

RBSG: SM Notizen zu Release 1

- Sprint 1 legt Grundlagen (Netzwerk, Datenverarbeitung, ...) fest
- Sprint 2 verknüpft diese in einer schönen Anwendung

Probleme

Aufgabenfestlegung

- Schwierigkeiten bei Aufgabenkategorisierung
- Klarheit der Aufgabenstellung

Codequalität

- Ein Entwickler musste sich zum Testen in ein neues Thema einarbeiten (Patrick ins Mocken)
- Einzelne Entwickler erhielten deutlich mehr als 50 (bis 77) Änderungswünsche vom SM am vorgeschlagenen Code (siehe Bart)
 - Umgang mit Versionsverwaltungssystem Git und Verständnis seiner Verästelung ausbaufähig
 - * Aufmerksamkeit ausbaufähig
 - Allgemeines Verständnis nur durch Sammeln von Erfahrung verbesserbar
 - Jederzeit Hilfe durch SM
- Einführung von Travis und Checkstyle durch den SM zur Vermeidung größter Fehler
- SM räumte den Code in größerem Umfang auf

Manpower

- Ein Entwickler verließ nach 3 von 4 Wochen aus gesundheitlichen Gründen das Team
 - Das eigene Feature “verschlüsselter Chat” wird nicht zu Release 1 fertig, aber Grundlagen existieren
- Arbeitszeiteinteilung teils schlecht: Entwickler reichte seinen Code am letzten Tag (nur) ein, doch waren andere Entwickler für ihre Arbeit auf seine angewiesen
- Einhaltung von Deadlines klappt nur bedingt

Höhere Gewalt

- Bereitgestellter Server ändert sein Verhalten, was in dieser Form nicht geschehen sollte.

Technologien

- Java: Programmiersprache

- Gradle: Build Tool, automatisiert das Erstellen der Anwendung aus Code
- JavaFX: Bibliothek für graphische Elemente
 - Wir verwenden Java 8, da JavaFX dort direkt eingebunden ist
 - * weniger Fehlerpotential
 - * Muss zur Ausführung installiert sein
 - Große Teile der Konfiguration sind für einen Wechsel auf neuere Javaversionen vorbereitet
- Checkstyle: Sicherstellung der Codequalität
- Travis: Sicherstellung der Funktionalität
- Mockito: Tests
- JFoenix: JavaFX-Elemente im Material Design
- FontawesomeFX: Icons
- Log4J: Logging
- Verschiedene andere low-level-Abhängigkeiten