

5 – Conclusio und Lessons Learned

Diego Krupitza, Jan Müller, Kian Pouresmaeil

15. Juni 2022

Inhaltsverzeichnis

1	Beurteilung des Projektablaufs	1
1.1	Was hat gut funktioniert, was weniger gut? Begründen Sie ihre Beurteilung.	1
1.2	Wenn zutreffend, warum haben Sie den ursprünglich geschätzten Aufwand überschritten?	1
1.3	Was würden Sie aufgrund der gewonnenen Erfahrung anders bzw. besser machen?	1
2	Beurteilung der finalen Software	2
2.1	Gut geglückt	2
2.2	Verbesserungsmöglichkeiten	2
3	Fazit zur Lehrveranstaltung	2

1 Beurteilung des Projektablaufs

1.1 Was hat gut funktioniert, was weniger gut? Begründen Sie ihre Beurteilung.

Im Großen und Ganzen war der Ablauf während der Entwicklung sehr gut. Aufgrund einer optimalen Projektkommunikation, dank regelmäßiger Meetings und Statusupdate innerhalb der Projektchats, haben wir sehr effizient arbeiten können. Des Weiteren wurden mithilfe von Pair-Programming Sessions Probleme technischer Natur schnell und effizient gelöst.

Der Start des Projektes verlief ein wenig holprig, da aufgrund von Abweichungen bei den Entwicklungsumgebungen der Entwickler Probleme bei der Inbetriebnahme der Services entstanden. Diese konnten jedoch durch rasche Kommunikation behoben werden.

Sehr gut hat der Einsatz eines zentralen Repositories bei GitHub funktioniert, in welchem alle Artefakte der Projektes hinterlegt wurden. Dies ermöglichte allen Entwickler jederzeit einen Zugriff auf den aktuellsten Stand des Projektes, ohne auf eine Vielzahl an Diensten zugreifen zu müssen. Insbesondere profitierte davon die Versionierung der Dokumentationsdokumente, welche mit \LaTeX erstellt wurden.

1.2 Wenn zutreffend, warum haben Sie den ursprünglich geschätzten Aufwand überschritten?

Die Aufwandsschätzung war relative genau und konnte dank einem Einsatz von modernen Frameworks zum Teil unterboten werden.

Hingegen wurde beim Einrichten des Deployments auf GKE beziehungsweise eines lokalen Kubernetes Clusters viel Zeit verloren, da das Team bisher kaum Erfahrung mit diesen Technologien gesammelt hatte und anfangs widersprüchliche Tutorials hinzugezogen wurden.

Letztendlich konnte dieser Mehraufwand durch eine effiziente Entwicklung ausgeglichen werden.

1.3 Was würden Sie aufgrund der gewonnenen Erfahrung anders bzw. besser machen?

Das Deployment der Anwendung wurde zu Beginn der Entwicklung nicht beachtet. Daher hatten wir einen Netflix-Eureka-Service für Service-Discovery konfiguriert und eingebunden. Erst beim Deployment ist uns aufgefallen, dass GKE diese Funktionalität nativ bereitstellt.

Deshalb mussten wir unsere Service-Discovery-Logik sowie deren Integration im Gateway umschreiben. Aufgrund dieser gewonnenen Erfahrung, werden wir zukünftig bereits zu Beginn der Entwicklung das Deployment sowie die Produktionsumgebung berücksichtigen.

2 Beurteilung der finalen Software

2.1 Gut geglückt

Mithilfe von Swagger-Grouping war es möglich, alle Swagger-Dokumentationen über den Gateway-Service bereitzustellen.

Da wir primär asynchrone Kommunikation nutzen, haben wir kaum blockierende Anfragen und konnten ein hohes Maß an Performanz erreichen.

Durch die Microservicearchitektur der Anwendung waren wir in der Lage für simple Services, welche lediglich CRUD-Operationen durchführen, leichtgewichtige Frameworks und Technologien einzusetzen. Analog dazu konnten wir auch für komplexere Services, welche beispielsweise wie der Simulator-Service Multithreading verwenden, Technologien mit den benötigten Funktionalitäten einsetzen.

2.2 Verbesserungsmöglichkeiten

Der Simulator-Service simuliert sowohl Ereignisse von Fahrzeugen als auch die von Ampeln. Es wäre möglich gewesen den Simulator-Service in Fahrzeug-Simulator-Service, Ampel-Simulator-Service und Simulator-Manager aufzuspalten.

3 Fazit zur Lehrveranstaltung

Thema und Aufgabenstellung waren gut beschrieben und interessant. Die freie Technologieauswahl kam dem Lerneffekt zu Gute und erlaubte eine Verwendung der neuesten Technologien, was leider nicht bei jeder Lehrveranstaltung möglich ist.

Jedoch empfanden wir dieses Projekt als ein Selbststudium. Dies erhöht zwar den Lerneffekt, gleichzeitig jedoch auch den Arbeitsaufwand. Insbesondere bei Technologien wie Kubernetes und GKE wäre es vorteilhaft, wenn im Rahmen der Lehrveranstaltung eine Einführung in die Verwendung und Konfiguration stattfinden würde. Auf der anderen Seite entspricht dies oft dem Berufsalltag eines Software-Engineers und bietet somit einen Einblick in zukünftige Arbeitsweisen.