

Vorgehensplan

Vorgehensmodell

Die Vorgabe in Iterationen zu arbeiten gibt uns indirekt auch vor ein iteratives Vorgehensmodell zu verwenden. Hierfür wurde 'Scrum' ausgewählt. Dazu ist dann die Idee oder besser gesagt der Wunsch am Ende von jedem Sprint eine funktionierende Iteration zu haben.

Durch die relativ kleine Größe des Teams macht es wenig Sinn einen Scrum-Master zu haben, insbesondere nach dem die Entscheidung getroffen wurde das Team zu halbieren damit Aufgaben eben parallelisiert werden können, aber auch um die Vorteile von Pair-Programming auszunutzen (deshalb eben zwei Paare).

Die Partner werden unter sich entsprechend die Minutiae des Scrum-Prozesses ausüben insbesondere den Daily-Scrum (je nach dem ob an dem Tag im Paar programmiert wird oder nicht).

Kommunikation

Die hauptsächliche Kommunikation erfolgt über eine eingerichtete open-source Messaging-Server/Plattform namens Mattermost(R). Da werden entsprechende Kanäle, gleichbenannt wie das Thema das in dem Kanal behandelt werden soll, veröffentlicht. Zugleich dient die Plattform als ein Speicher an allen projektrelevanten Daten wie Codeschnipsel, Artikel, Diagramme, etc.

Längere Diskussionen werden auf einem eingerichteten Mumble-Server durchgeführt, wenn eben die Bandbreite von Nachrichten schlichtweg nicht ausreicht.

Eine Plattform für Videoconferencing und Screensharing namens Jitsi ist auch vorgesehen, nun leider ein wenig anspruchsvoller einzurichten, bzw. zum Zeitpunkt der Erstellung dieses Dokumentes über die DevOps-Fähigkeiten des Umsetzers hinausgelaufen.

Arbeitsteilung

Die Arbeit wird durch das Assignment-System, das in Gitlab von vorne rein eingebettet ist, verteilt. Dazu wurde ein Team-Lead entwickelt gewählt, der im evtl. Zweifallsfall das finale Sagen bzw. die letztendliche Entscheidungsgewalt besitzt.

Die Aufgaben leiten sich aus einer Gesamtübersicht ab, die das bereits erreichte und vor allem Next-Actions in einer baumartigen Struktur anzeigt. Eine rudimentäre Übersicht der gesamten Arbeit wird auf einer Webseite (hier) angezeigt und (hoffentlich) täglich von dem Administrator mit frischen Daten neugeneriert.

Repository

Vorgegeben ist GitLab, die Idee ist dass wir jeweils eine lokale Kopie von unserer Repository haben an der dann privat gearbeitet wird und schlussendlich ein Merge-Request gemacht, der dann von jedem Glied angeschaut werden soll. Vier Upvotes bedeuten, dass der Merge-Request angenommen wird von dem Team-Lead, nach dem das ganze Team ein Rebase der eigenen lokalen Repository durchzuführen hat.

Bei Fehlverhalten des Programms soll mit Issues und Git-Branches gearbeitet werden, die dann nach der Behebung wieder an die Master-Branch gemerged werden.

In der Planung sind eine sinnvolle Benutzung von Tags, was 'sinnvoll' heißt wurde jedoch noch nicht entschieden.

Rest

Die restlichen Entscheidungen sind dem jeweiligen Team oder Teamglied frei überlassen, man soll schließlich auch Micro-Management vermeiden.

Damit hoffen wir auf eine erfolgreiche Teilnahme am Projekt