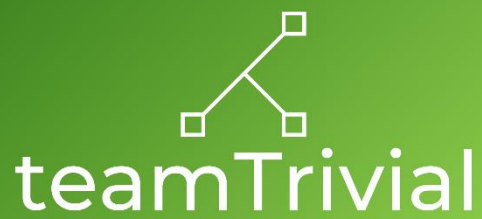


Hochschule für Technik und Wirtschaft Berlin

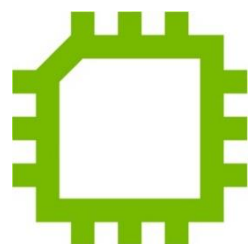
# Sprint Review & Planning

## 16.12.2020

Projekt Softwareentwicklung - Computer Engineering - 6. Semester



Frank Schütze, Konrad Münch, Philipp Horländer, Lukas Poeppel  
16.12.2020



## Inhalt

Abbildungsverzeichnis.....	1
Intro.....	2
Sprint 16.12.2020: .....	2
Aufgaben für den 16.12.2020.....	2
Eigene Ziele für den 16.12.2020.....	3
Kommentare der Dozenten/Kunden.....	4
Extras zum Sprint.....	4
Sprint Planning (16.12.2020 - 06.01.2021).....	5
Eigene Ziele(S.M.A.R.T) .....	5
Ausgewählte Aufgaben .....	5
Aufgaben .....	5
Kommentare der Dozenten / AG.....	5

## Abbildungsverzeichnis

Abbildung 1: Jenkins Pipeline .....	2
Abbildung 2: Bidirektionale Funktionalität .....	3
Abbildung 3: Sonar .....	3
Abbildung 4: Diagramm Baudauer .....	3

## Intro

Das folgende Sprintprotokoll dient dazu die erreichten Ziele zu erläutern und definiert Vorgehen, Einteilungen, Arbeitsweise sowie Arbeitspakete.

Diese Sprint Reviews sollen zwischen den Auftraggebern Thorsten Trzewik und Robert Magnus im folgenden AG genannt und den Auftragnehmern Frank Schütze, Konrad Münch, Philipp Horländer und Lukas Poeppel im folgenden TT (teamTrivial) genannt, abgehalten werden.

Diese Reviews finden in einem 14-tägigen Intervall statt, somit soll eine regelmäßige Kommunikation gewährleistet werden.

## Sprint 16.12.2020:

Teilnehmer 16.12.2020 anwesend:

- AG
- TT

## Aufgaben für den 16.12.2020

1.) Alle 5 Statistiken anzeigen

→ **Erfolgt:** Letzte Statistik durchschnittlicher Bauzeitraum zu 100% implementiert.

2.) Tests erweitern (mehr Tests schreiben)

→ **teilweise Erfolgt:** Testabdeckung *Backend* : 100% , *Frontend* : 56.5%

3.) Sonar:

- Tripple-A ("Bugs", "Vulnerabilities" und "Code Smells" sollen jeweils mit "A" bewertet werden)

→ **Erfolgt:** Tripple-A im *Backend* und *Frontend*.

- Keinen Third-Party-Code in die Analyse einbeziehen

→ **Erfolgt:** Third-Party-Code wurde von der Analyse ausgeschlossen

4.) Automatischer Jenkins-Trigger

→ **Erfolgt:** Seit Sprint Review vom 02.12.2020

### Stage View

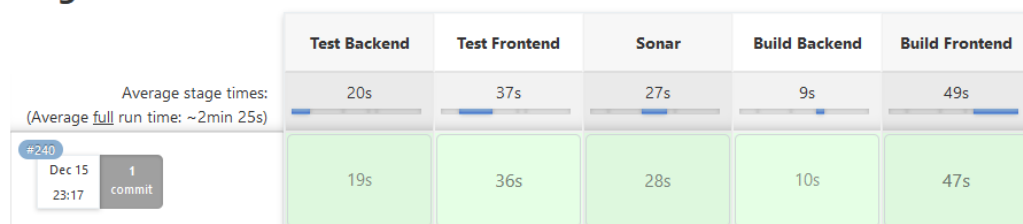


Abbildung 1: Jenkins Pipeline

## Eigene Ziele für den 16.12.2020

### 1.) Frontend : Verbindung der Übersichtstabelle und Kartenansicht herstellen:

- Ein angewählter Marker soll dementsprechenden Tabelleneintrag markieren → **Erfolgt**
- Ein angewählter Tabelleneintrag soll den dazugehörigen Marker markieren → **Erfolgt**
- Datumsauswahl soll auch für die Tabelle angewandt werden → **Erfolgt**

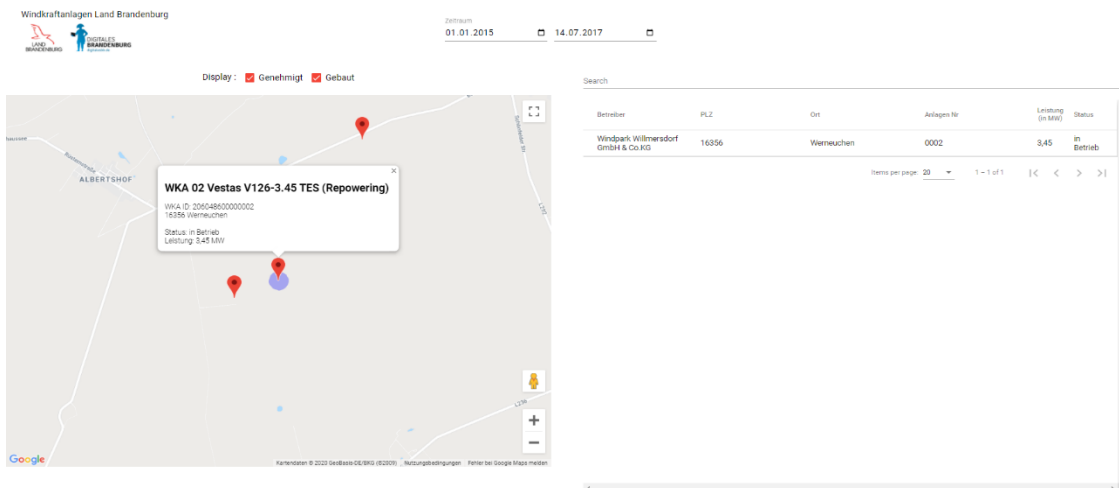


Abbildung 2: Bidirektionale Funktionalität

### 2.) Testabdeckung von mind. 70% Code Coverage in Sonar → **Erfolgt** im Backend

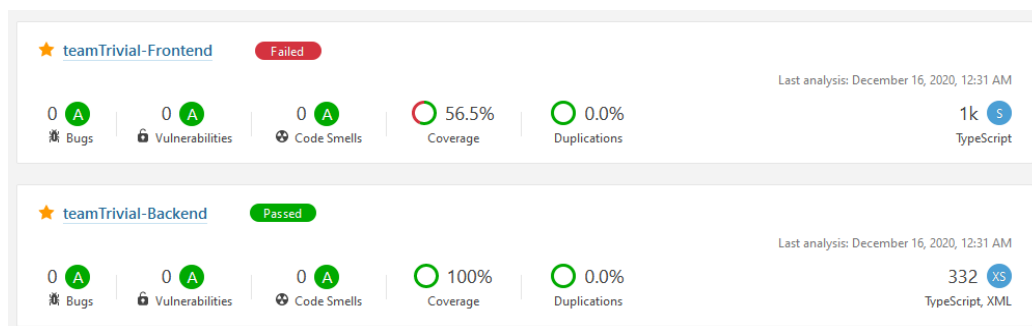


Abbildung 3: Sonar

### 3.) API für die dynamische Anzeige der Baudauerentwicklung der WKA's im gewählten Zeitraum implementieren. Daten im Frontend durch ein Liniendiagramm anzeigen lassen → **Erfolgt**

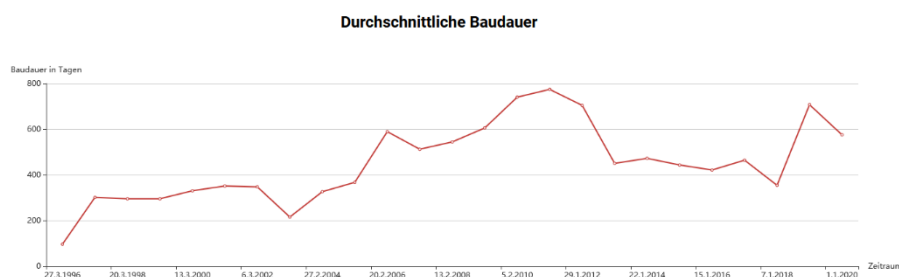


Abbildung 4: Diagramm Baudauer

4.) Dokumentation der vorhandenen APIs im Wiki → **Erfolgt**

### Kommentare der Dozenten/Kunden

1.) Mouseover Tooltips für das Balkendiagramm (Top 10 Postleitzahlen) hinzufügen → **Erfolgt**

2.) Überschrift auf der Startseite hinzufügen → **Erfolgt**

3.) Bug beheben, welcher von Sonar referenziert wurde → **Erfolgt**

4.) Fragen bezüglich Sonar als E-Mail formulieren → **Erfolgt**

### Extras zum Sprint

1.) Trennung der Sonar Projekte in *Backend* und *Frontend* (siehe Abbildung 3)

2.) Alle Backend-Tests befinden sich nun in einem Verzeichnis

3.) Refactoring und Rework des gesamten Backend Codes

- jede API in eigene Dateien ausgelagert
- häufig verwendete Unterfunktionen in *helper.ts* ausgelagert
- *server.ts* von über 150 auf 40 Zeilen optimiert

→ Backend Code von **526** Zeilen Code auf **332** optimiert ( Sonar war hilfreich )

- Variablen- und Funktionsnamen eindeutiger
- Kommentierung der Funktionen im Doxygen-Format vorgenommen

→ Lesbarkeit deutlich erhöht (Clean Code)

4.) Guide zur Projekt-Kompilierung im WIKI verbessert

## Sprint Planning (16.12.2020 - 06.01.2021)

### Eigene Ziele(S.M.A.R.T)

- Testabdeckung des Frontend auf mind. 60 % anheben um den Status „PASSED“ bei Sonar zu erreichen
- Ladebildschirm hinzufügen (Waiting Spinner)
- API-Schaltfläche durch Impressum ersetzen und Inhalt dort anzeigen.
- Ursachen für Performance-Einbußen näher untersuchen um Geschwindigkeit zu optimieren.

### Ausgewählte Aufgaben

- Testabdeckung auf 75%
- Welche Lizenzen werden bei Ihnen verwendet?

### Aufgaben

- 1.) Bild für System- und Komponenten-Architektur
- 2.) Links für Sonar
  - Continuous integration
  - Sources
- 3.) Jenkins-Jobs für das Starten und Stoppen der Anwendung
  1. für MSGA,RTFM, teamTrivial, 404Bandits: Die Anwendung muss dafür eine Authentisierung erfordern
  2. Für die Teams, die schon einen eigenen Server haben: Bitte extra (manuell zu triggernde) Deployment-Jobs anlegen, der das aktuelle Artefakte (Backend/Frontend) auf dem Server aktualisiert.
- 4.) Zusatz (MSGA,RTFM, teamTrivial, 404Bandits). Hier sind **zwei** von den **drei** Punkten zu erledigen. Die tragen sie die ausgewählten Punkte mit in ihr Planing ein.
  - 1.) Testabdeckung auf 75%
  - 2.) Welche Lizenzen werden bei Ihnen verwendet?
  - 3.) Prüfung auf Barrierefreiheit

### Kommentare der Dozenten / AG

Dozenten / AG hatten keine Wünsche für Änderungen geäußert.