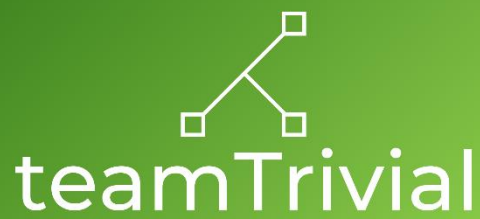


Hochschule für Technik und Wirtschaft Berlin

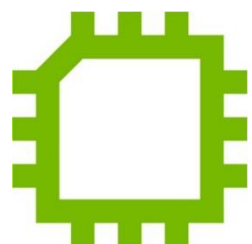
Sprint Review & Planning

18.11.2020

Projekt Softwareentwicklung - Computer Engineering - 6. Semester



Frank Schütze, Konrad Münch, Philipp Horländer, Lukas Poeppel
18.11.2020



Inhalt

Abbildungsverzeichnis.....	1
Intro.....	2
Sprint 18.11.2020:.....	2
Aufgaben für den 18.11.2020.....	2
Eigene Ziele für den 18.11.2020.....	2
Fragen vom 4.11.2020.....	2
Grüner Jenkins Job	3
Durchstich.....	3
Datenanalyse.....	5
Kostenabschätzung.....	5
Analyse der Statistik-Anforderungen	6
Sprint Planning (18.11.2020 – 2.12.2020).....	7
Eigene Ziele.....	7
Aufgaben	7
Kommentare der Dozenten.....	7

Abbildungsverzeichnis

Abbildung 1: Grüner Jenkins Job	3
Abbildung 2: Neues funktionales Frontend mit Google Maps, Pin Clustern und Tabelle.....	3
Abbildung 3: Suchfunktion nach Ort und automatischer Ausschnitt auf der Karte.....	4
Abbildung 4: Details einer Gewählten WKA (blau) auf der Karte und deren Details in der Tabelle.....	4
Abbildung 5: Preistabelle für Google Maps Javascript API.....	5

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.

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

Sprint 18.11.2020:

Teilnehmer 18.11.2020 anwesend:

- AG
- TT (Abwesend: Frank Schütze, Grund: Internetstörung)

Aufgaben für den 18.11.2020

1. grüner Jenkins-Job (der Ihre Anwendung baut...)
2. Durchstich (erste Anzeige von Daten auf der Web-Oberfläche)
3. Datenanalyse der DB (Qualität der Daten)
4. Kostenabschätzung bei kostenpflichtigen GEO-Mappings
5. Analyse der Statistik-Anforderungen (KUS...)
6. SMARTe Planung

Eigene Ziele für den 18.11.2020

1. Funktionierendes Frontend in master mergen
➔ erfolgt
2. Sonar und Jenkins eingerichtet
➔ Jenkins siehe Grüner Jenkins Job
➔ Sonar nicht erfolgt
3. Datenbankanbindung
➔ Mit dem Durchstich bereits erfolgt (4.11.2020)

Fragen vom 4.11.2020

1. Was ist, wenn Angular LTS ausläuft?
 - Genutzt wird Angular 10.0 (nicht AngularJS). Daher ist das Ende des LTS nicht gravierend. Angular 9.x und neuer nutzt standardmäßig die *Ivy Pipeline* und ist somit auf die neue Version Angular 11.0 vorbereitet. Ein Upgrade sollte ohne Probleme erfolgen.¹
 - Für AngularJS ist ein endgültiges Ende durch Google vorgesehen

¹ <https://angular.io/guide/ivy> (14.11.2020)



Branches (1)			
S	W	Name ↓	Letzter Erfolg
		master	26 Sekunden - #6

Abbildung 1: Grüner Jenkins Job

Durchstich

Bereits im letzten Sprint erreicht (4.11.2020)

Neue Version des Frontend mit Karte und Tabelle jetzt vorhanden (s. Abbildung 2, Abbildung 3, Abbildung 4) und in den master branch gemerged (Eigene Ziele für den 18.11.2020, 2.).

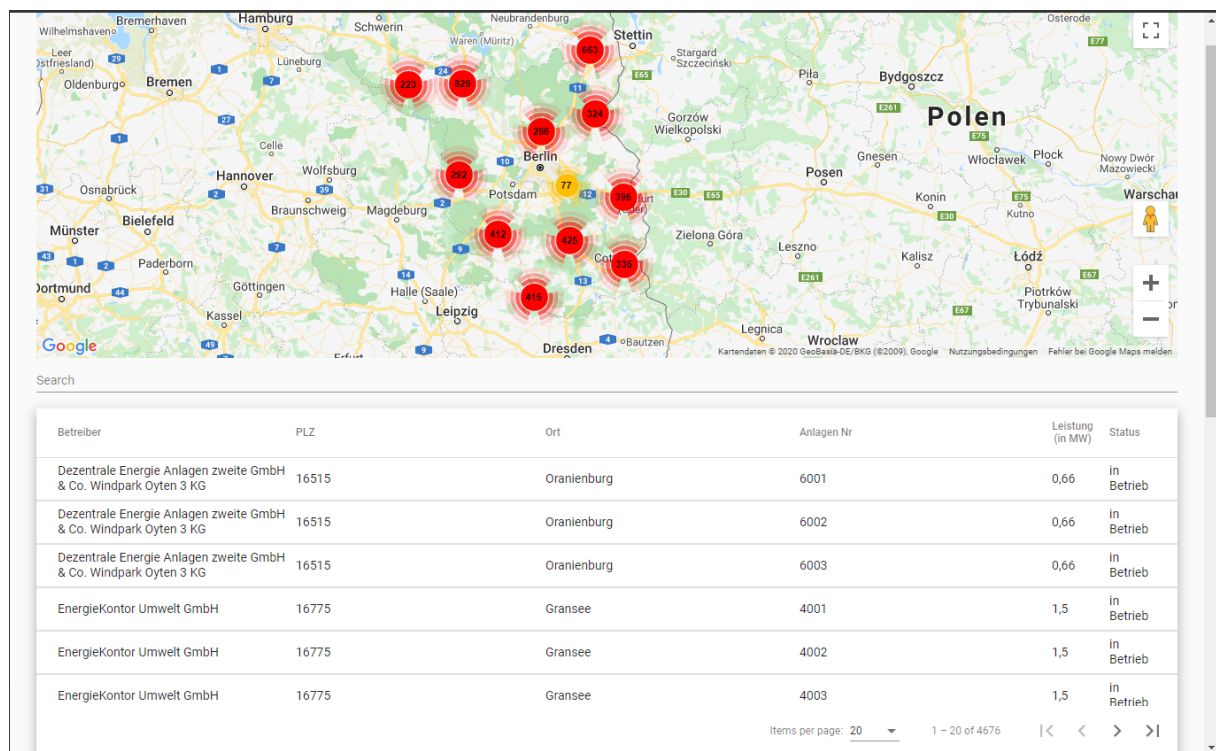


Abbildung 2: Neues funktionales Frontend mit Google Maps, Pin Clustern und Tabelle

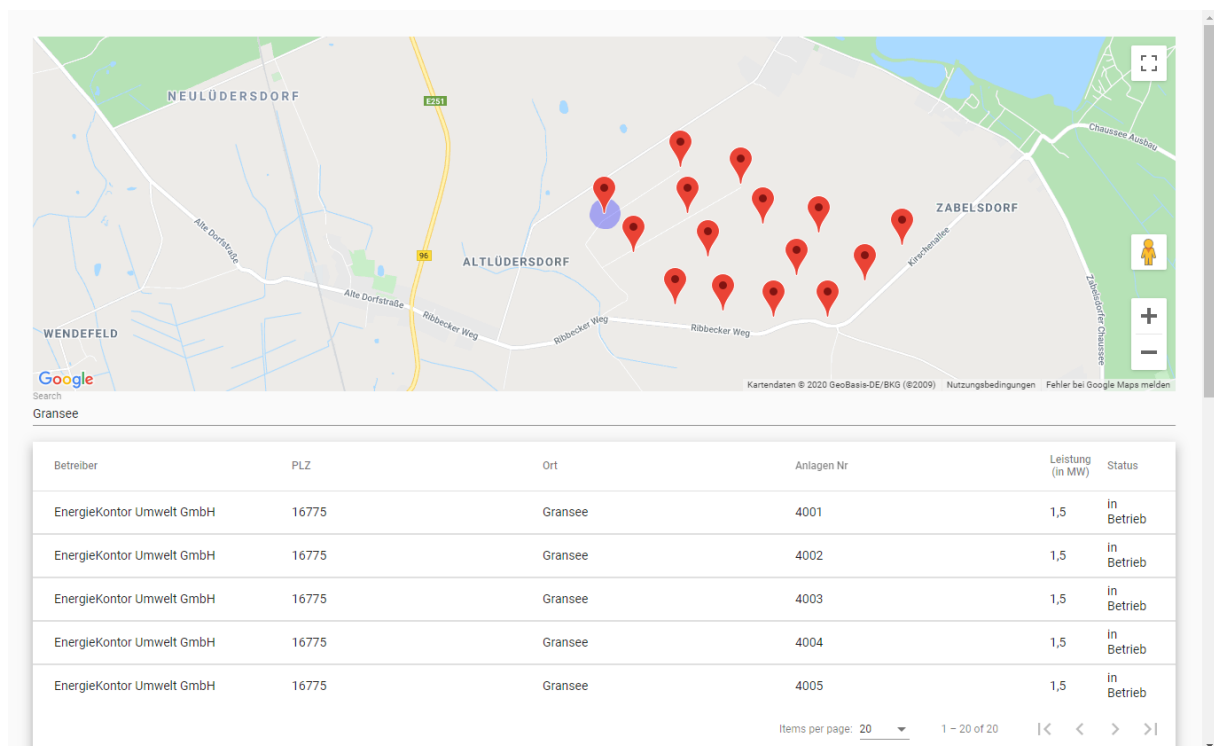


Abbildung 3: Suchfunktion nach Ort und automatischer Ausschnitt auf der Karte

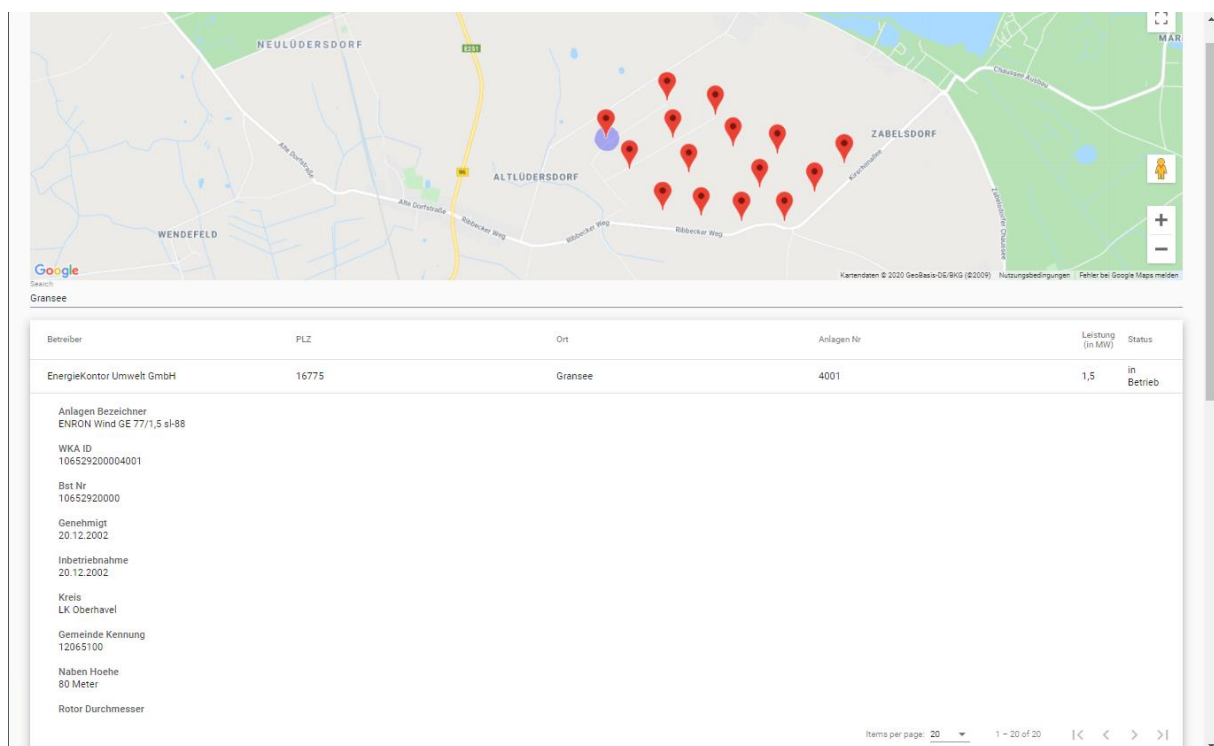


Abbildung 4: Details einer Gewählten WKA (blau) auf der Karte und deren Details in der Tabelle

Datenanalyse

Unklar:

- Variablennamen mit Kommata
 - ➔ Als Objekt nativ schlecht verwendbar
 - ➔ Direkt verwendbar sind: „Latitude“ „Longitude“
- Inkonsistente Datentypen
 - ➔ Bsp.: Leistung einmal „string“ und einmal „number“ (‘0,6‘ vs. 2)
 - ➔ Laut wka.shp verletzt Leistung in einigen den Typ
- Zwei IDs für jede WKA. Warum eine extra _id, wenn Wka_id vorhanden?
 - ➔ Wka_ID und _id

Kostenabschätzung

Unter der Voraussetzung von 35 Mitarbeitern die ca. 10 Anfragen am Tag stellen, kann mit ca. 7.000 Anfragen im Monat gerechnet werden. Genutzt im Projekt wird die „Maps Javascript API“, welche unter den Punkt „Dynamic Maps“ fällt.

Nach Googles API-Kostenübersicht² kann mit in unserem Fall von 7.000 Anfragen mit Kosten von 0\$ gerechnet werden, da diese deutlich unter die 28.000/Monat fallen, welche durch das von Google bereitgestellte Guthaben von 200\$ gedeckt sind. Sollte sich die Anfragenmenge über 28.000 erhöhen, fallen je 1.000 Aufrufe darüber (bis 100.000 Aufrufe) 7\$ bzw. 0,7ct/Aufruf an (siehe Abbildung 5).

SKU: Dynamic Maps

A web page or application that displays a map using the [Maps JavaScript API](#). A map is created with the `google.maps.Map()` class.

MONTHLY VOLUME RANGE (Price per MAP LOAD)		
0–100,000	100,001–500,000	500,000+
0.007 USD per each (7.00 USD per 1000)	0.0056 USD per each (5.60 USD per 1000)	Contact Sales for volume pricing

User interactions with the map, such as panning, zooming, or switching map layers, do not generate additional map loads.

Abbildung 5: Preistabelle für Google Maps Javascript API

² <https://cloud.google.com/maps-platform/pricing/sheet/> (10.11.2020)

Analyse der Statistik-Anforderungen

Die Anforderungen an die Statistik sind für den aktuellen Zeitpunkt umsetzbar und verständlich.

Klar:

- Nabhöhe vs. Rotordurchmesser
- Top 10 Leistungsstärkste/Anlagenstärkste PLZ Bereiche
- Box Plot über Genehmigung und Inbetriebnahme
- Bau Dauer WKA

Unklar:

- Welche Leistung? Theoretische Leistung oder tatsächliche? Nur online oder auch offline?
 - ➔ Alle WKA in DB haben Leistung eingetragen

Sprint Planning (18.11.2020 – 2.12.2020)

Eigene Ziele

- Statistikanfragen im Backend implementieren
 - ➔ Korrekte Werte werden als Funktionswerte zurückgegeben
- Statistikanzeige der fünf Diagramme im Frontend implementieren
 - ➔ Gelieferte Werte werden korrekt nach Anforderung angezeigt
- Unit-Tests für die Automatisierung durch Sonar und Jenkins erstellen
 - ➔ Erster erfolgreicher Sonar Job in Jenkins

Aufgaben

- Datumsfilter (von-bis) auch für die Kartenanzeige
- Sonar-Anschluss und
 - ➔ „alles grün“
 - ➔ Einbindung der Testabdeckung
- So viele Tests geschrieben, dass die CodeCoverage der Anwendung mindestens 20% beträgt
- Anzeige mindestens einer Statistik
- Neue Datumsforderung aufgenommen, eingeplant (und umgesetzt)

Kommentare der Dozenten/Kunden

Datenanalyse:

- Keine gute Lösung für Variablennamen
- Inkonsistente Datentypen werden überprüft
- Als ID reicht Wka_ID

Statistik:

- Gesamtleistung soll abrufbare Leistung darstellen
- Negative Bau Dauer weglassen/filtern

Anmerkungen zum aktuellen Design:

- Zwei Spalten System statt zwei Zeilen (Karte und Tabelle)
- Anpassbare Kartengröße

Neue Anforderung der Kunden:

- Datumsfilter (von-bis) auch für Karte
- Auswahl der angezeigten WKAs nach genehmigte und/oder gebaute WKA