

# tELVIS -“Lesson Learned”

Kevin Streiter

18.04.2018

Das tELVIS-Projekt steht kurz vor dem Abschluss - Während den letzten Wochen hat sich unsere Projektgruppe “TaggY” intensiv mit den in Auftrag gegebenen Anforderungen auseinandergesetzt. Aus den Anforderungen ein gesamtes System zu entwickeln, schien mir zu Beginn des Projekts noch kaum vorstellbar. Während den ersten Schritten mit der Projektgruppe stiessen wir bezüglich der Architektur auf viele Unklarheiten. Uns war noch nicht klar, wie wir die beiden zur Verfügung stehenden Server effektiv nutzen sollen. Die Statustreffen gaben uns jeweils Hinweise, in welche Richtung wir uns bewegen sollen. Dies ist auch der erste Punkt, den ich gelernt habe und auch während den Treffen betont wurde. Wir versuchten schon zu Beginn sehr spezifisch auf die Anforderungen einzugehen. Sie machten uns klar, dass eine zu hohe Spezifität den Handlungsspielraum überaus einschränken kann. Um den Entwicklungsprozess flexibler zu gestalten, legten sie uns auch den Einsatz von Scrum nahe. Durch die zuvor definierten Sprints, Milestones und Issues hatten wir ein definiertes Ziel, konnten uns aber auch an Veränderungen bezüglich des Systems anpassen.

Die Leitung der Statustreffen waren ebenfalls eine interessante Erfahrung. Noch zu Beginn eher unstrukturiert, konnten wir uns in den nachfolgenden Treffen, je länger je mehr, an unseren zuvor vereinbarten Ablauf halten. Auch wenn sie während ihren Rollen immer wieder von ihrer Lehrfunktion Gebrauch machten, so gab die Rollenverteilung doch einen Eindruck, wie es in Firmen mit Kunden in Statusmeetings ablaufen könnte.

Die Implementierung mit Spring Boot ist ein weiteres Framework, das ich kennenlernen durfte. Durch die kleine Einführung im Modul Software Engineering II konnte man bereits einen Bezug aufbauen, wie das Framework genau eingesetzt wird und welche Erleichterung es bietet. Die Implementierung der Repositories für die Tags und Pictures gestaltete sich viel einfacher als zuerst angenommen. Dass man dabei vollständig auf die Eingabe von Datenbank-Queries verzichten kann und noch vieles mehr, ist schlichtweg genial. Auch die Handhabung mit Beans wurde mit fortschreitender Implementierung verständlicher.

Den Entscheid die Datenbank und Orthanc mithilfe der Docker-Technologie zu implementieren eröffneten mir zusätzlich die Vorteile der Nutzung von Docker.

Alles in Allem habe ich während des Projekts viel Neues gelernt und bin sehr erfreut, dass ich Teil dessen sein durfte.