

Wochenbericht I - Praktikum "Verteilte Systeme": Organisation und Projektvorbereitung

Marc Siekmann

April 27, 2025

1 Mitglieder des Projektes

Mitglied des Projektes	Aufgabe	Fortschritt	Zeiteinsatz	Check
Manh-An David Dao	Aufbau der DevOps-Toolchain	70%	2h	ok
Jannik Schön	Erstellung der Projektdokumentation nach arc42	100%	3h	ok
Marc Siekmann	Einarbeitung des ITS-BRD	60%	4h	ok
Phillip Patt	Vorbereitung und Aufbau des Kransystems	100%	8h	ok

2 Zusammenfassung der Woche

In der ersten Woche fand die Gruppenfindung statt. Ausserdem wurden wichtige organisatorische Grundlagen, wie ein Jour-Fixe, geplant und festgelegt. In der zweiten Woche wurden die Rahmenbedingungen, sowie ein grundlegendes Verständnis der vorliegenden Aufgabe erörtert. Zudem wurde auf GitLab ein Repository erstellt. Weiterhin haben die verschiedenen Teammitglieder ihre zugeteilten Aufgaben vorbereitet. Es wurde sich in technischer Sicht mit dem zu bedienenden Roboter und dem ITS-Board beschäftigt. Ausserdem wurde ein Entwicklungsdokument mithilfe des ARC-42 Templates entworfen. Mithilfe dieses Templates wurde die erste Iteration des Entwicklungsprozesses gestartet. Dies war nach den ersten beiden Vorlesungsinhalten möglich und wurde der ersten Praktikumseinheit durchgeführt, da nun ein grundlegendes Verständnis der Aufgabenstellung vorhanden war. Die erste Iteration wird vorerst mit allen Teammitgliedern durchgeführt, damit alle sich in die arc42 Dokumentation einarbeiten können und zukünftig eine sinnvolle Aufgabenteilung möglich ist. Zusammengefasst lässt sich feststellen, dass dem Team die Aufgabenstellung bis zum ersten Praktikumstermin unklar war. Es konnte aber die Organisation des Teams und in die einzelnen gegebenen und vorgeschlagenen Aufgabenteile vorbereitet werden.

3 Bearbeitete Themen und Schlüssel Erkenntnisse

3.1 Aufbau des DevOps

Planung:

Durch Erfahrungen aus vorherigen Projekten hat sich der Kanban-Stil als sehr hilfreich erwiesen, um unsere Arbeitsweise strukturierter zu gestalten.

Für die Planung nutzen wir dabei hauptsächlich die kostenlose Version von Trello, da das Team bereits in früheren Projekten positive Erfahrungen damit gesammelt hat.

Entwicklung:

Es wurde ein GitLab-Repository für das VS-Projekt eingerichtet. Zur besseren Organisation des Projekts wurden verschiedene Ordnerstrukturen angelegt und erste README-Dateien vorbereitet, um eine klare

Struktur und Orientierung für alle Beteiligten zu schaffen. Alle Key-Member des Projekts wurden in das Repository eingeladen und mit den jeweils erforderlichen Rollen und Rechten ausgestattet.

Für die Zusammenarbeit wurde eine Branch-Strategie eingeführt:

- Es wird nicht direkt auf der Main-Branch gearbeitet.
- Änderungen, Protokollierung und Dokumentation werden über Subbranches entwickelt.
- Auf die Main-Branch wird nur der jeweils aktuellste, getestete und saubere Stand gepusht, um die Stabilität sicherzustellen.

Im GitLab-Repository werden künftig sowohl die verschiedenen Versionen des Projekts als auch die zugehörige Dokumentation gemäss Arc42 sowie das wöchentliche Protokoll zentral gepflegt und abgelegt. Durch die Nutzung von Git stellen wir eine saubere, klare Struktur sicher und schaffen gleichzeitig eine Absicherung durch eine kontrollierte Versionsverwaltung.

Entwicklungsumgebung:

Als Entwicklungsumgebung nutzen wir Visual Studio Code (VS Code), da das Team bereits in früheren Projekten gute Erfahrungen damit gesammelt hat und deshalb weiterhin darauf setzt. Zusätzlich wurde uns eine Guideline zur Verfügung gestellt, in der VS Code als Beispiel verwendet wurde. Dadurch konnte eine einheitliche Arbeitsweise geschaffen werden, die sich als ideal erwiesen hat und perfekt zu unserem Workflow passt.

3.2 Erstellung der Dokumentation

Es wurde sich in das arc42-Template eingearbeitet, wobei der Fokus auf dem Verständnis des Aufbaus, der Struktur und der typischen Anwendung lag. Anschliessend wurde eine LaTeX-Schablone des arc42-Templates recherchiert und in das Repository hochgeladen. Dieses Dokument bietet nun eine iterativ bearbeitbare Möglichkeit, das Projekt zu gestalten. Im Anschluss wurde den Teammitgliedern eine grundlegende Einführung in arc42 gegeben, sodass diese nun die Möglichkeit haben, aktiv an der Projektdokumentation mitzuarbeiten.

3.3 Einarbeitung in das ITS-BRD

Für das ITS-Board liegt ein git-Repository vor. Auf diesem ist ein Beispielprogramm für einen TCP-Server hinterlegt. Die eigentliche Schwierigkeit bestand darin Visual Studio Code einzurichten, um die Programmierung in den späteren Workflow einzugliedern. Dabei sind diverse Probleme aufgetreten, die das Kompilieren des Codes verhindert haben. Um dieses Problem zu lösen wurde die meiste Zeit investiert. Der/die Fehler scheinen aber Plattformabhängig zu sein, da sie nicht zuverlässig reproduziert werden konnten. Letztlich konnte der bereitgestellte Beispielcode auf dem ITS-Board ausgeführt werden. Der TCP-Server auf dem ITS-Board konnte per "ping"-Befehl von einem externen Rechner mit einer 1:1 Verbindung gefunden werden.

3.4 Aufbau des Kransystems

Der Aufbau eines Kranes wurde sich angeschaut und festgestellt, dass es drei unabhängige Servos besitzt. Der Kran wird von einem Raspberry Pi gesteuert. Der Raspberry Pi wird über ein Netzwerk angesprochen. Die Software dazu wurde sich angeschaut, sowie in einer virtuellen Umgebung getestet (Windows und Linux). Alle bereitgestellten Funktionalitäten konnten somit erfolgreich getestet werden. Im nächsten Schritt wurde versucht den Kran in der ICC Cloud bereitzustellen. Dieser Versuch konnte noch nicht abgeschlossen werden. Die ICC Cloud selbst wurde erfolgreich eingerichtet und ein kleines Beispielprogramm getestet. Damit ist der Aufbau eines Kransystem verstanden und die Steuerung mithilfe der API theoretisch in den nächsten Schritten möglich. Die ICC Cloud muss noch genauer verstanden werden, bzgl. der verfügbaren Grenzen (dedizierte IP-Adresse, Einrichten einer GUI).

4 Bezug zur Vorlesung

Da der Aufbau der Vorlesung einen iterativen Projektaufbau begünstigt, wird wie vorgeschlagen die arc42 Dokumentation verwendet. In der angefangenen ersten Iteration wurden die ersten drei Schritte

durchgeführt, da diese das grobe Software Engineering betreffen und sich noch nicht mit dem Design und der Architektur beschäftigen. Für das Use-Case Diagramm eignet sich laut Vorlesung eine funktionale Zerlegung. Diese wird demnach für die erste Iteration und die erste grobe Systembeschreibung im ersten Kapitel angefertigt. Wir haben uns danach mit den allgemeinen Zielen (Skript S.23) beschäftigt, und uns darauf geeinigt Platzhalter zu erstellen, damit die Ziele in den nächsten Iterationen messbar gemacht werden können. Danach haben wir uns mit den Zielen der Verteilten Systeme beschäftigt und uns da auf die Skalierbarkeit und Verteilungstransparenz konzentriert. In der Vorlesung wurde bzgl. der Transparenzen festgelegt, dass die Zugriffstransparenz wichtig ist, da nur eine GUI für den Nutzer bereitsteht. Parametereingaben sollten ebenfalls vermieden werden, damit die Art der Steuerung verschleiert wird. Zudem ist in der ersten Iteration in Bezug auf die Systemarchitektur angedacht, eine zusätzliche Schicht oder Middleware als Kommunikationssteuerung einzusetzen, um das System als kohärent sichtbar zu machen. Bei der Skalierbarkeit wurde festgestellt, dass die räumliche Skalierbarkeit und die administrative Skalierbarkeit vernachlässigbar ist, da diese im nach Aufgabe mit 1 festgelegt werden. Es gilt also die Problemgrösse (Befehle pro Sekunde) zu skalieren, da Delay und Bandbreite in einem lokalen Netzwerk beeinflusst werden können.

5 Nächste Schritte

Die erste Iteration des arc42-Protokolls weiterführen und beenden. Das ITS-Board wird in ein bestehendes /24 Netzwerk eingebunden und kann per "ping" gefunden werden. Danach wird beispielhaft ein Roboter über dessen API angesprochen. Dies dient dazu mögliche Probleme des Gesamtsystems zu erkennen und an die festgelegten Ziele im arc42 Dokument anzupassen. Grundsätzlich muss innerhalb der Gruppe eine sinnvolle Aufgabenteilung erarbeitet werden, da eine Iteration des arc42 Dokuments momentan nicht von einer Person gemacht werden kann.