

Meeting 11 (01.12.20)

Datum: 01.12.2020

Moderation: Florian

Protokoll: Cuong

Auswertung Abgabe Planungsphase

User Stories:

- Kein Fehler, nur eine kleine Anmerkung: Durchnummerieren, um besser diskutieren zu können
- Warum will MFF eine interessante Konfiguration speichern wollen und nicht nur der Nutzer? Es müsste reichen es nur einmal aufzunehmen → Durch unsere Erklärung doch valide

Sequenzdiagramm:

- Abgabe mit 2 Pfeilen wäre besser gewesen
- Loop invalide Konfiguration bisschen ausführlicher darstellen, dass die zurückgegeben Konfiguration noch verändert werden kann
- Sonst super

Use Case Diagramme:

- Auch super
- Alles sehr klar dargestellt

Klassendiagramm:

- Datenbank ja oder nein?
 - Für die Userverwaltung
 - Mit Spring Boot wäre es kein Ding
- Getter und Setter über Lombok (Muss nicht unbedingt in der Klasse angegeben werden → nicht praxistauglich)
- Noch nicht auf ein Framework angepasst, aber war auch nicht notwendig
- Sonst super

Frameworks

JHipster:

- Eigentlich nur ein Code Generator
- Generiert aus einer UML-artigen Datei (JDL) ein Spring Boot Backend und ein Angular/React/Vue Frontend
- Viel Arbeit mit Interfaces und Abstraktionen
- Es fällt viel Boiler Code weg

Spring Boot:

- Falls wir keine Datenbanken verwenden wäre Spring Boot fast schon ein Overkill
- Alternative: Zu Python wechseln und Flask benutzen lul

Implementierungsphase

Sprint:

- Direkt alle Sprints anlegen
- Zeitplan in Milestones umwandeln

GitLab Issues:

- Alles Issues erstellen aus den ganzen User Stories
- Beispiel: *“Anfrage des Frontends entgegennehmen und antworten”*
 - Alle Endpunkte für den User festlegen und alle Endpunkte für die Konfiguration
 - Service für den User anlegen, löschen, updaten
- Zeit schätzen → Magic Estimation Prinzip anwenden
- Ein Issue heraussuchen, bei der man die Zeit ziemlich gut abschätzen kann
- Alle bewerten die Issues und geben Punkte ab (1,2,3,5,8,13,40,100)
 - Weight benutzen in GitLab
- Viele Issues bereitstellen, um bei einem kurzen Sprint noch Issues im Backlog zu haben, die dann in der freien verfügbaren Zeit bearbeitet werden können

Retrospektive

- Meetingstruktur neu strukturieren

Follow-Ups:

- Gitflow bestimmen
- DevOps CI/CD vorstellen
- DevOps Issues erstellen
- Testing Vorstellung