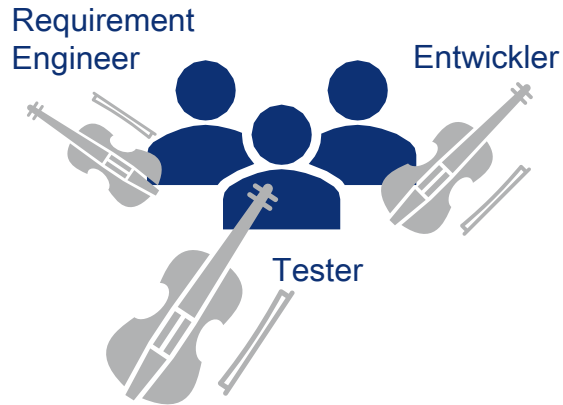
## Ein Szenario ist

- Konkretes Beispiel für den Requirement Engineer
- Akzeptanzkriterium für den Entwickler
- 1 Testfall für den Tester

## 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

# Zusammenspiel konkret



# User Story „Pneumonia Analysis“

## Description

As a [type of user] I want [some goal] so that [some reason]

- Algorithm classifies "opacity" and "high opacity"
- Result Table: Additional (3rd) page, layout still to be defined

- MPR Series:

- opacity regions are highlighted by colored contour. (OPEN different colors for high opacity?)
- OPEN: Lung lobes or left/right lung are highlighted by color contour.
- axial orientation

- VRT Series:

- number of images and quality of rendering can be changed by configuration

- opacity regions are highlighted by color (e.g. red) (OPEN: and the rest of the lung parenchyma in green)

- Configuration model (UI details will follow)

## Feature „MPR Series“

As a user I want to generate the MPR series for Pneumonia Extension

## Intro – Feature wird vorgestellt



- Feature wird vorgestellt
- Erste Acceptance Criteria werden gemeinsam entwickelt
- Ergebnis
  - Gemeinsames Verständnis davon, was entwickelt werden soll und wie geprüft wird, ob das Richtige entwickelt wurde

# Acceptance Criteria

## Acceptance Criteria



13) Pneumonia Assessment series is generated with configured Windowing values in the configuration UI -> Covered in [Unit Test]

Given a dataset

When Windowing values are changed

Then Pneumonia Assessment series is generated with modified Windowing values.

14) Pneumonia MPR images Should Contains Opacity Color Lesions Marked with Red Color--> [System Test]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images Contains Opacity Color Pixels

15) Pneumonia Assessment series is generated with configured "Thickness of Segmentation contour" parameter in the configuration UI--> Covered in [Unit Test]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images generated with configured "Thickness of Segmentation contour"

16) Pneumonia Assessment series is generated with Orientation Cube displayed --> [System Test use case]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images generated with Orientation Cube displayed

# Durchführung

- Entwickler implementiert
  - Acceptance Criteria = vorhandene Szenarien
  - Wöchentlich lauffähiges System
- Tester ergänzt und verfeinert Szenarien
  - Feature File
- Requirements Engineer konkretisiert Anforderungen
- Wöchentliche gemeinsame Durchsprache





# Acceptance Criteria

## Acceptance Criteria



When Pneumonia Assessment series results are not generated.

13) Pneumonia Assessment series is generated with configured Windowing values in the configuration UI -> Covered in [Unit Test]

Given a dataset

When Windowing values are changed

Then Pneumonia Assessment series is generated with modified Windowing values.

14) Pneumonia MPR images Should Contains Opacity Color Lesions Marked with Red Color--> [System Test]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images Contains Opacity Color Pixels

15) Pneumonia Assessment series is generated with configured "Thickness of Segmentation contour" parameter in the configuration UI--> Covered in [Unit Test]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images generated with configured "Thickness of Segmentation contour"

16) Pneumonia Assessment series is generated with Orientation Cube displayed --> [System Test use case]

When Pneumonia extension is finalized with input images

Then The Pneumonia MPR images generated with Orientation Cube displayed

# Acceptance Criteria

14) Pneumonia MPR images Should Contains Opacity Color Lesions Marked with Red Color--> [System Test]  
When Pneumonia extension is finalized with input images  
Then The Pneumonia MPR images Contains Opacity Color Pixels

- Tester erstellt aus Acceptance Criteria
  - Je ein Scenario in Gherkin Syntax
  - = Ausführbarer Testfall

# Scenario / Test Case

Feature: `s_swr_Pneumonia_Results_Axials`

@tc:4295448

Scenario: STE\_s\_swr\_Pneumonia\_Results\_Axials\_\_Pneumonia\_Axial

Given REQUIREKEY `s_swr_Pneumonia_Results_Axials(Lung)`

Given the configuration parameters are set as follows

Parameter	value
ALL	DEFAULT
All: Send Results to Confirmation	off
Pneumonia Assessment: Detection of Pneumonia	on

When the study of patient `Ver_COVID19_Pilsen_4` [ID: `AID_17282000`] is uploaded through the teamplay Receiver

And all results of the study are available in the PACS

Then the following result series are available in PACS

Result
Pneumonia axial MPR

And the name of the 'Pneumonia MPR' series in PACS is the following

Name
AI-Rad Companion Pulmonary Density

And the opacity regions are shown in red color in the Pneumonia axial MPR series in PACS

And pink contours around the regions with high opacity are visible in the Pneumonia axial MPR series in PACS

## Step / Methode

[Given(@"the study of patient (.\*) \[(.\*)\]: (.\*)\] is uploaded through the teamplay Receiver")]

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

2 references | Bayer, Uwe (SHS DI CT R&D ES P... | 1 author, 1 change | 1 incoming change | 1 work item | 0 exceptions, - live

public void WhenTheStudyOfPatientChestAI\_NbgImagePool\_IDIsUploadedThroughTheTeamplayReceiver(string patientName)

```
{
    Debug.WriteLine("Original Patient name: " + patientName);
    myScenarioContext.Set(seriesId, "ImagePool-ID");
    string reusedPatient = "";
    string reusedCaseId = "";

    string configIdentifier = GetConfigIdentifierFromTag();
    if (myScenarioContext.ContainsKey("reuseExistingCase") && myScenarioContext.Get<bool>("reuseExistingCase"))
    {
        return;
    }
}
```



# Requirement

## s\_swr\_Pneumonia\_Results\_Axials

The following segmentation contours are inserted in the axial slices

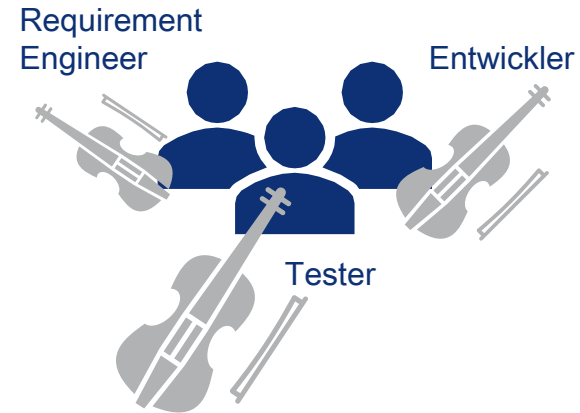
- The 5 lung lobe contours.
- If lung lobes contours detection fails, the contour of the lung is shown to the user.
- Region of opacity shown in red color.
- High opacity regions inside Region of opacity shown in pink color. High opacity regions are the areas where the HU value is -200 or higher (inside the detected opacity regions).

The Graphics are rendered as RGB data in the result images.

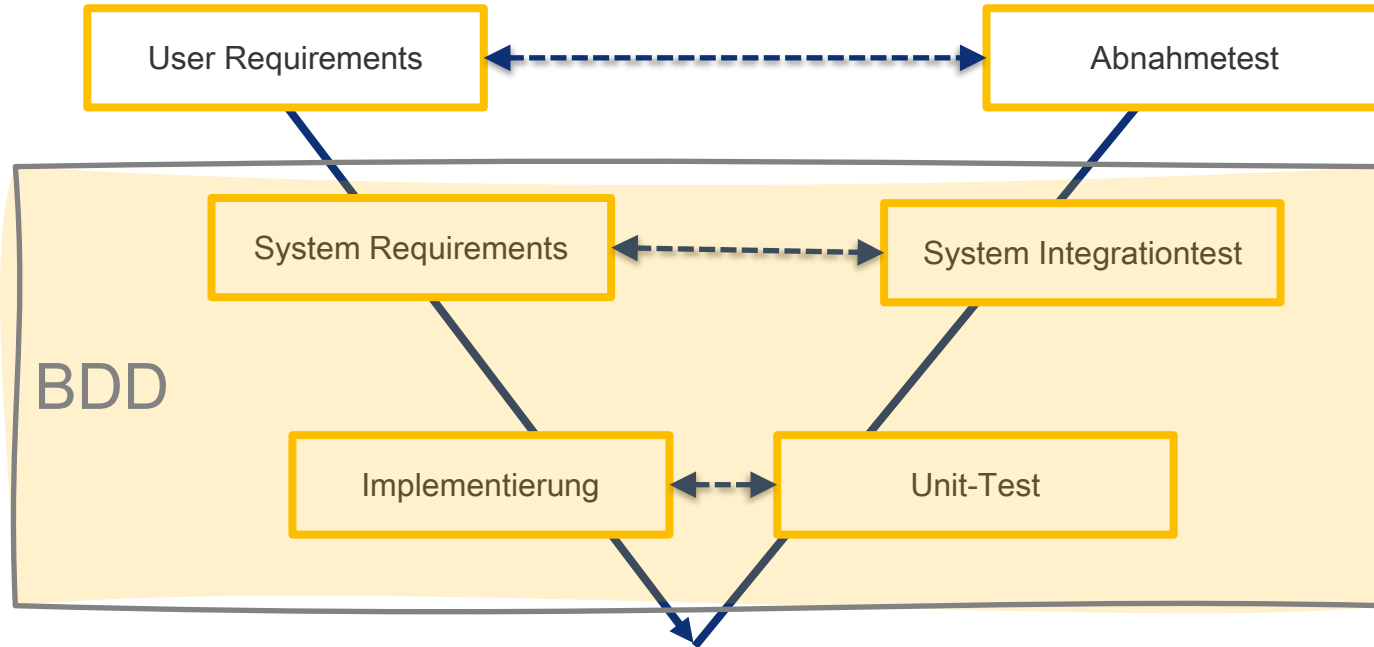
The name of the result series shall be "AI-Rad Companion Pulmonary Density".

# Finale – Abnahme des Features durch das Team

- Implementierung
  - abgeschlossen
- Anforderungen
  - Vollständig
  - Präzise
- Test
  - Vollständig durchgeführt
  - Teilweise automatisiert



# Traceability



## Fazit

- BDD ist nicht (nur) für Tester
- Requirement Engineer, Entwickler und Tester entwickeln von Anfang an ein gemeinsames Verständnis davon, was entwickelt werden soll, und wie geprüft werden soll, ob das Richtige entwickelt wird.





## Ziele

- Frühe Einbindung von Entwicklern und Testern
  - ✓ Gemeinsames Verständnis, was entwickelt werden soll
  - ✓ Code und Test entstehen gleichzeitig
- hohe Testabdeckung
  - ✓ Früh anfangen zu testen
  - ✓ Hoher Anteil an automatisierten Tests
- Schnelle Reaktion auf Feature-Änderungen
  - ✓ Einfache Anpassung von Testszenarien
  - ✓ Review durch den Requirement Engineer

## Warum BDD

- Strenge Syntax
  - Prägnante, klare Requirements
  - Präzise Testfälle
- Gemeinsame Sprache
  - Entwickler versteht die fachliche Anforderung
  - Requirement Engineer kann kontinuierlich mitreden
- Test ist von Beginn an dabei
  - Bessere Klärung
  - Frühzeitige Diskussion

# Tipps

## Problem

- Requirements Engineer ist nicht regelmäßig dabei
- Synchronisierung der Arbeit in größerer Gruppe
- Es erfordert viel Disziplin Ordnung zu halten

## Tipp

- Personaleinsatz anders planen
- Teams von 3 – 5 Personen
- Klein anfangen
- Regelmäßig aufräumen
- Das Vorgehen muss erlernt werden

# Fragen?



# Kontaktdaten

- **Beatrix Forster**
  - Tel: +49 9195 931-116
  - E-Mail: [beatrix.forster@seppmed.de](mailto:beatrix.forster@seppmed.de)





**sepp**.med  
Qualität sichert Erfolg



Nguyen Dang

**Vielen Dank!**

...an meinen Kollegen  
für seine Unterstützung

...an Sie  
für Ihre Aufmerksamkeit

# Auf das Zusammenspiel kommt es an

