

HTL Saalfelden

Systemplanung und Projektentwicklung



Projektbezeichnung	MBot2
Projektteam	Philipp Kirchtag, Sebastian Krallinger, Fabian Scharfetter, Eduard Voicescu
Erstellt am	10.01.2024
Letzte Änderung am	21.02.2024
Status	[fertiggestellt]
Aktuelle Version	1.0

Änderungsverlauf

Nr.	Datum	Version	Geänderte Kapitel	Art der Änderung	Autor
1	10.01.2024	1.0	Alle	Erstellung	Sebastian Krallinger
2	24.01.2024	1.0	2	Anforderungen	Sebastian Krallinger
3	21.02.2024	1.0	6.1	Sprint 1 Doku,	Philipp Kirchtag
4	22.02.2024	1.0	3,4.2	Nichtfunktionale Anforderungen, Machbarkeitsstudie	Philipp Kirchtag
5	07.03.2024	1.0	6.2	Sprint 2 Doku	Fabian Scharfetter
6	11.04.2024	1.0	5.4	Verteilungsdiagramm erstellt	Philipp Kirchtag
7	17.04.2024	1.0	6.3	Sprint 3 Doku	Philipp Kirchtag
8	18.04.2024	1.0	6.4	Sprint 4 Planung	Philipp Kirchtag
9	25.04.2024	1.0	5.6, 2	Andwendungsfalldiagramm, Use Cases	Sebastian Krallinger, Eduard Voicescu
10	16.05.2024	1.0	5.4, 5.5.1	Sprint 4 Doku und Sprint 5 Planung	Philipp Kirchtag
11	11.06.2024	1.0	3, 4, 5, 6, 7	Doku fertigstellen	Sebastian Krallinger

Inhalt

1.	Allgemeines / Projektübersicht.....	5
1.1	Projektbeschreibung	5
1.2	Projektteam und Schnittstellen	5
2.	Funktionale Anforderungen.....	5
2.1	Use Cases	5
2.1.1	Verbindung zwischen Roboter und Anwendung herstellen.....	5
2.1.2	Sensordaten anzeigen.....	6
2.1.3	Roboter steuern	6
2.1.4	Sicherheitsmodus aktivieren/deaktivieren.....	6
2.1.5	Netzwerkanbindung des Roboters	6
2.1.6	Autonomer Linienfolgemodus	6
2.1.7	LED-Farben einstellen	7
2.1.8	Status der Lichtsensoren im Linienfolgemodus anzeigen	7
2.1.9	Darstellung des mBot im 3D-Raum basierend auf dem Lagesensor	7
3.	Projektplanung.....	7
3.1	Variantenbildung	7
3.2	Machbarkeitsstudie	8
3.3	Projektumfeldanalyse	8
4.	Softwarearchitektur.....	8
4.1	Aktivitätsdiagramm.....	8
4.2	Verteilungsdiagramme.....	9
4.3	Anwendungsfalldiagramm	9
4.4	Softwarekomponenten / Programme	10
4.4.1	SW-Programme.....	10
4.4.2	SW-Komponenten.....	10
5.	Projektdurchführung.....	12
5.1	Sprint 1	12
5.1.1	Sprintplanung.....	12
5.1.2	Sprint Demo	12
5.1.3	Sprint Retrospektive	12
5.1.4	Sprint Zusammenfassung.....	13
5.2	Sprint 2	14
5.2.1	Sprintplanung.....	14
5.2.2	Sprint Demo	14
5.2.3	Sprint Retrospektive	14
5.2.4	Sprint Zusammenfassung.....	15
5.3	Sprint 3	16
5.3.1	Sprintplanung.....	16
5.3.2	Sprint Demo	16
5.3.3	Sprint Retrospektive	16
5.3.4	Sprint Zusammenfassung.....	17
5.4	Sprint 4	18
5.4.1	Sprintplanung.....	18
5.4.2	Sprint Demo	18
5.4.3	Sprint Zusammenfassung.....	19

5.5 Sprint 5	19
5.5.1 Sprintplanung.....	19
5.5.2 Sprint Demo	20
5.5.3 Sprint Zusammenfassung.....	20
6. Installation / Software deployment.....	21
7. Projektabschluß	21
7.1 Projektzusammenfassung.....	21

1. Allgemeines / Projektübersicht

1.1 Projektbeschreibung

Der mBot2 von mblock ist ein Roboter, der viele verschiedene Sensoren besitzt, mit denen er mit der Außenwelt kommunizieren kann. Programmiert wird er über MicroPython, eine abgespeckte Variante von Python3. Über einen cyberpi, eine „Controllereinheit“, welche auf den mBot2 gesteckt wird, kommuniziert man mit dem Roboter. Durch die verschiedenen Sensoren sollen verschiedenen Anforderungen, wie zum Beispiel die Steuerung über ein Endgerät, Anzeige des Status mittels LEDs oder die Implementierung eines Sicherheitsmodus, realisiert werden.

Das Projekt wird mit SCRUM umgesetzt und der Quellcode sowie die Dokumentation in einem GITHUB Repository abgelegt.

1.2 Projektteam und Schnittstellen

Rolle(n)	Name	Telefon	E-Mail	Team
Projektleiter	Fabian Scharfetter	0664 1900355	fabian.scharfetter@htl-saalfelden.at	3
Projektmitarbeiter	Philipp Kirchttag	0664 75039947	philipp.kirchttag@htl-saalfelden.at	3
Projektmitarbeiter	Eduard Voicescu	0677 6200810504	eduard.voicescu@htl-saalfelden.at	3
Projektmitarbeiter	Sebastian Krallinger	0660 2384312	sebastian.krallinger@htl-saalfelden.at	3

2. Funktionale Anforderungen

2.1 Use Cases

2.1.1 Verbindung zwischen Roboter und Anwendung herstellen

Ziel: Eine Verbindung zwischen dem Roboter und der Anwendung herstellen.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Der Benutzer startet die Anwendung auf seinem Computer oder mobilen Gerät.

Die Anwendung sucht nach verfügbaren Robotern im Netzwerk.

Der Benutzer wählt einen Roboter aus der Liste aus.

Eine Verbindung zwischen der Anwendung und dem Roboter wird hergestellt.

Der Roboter zeigt seinen eindeutigen Namen und seine IP-Adresse in der Anwendung an.

2.1.2 Sensordaten anzeigen

Ziel: Die Werte der eingesetzten Sensoren und Aktoren anzeigen.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Die Anwendung zeigt die aktuellen Werte der Sensoren und Aktoren des verbundenen Roboters an.

Die Werte werden in regelmäßigen Abständen aktualisiert.

2.1.3 Roboter steuern

Ziel: Den Roboter mit Steuerbefehlen bewegen.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Der Benutzer gibt über die Anwendung Steuerbefehle (vorwärts, rückwärts, links, rechts) ein.

Die Geschwindigkeit kann über die Anwendung angegeben werden.

2.1.4 Sicherheitsmodus aktivieren/deaktivieren

Ziel: Den Sicherheitsmodus ein- oder ausschalten.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Der Benutzer aktiviert oder deaktiviert den Sicherheitsmodus über die Anwendung.

Im Sicherheitsmodus werden bestimmte Befehle blockiert, um Unfälle zu verhindern.

2.1.5 Netzwerkanbindung des Roboters

Ziel: Automatische Verbindung des Roboters mit dem Schul-WLAN.

Akteure: Benutzer, Roboter

Beschreibung:

Der Roboter wird eingeschaltet.

Der Roboter versucht, sich automatisch mit dem Schul-WLAN zu verbinden.

Die LEDs der Controller-Einheit zeigen den Status der WLAN-Verbindung an (Blau, Grün, Rot).

2.1.6 Autonomer Linienfolgemodus

Ziel: Der Roboter folgt automatisch einer Linie.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Der Benutzer aktiviert den autonomen Linienfolgemodus über die Anwendung.

Der Roboter beginnt, die Werte des Lichtsensors auszuwerten, um einer Linie zu folgen.

Die Anwendung zeigt den Status der 4 Lichtsensoren an.

2.1.7 LED-Farben einstellen

Ziel: Die Farben der LEDs der Controllereinheit einstellen.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Der Benutzer wählt in der Anwendung die gewünschten Farben für die LEDs aus.

Die Anwendung sendet die Farbeinstellungen an den Roboter.

Die LEDs der Controllereinheit leuchten in den ausgewählten Farben.

2.1.8 Status der Lichtsensoren im Linienfolgemodus anzeigen

Ziel: Den Status der 4 Lichtsensoren im Linienfolgemodus anzeigen.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Im Linienfolgemodus zeigt die Anwendung den aktuellen Status der 4 Lichtsensoren an.

Die Anzeige ändert sich je nachdem, ob ein Sensor über weißem oder schwarzem Hintergrund ist, basierend auf den Codierungen der Sensoren.

2.1.9 Darstellung des mBot im 3D-Raum basierend auf dem Lagesensor

Ziel: Den mBot im 3D-Raum darstellen, basierend auf den Werten des Lagesensors.

Akteure: Benutzer, Roboter, Anwendung

Beschreibung:

Die Anwendung stellt den mBot im 3D-Raum dar.

Bei einer geraden ebenen Fläche wird der mBot auch gerade und eben dargestellt.

Wenn der mBot um die Längsachse geneigt ist (z.B. um 45 Grad), wird diese Neigung ebenfalls in der Darstellung berücksichtigt.

3. Projektplanung

3.1 Variantenbildung

Es wurde ein Spring Boot Server verwendet, da diese Technologie in einem anderen Fach schon gelernt wurde und der Umgang vertrauter war als der anderer Server Technologien. Für die Webseite wurde HTML, CSS und JS gewählt da auch diese Technologien in der Schule gelernt wurden. Für das Programmieren des mBots wurde die vorgegebene IDE verwendet.

3.2 Machbarkeitsstudie

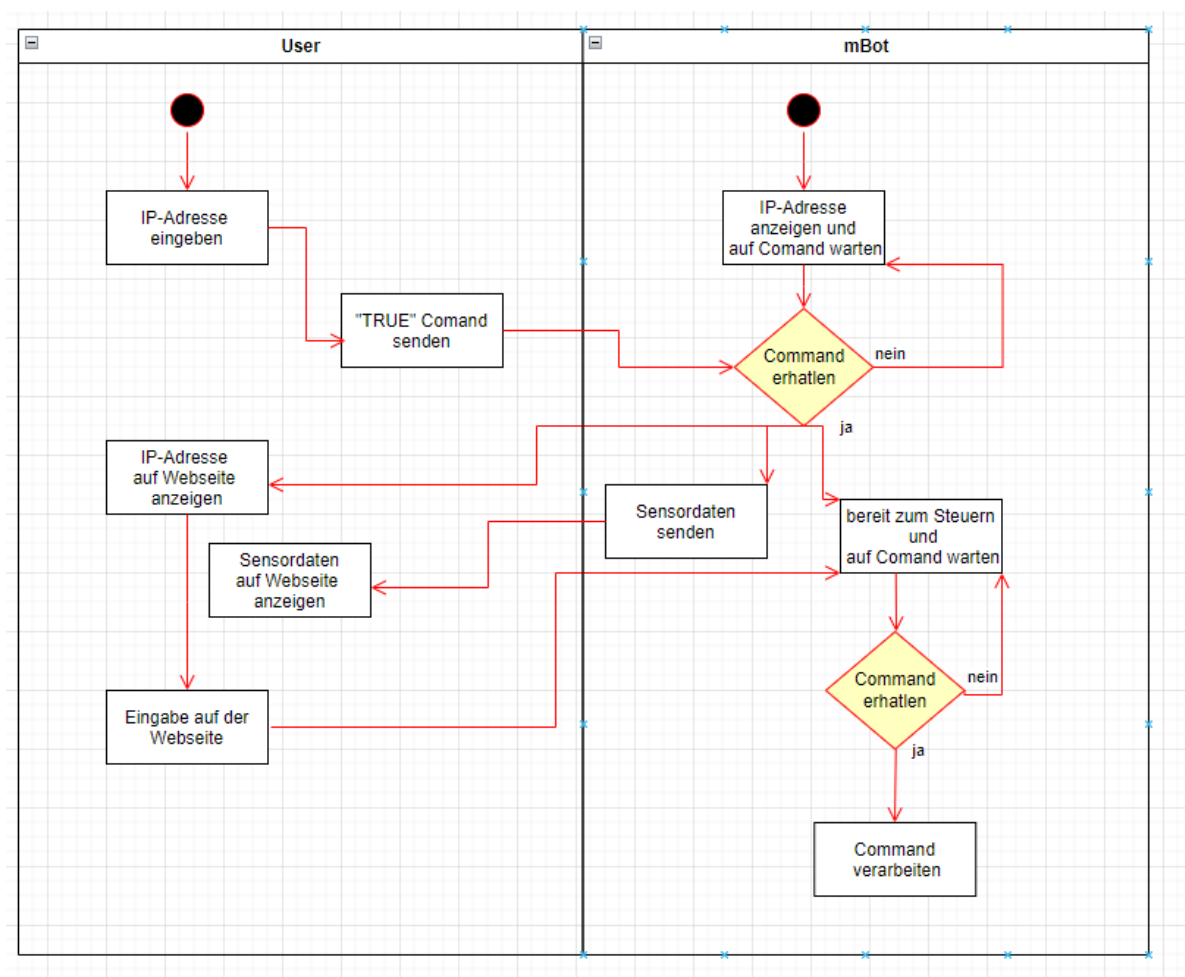
Um mit dem mBot vertraut zu werden, wurden vor dem Projektstart in einem demo Projekt verschiedene Funktionen getestet wie z. B. die Verbindung im Netzwerk zwischen PC, Smartphone und mBot sowie die Fortbewegung des mBots über Buttons auf einer Website welche von einem Endgerät gesteuert werden können.

3.3 Projektumfeldanalyse

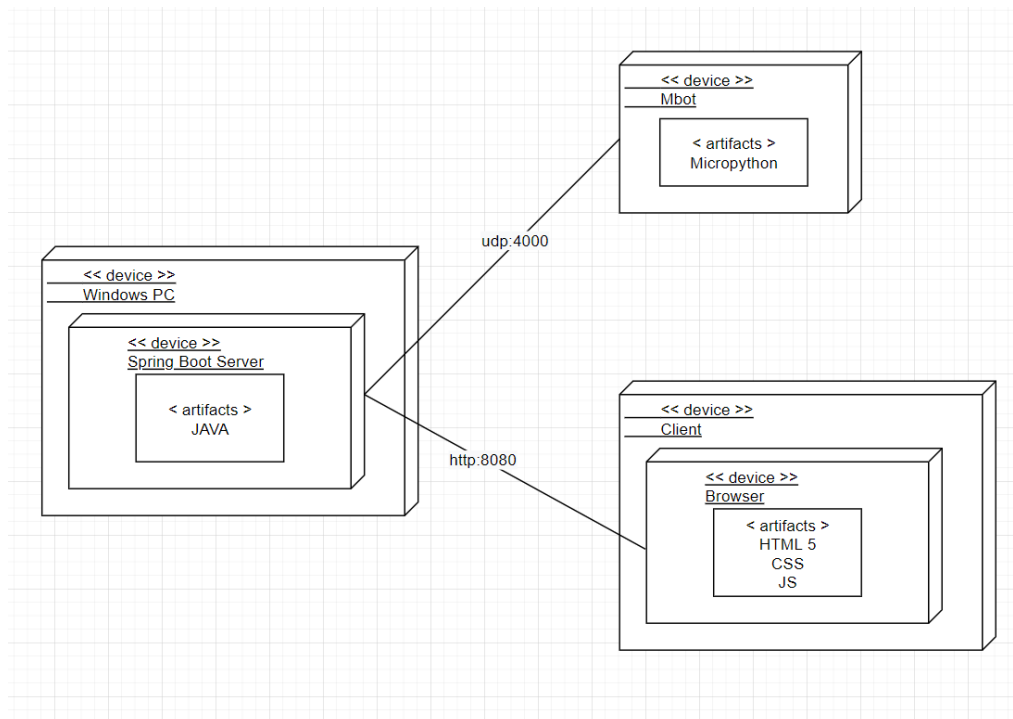
Dieses Projekt ist für alle Mitarbeiter das Erste seiner Art und es wurde in der Vergangenheit nichts Vergleichbares umgesetzt. Die Stakeholder sind Herr Prof. Falkensteiner und Eigner, diese stellen das Pflichtenheft und die Anforderungen an das Projektteam.

4. Softwarearchitektur

4.1 Aktivitätsdiagramm

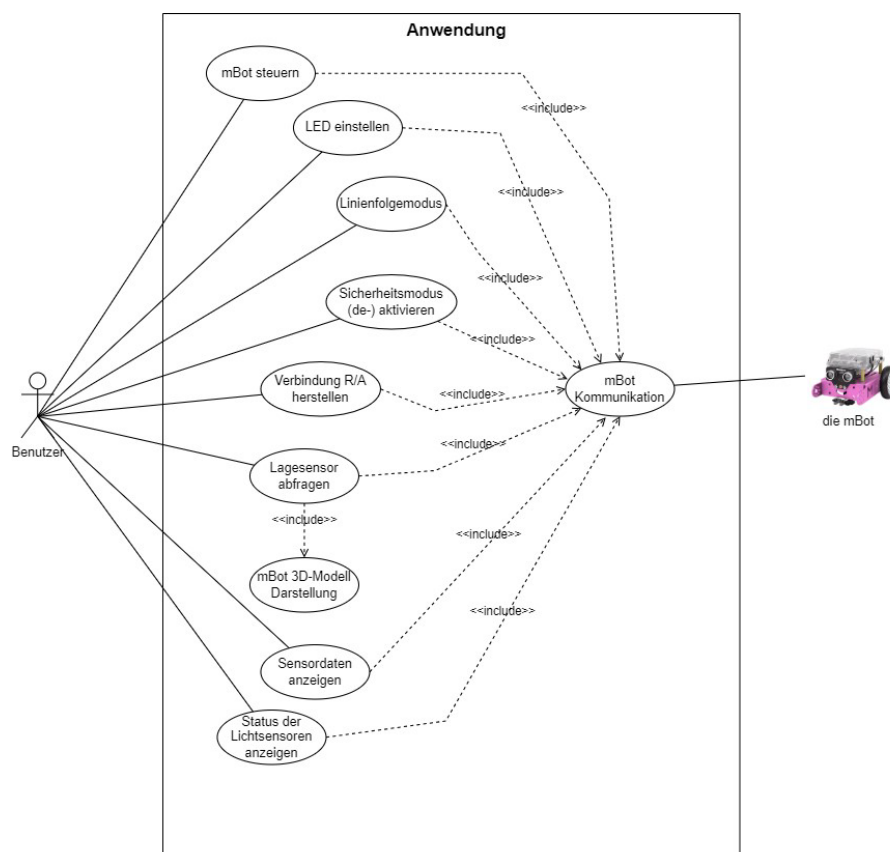


4.2 Verteilungsdiagramme



(11.04.2024)

4.3 Anwendungsfalldiagramm



4.4 Softwarekomponenten / Programme

4.4.1 SW-Programme

IntelliJ IDEA 2022.2.1

[<https://www.jetbrains.com/idea/whatsnew/2022-2/>]

Visual Studio Code 1.89.1

[<https://code.visualstudio.com/docs>]

Visual Studio 2022 17.10.1

[<https://visualstudio.microsoft.com/de/thank-you-downloading-visual-studio/?sku=Community&channel=Release&version=VS2022&source=VSLandingPage&cid=2030&passive=false>]

mBlock PC version 5.4.3

[<https://mblock.makeblock.com/en/download/>]

GitHub Desktop 3.3.18

[https://desktop.github.com/?ref_cta=download+desktop&ref_loc=installing+github+desktop&ref_page=docs]

4.3.2 SW-Komponenten

- Spring Boot
 - spring-boot-starter-web 3.2.2
 - spring-boot-starter-test 3.2.2
 - spring-boot-starter-thymeleaf 3.2.2
 - springdoc-openapi-starter-webmvc-ui 2.5.0
 - spring-boot-starter-test 3.2.3
 - lombok 1.18.30
 - spring-boot-starter-websocket 3.2.3
 - Gson 2.10.1
- Webseite
 - <https://ajax.googleapis.com/ajax/libs/jquery/3.5.1/jquery.min.js>
 - <https://cdnjs.cloudflare.com/ajax/libs/three.js/r128/three.min.js>
 - <https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/controls/OrbitControls.js>
 - <https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/OBJLoader.js>
 - <https://cdn.jsdelivr.net/npm/three@0.128.0/examples/js/loaders/MTLLoader.js>
 - <https://cdnjs.cloudflare.com/ajax/libs/sockjs-client/1.5.1/sockjs.js>
 - <https://cdnjs.cloudflare.com/ajax/libs/stomp.js/2.3.3/stomp.min.js>
 - <https://static.sketchfab.com/api/sketchfab-viewer-1.0.0.js>



- Mblock
 - time
 - usocket
 - ujson
 - os
 - mbuild
 - cyberpi
 - _thread

5. Projektdurchführung

5.1 Sprint 1

5.1.1 Sprintplanung

Dauer: 07.02.2024 – 27.02.2024

Ausgewählte User Stories:

- Steuerung PC, 5 SP, 11 VP
 - Der Benutzer soll über einen PC mit Buttons auf der Website sowie über Pfeil- und WASD – Tasten den mBot steuern können.
- Statusanzeige, 2 SP, 2 VP
 - Die LEDs des mBots sollen je nach Verbindungsstatus angezeigt werden
- Verbindung herstellen, 3 SP, 11 VP
 - Alle mBots im verbundenen Netzwerk werden angezeigt und können ausgewählt werden, um eine Verbindung herzustellen.

Anzahl Story points: 10

5.1.2 Sprint Demo

Die User Stories Steuerung PC sowie Statusanzeige wurden erfolgreich umgesetzt. Der mBot kann über Pfeil- oder WASD-Tasten angesteuert werden und die LEDs leuchten je nach Verbindungsstatus in anderen Farben.

Die User Story Verbindung herstellen wurde aufgrund von einigen Problemen nicht abgeschlossen und wird in den nächsten Sprint übernommen.

5.1.3 Sprint Retrospektive

Was gut lief:

- Sprint-Planung
- Dokumentation
- Schätzung des Aufwands (Value Points)
- Umgang mit GitHub

Verbesserungsbedarf:

- Arbeitsaufteilung
- Konzept bei Verbindung mit mBot

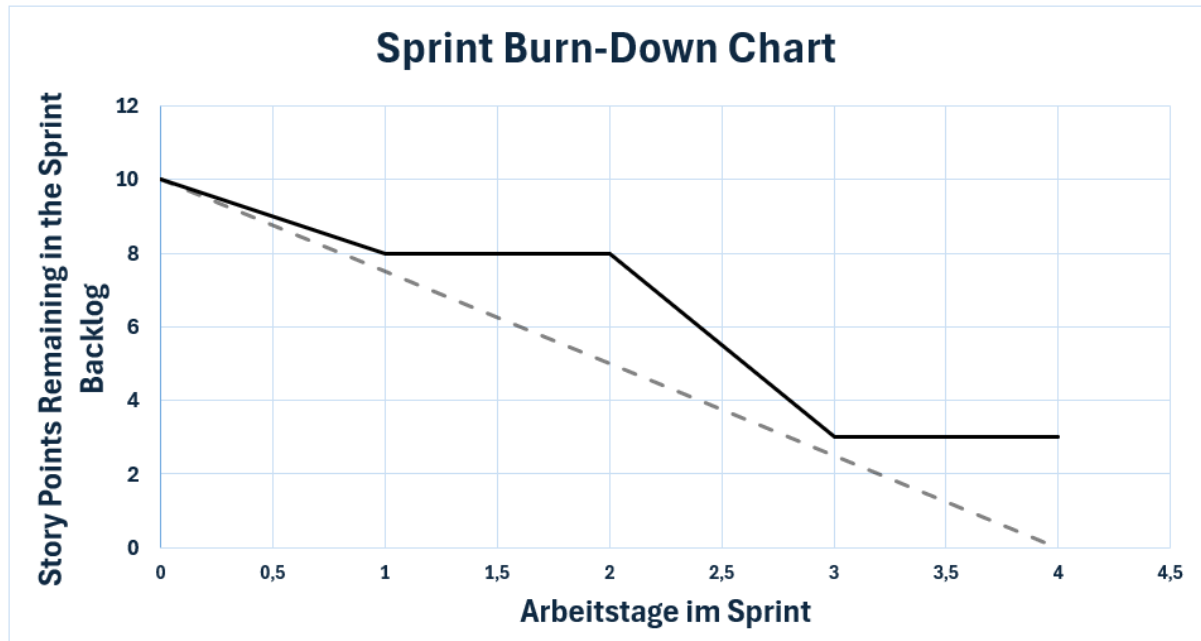
Auflistung der Impediment Taskliste:

- Verbindung herstellen

5.1.4 Sprint Zusammenfassung

Während des Sprints wurden keine neuen User Stories in das Product Backlog eingefügt bzw. entfernt.

Burndownchart:



Von 10 eingeplanten Story-Points wurden 7 komplett abgeschlossen.

Die übrigen 3 Story Points sind bereits in Bearbeitung und fast fertig – sie werden in den nächsten Sprint übernommen.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints: 7

5.2 Sprint 2

5.2.1 Sprintplanung

Dauer: 29.02.2024 – 20.03.2024

Ausgewählte User Stories:

- Website: 20 SP, 10 VP
 - Als Benutzer möchte ich eine Website über die ich den mBot2 steuern kann und alle Werte angezeigt bekomme.
- Steuerung Handy: 8 SP, 11 VP
 - Als Benutzer möchte ich den mBot2 über mein Handy steuern können, um herumzufahren.
- Sicherheitsmodus: 8 SP, 8 VP
 - Als Benutzer möchte ich eine Sicherheitsmodus ein- und ausschalten können, um nicht gegen die Wand fahren zu können.

Anzahl Story points: 36

Ausgewählte Punkte aus der Impediment Liste:

- Verbindung herstellen: 3 SP, 11 VP

5.2.2 Sprint Demo

Die User Stories Sicherheitsmodus und Steuerung Handy wurden erfolgreich abgeschlossen. Zusätzlich wurde die User Story Verbindung herstellen vom letzten Sprint fertiggestellt. Die Website ist nun responsive und kann per mobilen Endgerät mit einem Joystick gesteuert werden.

Die Website konnte noch nicht komplett fertiggestellt werden und die User Story wird in den nächsten Sprint übernommen.

5.2.3 Sprint Retrospektive

Was gut lief:

- Dokumentation
- Arbeit mit GitHub
- Zusammenarbeit
- Teamchemie
- Arbeits-Velocity

Verbesserungsbedarf:

- Arbeitsaufteilung
- Website
- User Stories unterteilen
- Planung
- Schätzung der User Storys

Auflistung der Impediment Taskliste:

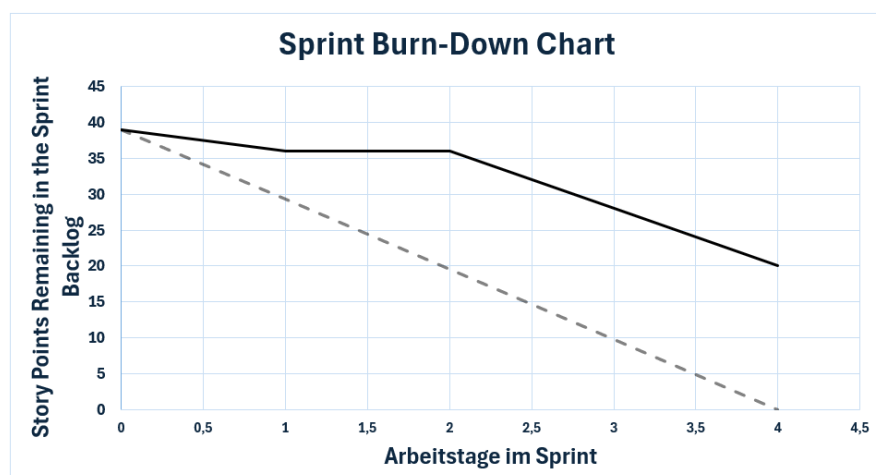
- Website

5.2.4 Sprint Zusammenfassung

Zu Beginn dieses Sprints wurden zwei neue User Stories zum Product Backlog hinzugefügt:

- GUI-Linienfolgen #13: 10 VP, 10 SP
 - Als Benutzer möchte ich eine GUI haben die mir beim Linienfolgemodus anzeigt welchen Status die 4 Lichtsensoren haben.
- MBot Lagedarstellung #14: 4 VP, 20 SP
 - Als Benutzer möchte ich auf der Webseite ein 3D Modell des Mbots angezeigt bekommen, um die Lage im 3-dimensionalen Raum zu verbildlichen.

Burndownchart:



Von 39 eingeplanten Story-Points wurden 19 komplett abgeschlossen.

Die Fertigstellung Website konnte noch nicht erreicht werden wodurch 20 Story Points noch offen sind – sie werden in den nächsten Sprint übernommen.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints: 13

5.3 Sprint 3

5.3.1 Sprintplanung

Dauer: 03.04.2024 – 17.04.2024

Ausgewählte User Stories:

- Mehrere Geräte: 8 SP, 7 VP
 - Als Benutzer möchte ich mehrere Geräte steuern und den gerade Verbundenen Roboter mit einem Namen und seiner IP-Adresse angezeigt bekommen, um einen Überblick über die Verbindungen zu haben.
- Geschwindigkeit: 3 SP, 1 VP
 - Als Benutzer möchte ich sehen wie schnell der mBot2 fährt und es in der UI angezeigt zu bekommen.
- Sensorenwerte: 3 SP, 7 VP
 - Als Benutzer möchte ich in der UI sämtliche Sensoren angezeigt und in regelmäßigen Abständen aktualisiert bekommen, um die Daten auszuwerten.

Ausgewählte Punkte aus der Impediment Liste:

- Website: 20 SP, 10 VP (User Story wurde aus Sprint 2 übernommen)
 - Als Benutzer möchte ich eine Website, über die ich den mBot2 steuern kann und alle Werte angezeigt bekomme.

Anzahl Story points: 34

5.3.2 Sprint Demo

Die User Storys mehrere Geräte, Geschwindigkeit und auch die Website, die vom Sprint 2 übernommen wurde, wurden abgeschlossen. Über ein Dropdown-Menü können alle MBots die verbunden sind ausgewählt werden. Unter den Steuerelementen wurde sowohl für PC-Nutzer als auch für Smartphone-Nutzer ein Slider platziert über den die Geschwindigkeit beliebig veränderbar ist. Die Website wurde neben den Funktionen, die sie vorher schon hatte durch Design und Inhalt bezüglich des Teams aufgewertet und konnte nun, als Fertig definiert werden.

Die User Story Sensorenwerte stellte sich als um einiges umfangreicher als geplant heraus und konnte somit nicht fertiggestellt werden. Sie wird in den nächsten Sprint übernommen.

5.3.3 Sprint Retrospektive

Was gut lief:

- Arbeit mit GitHub
- Zusammenarbeit
- Teamchemie
- Offene Kommunikation mit Problemen

- Hilfe des Teams bei Problemen
- Arbeits-Velocity weiter gestiegen

Verbesserungsbedarf:

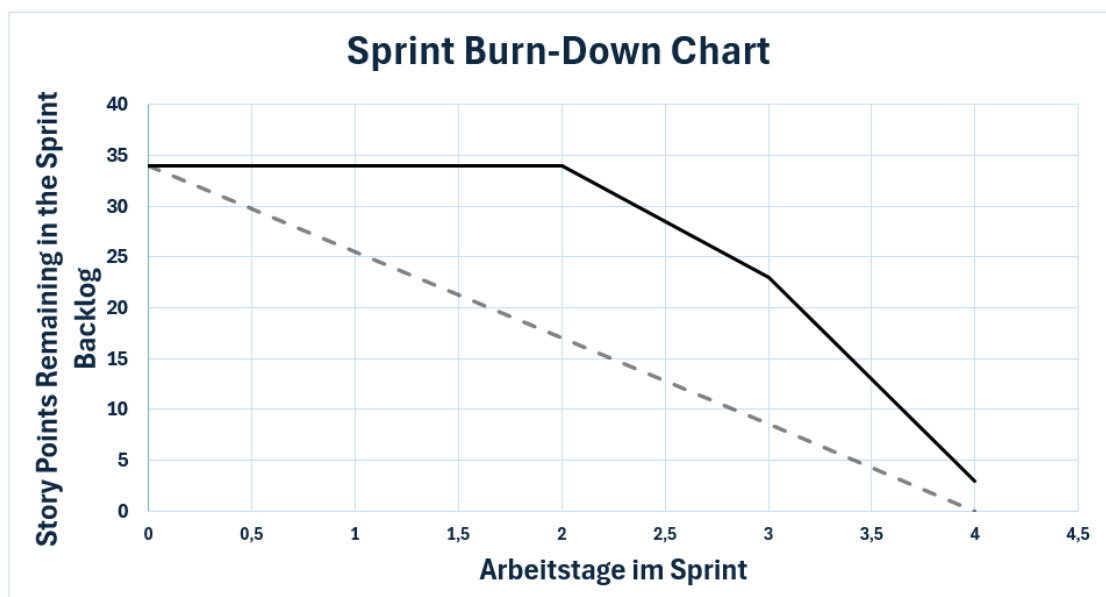
- Zu große User Storys
- Planung
- Schätzung der User Storys

Auflistung der Impediment Taskliste:

- Sensorenwerte

5.3.4 Sprint Zusammenfassung

Es wurden keine weiteren User Storys hinzugefügt.



Von 34 eingeplanten Story-Points wurden 31 komplett abgeschlossen.
Die Fertigstellung der Sensorenwerte konnte noch nicht erreicht werden wodurch 3 Story Points noch offen sind – sie werden in den nächsten Sprint übernommen.

Durch den Abschluss der Website, welche in diesem Sprint abgeschlossen wurde hat sich unsere Velocity enorm gesteigert.

Durchschnittliche Sprint Velocity über alle bisherigen Sprints: 19

5.4 Sprint 4

5.4.1 Sprintplanung

Dauer: 18.04.2024 – 16.05.2024

Ausgewählte User Stories:

- GUI-Linienfolgen: 10 SP, 10 VP
 - Als Benutzer möchte ich eine GUI haben die mir beim Linienfolgemodus anzeigt welchen Status die 4 Lichtsensoren haben.
- Linie folgen: 20 SP, 10 VP
 - Als Benutzer möchte ich einen Modus haben, bei dem der mBot2 automatisch einer Linie auf dem Boden folgt, um ein Rennen zu fahren.
- MBot Lagedarstellung: 20 SP, 4 VP
 - Als Benutzer möchte ich auf der Webseite ein 3D Modell des Mbots angezeigt bekommen, um die Lage im 3-dimensionalen Raum zu verbildlichen.

Ausgewählte Punkte aus der Impediment Liste:

- Sensorenwerte: 3 SP, 7 VP
 - Als Benutzer möchte ich in der UI sämtliche Sensoren angezeigt und in regelmäßigen Abständen aktualisiert bekommen, um die Daten auszuwerten.

Anzahl Story points: 53

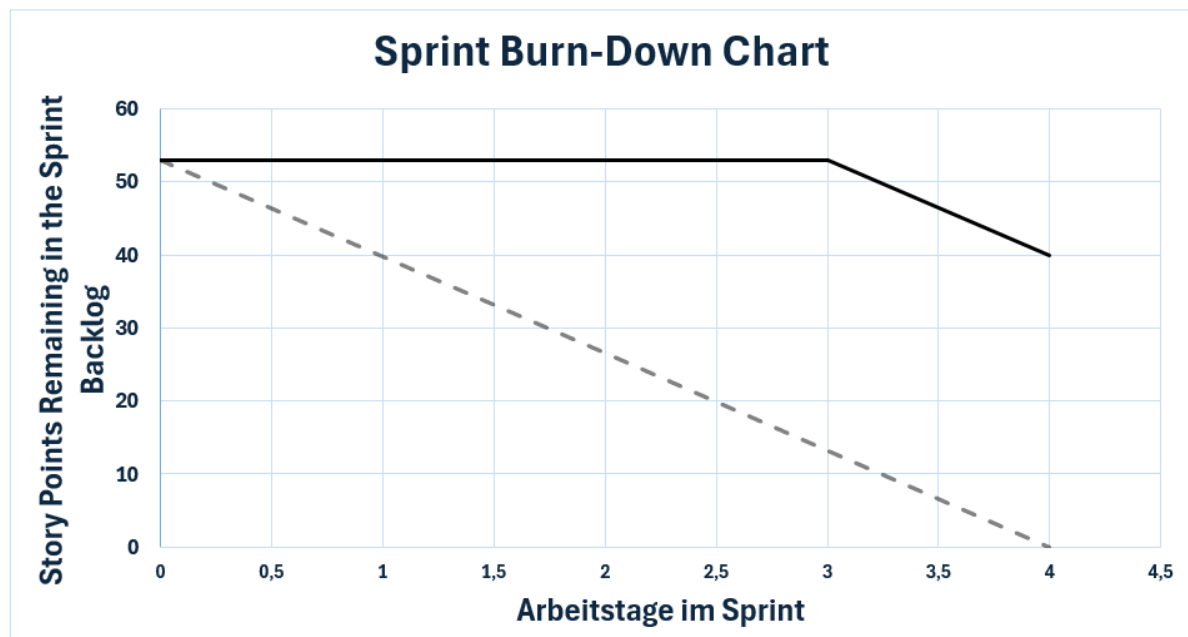
5.4.2 Sprint Demo

Die User Storys Sensorenwerte (aus Sprint 3 übernommen) und GUI-Linienfolgen konnten erfolgreich abgeschlossen werden. Auf der Website werden jetzt die einzelnen Sensorwerte, welche von den MBot Sensoren über UDP laufend an den Controller geschickt werden, detailliert angezeigt und ca. Jede Sekunde aktualisiert. Durch 4 Lichtsensoren werden extra auf der Website die Farbe unter den Sensoren über eine GUI angezeigt.

Die User Storys Linien folgen und MBot Lagedarstellung konnten nicht fertig gestellt werden und werden in Sprint 5 übernommen.

5.4.3 Sprint Zusammenfassung

Es wurden keine weiteren User Storys hinzugefügt.



Von 53 geplanten Story-Points wurden 13 komplett abgeschlossen. Die User Storys MBot Lagedarstellung und Linien folgen und somit 40 Story-Points werden in den nächsten Sprint übernommen.

Die durchschnittliche Sprint Velocity nimmt somit wieder ab und liegt nun bei 17,5.

5.5 Sprint 5

5.5.1 Sprintplanung

Dauer: 17.05.2024 – 06.06.2024

Ausgewählte User Stories:

- LEDs: 3 SP, 1VP
 - Als Benutzer möchte ich die 4 LEDs der Controllereinheit über die UI steuern können.

Ausgewählte Punkte aus der Impediment Liste:

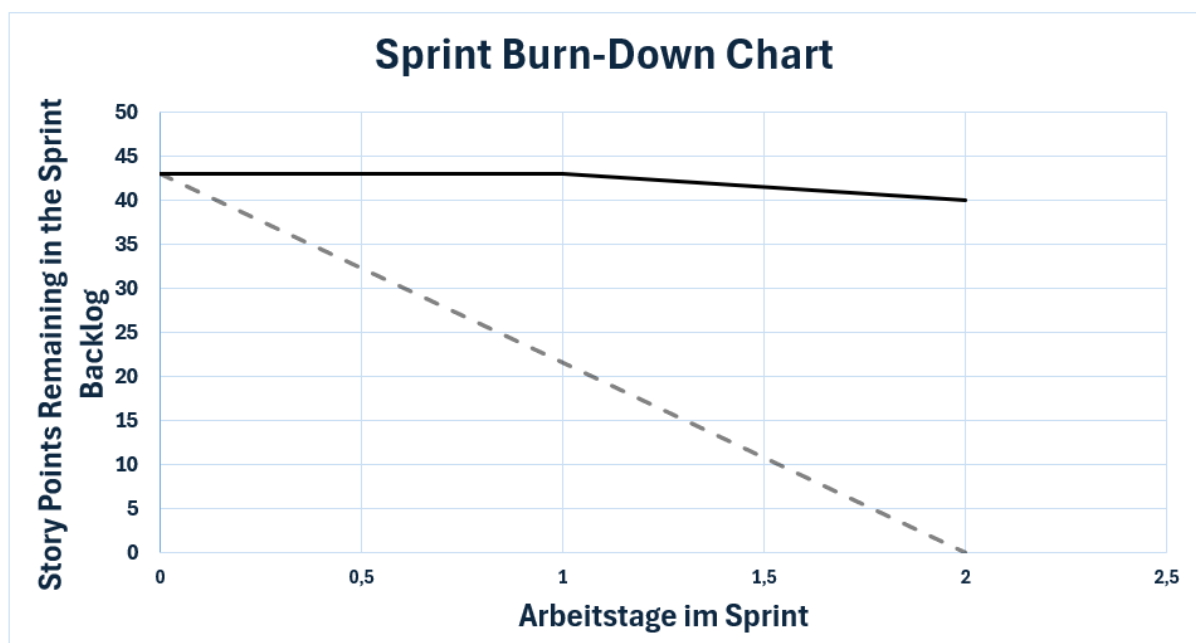
- Linie folgen: 20 SP, 10 VP
 - Als Benutzer möchte ich einen Modus haben, bei dem der mBot2 automatisch einer Linie auf dem Boden folgt, um ein Rennen zu fahren.
- MBot Lagedarstellung: 20 SP, 4 VP
 - Als Benutzer möchte ich auf der Webseite ein 3D Modell des Mbots angezeigt bekommen, um die Lage im 3-dimensionalen Raum zu verbildlichen.

Anzahl Story-Points: 43

5.5.2 Sprint Demo

Die User Storys Linienfolgen und Lagedarstellung (aus Sprint 4 übernommen) konnten leider wieder nicht abgeschlossen werden. Die User Story LEDs konnte erfolgreich abgeschlossen werden und der Benutzer kann nun die LEDs des mBots mithilfe eines Colour Pickers verändern.

5.5.3 Sprint Zusammenfassung



Von 43 geplanten Story-Points wurden 3 komplett abgeschlossen. Die User Storys MBot Lagedarstellung und Linien folgen und somit 40 Story-Points konnten nicht abgeschlossen werden.

Die durchschnittliche Sprint Velocity nimmt somit wieder ab und liegt nun bei 14,6.

6. Installation / Software deployment

Um das fertige Programm zu verwenden, benötigt man einen Server, auf dem die Spring Boot Applikation läuft, weiter wird ein mBot mit dem richtigen Micro Python Skript benötigt. Zuletzt braucht man nur noch ein Gerät, mit dem man die Webseite aufrufen kann und schon kann man unser Programm benutzen. Alle Komponenten müssen sich im gleichen Netzwerk befinden damit sie kommunizieren können.

7. Projektabschluß

7.1 Projektzusammenfassung

Durch das SYP-Projekt 2024 konnten viele erlernte Fähigkeiten fächerübergreifend verwendet werden und die Theorie in einem praktischen Beispiel angewendet werden. Insgesamt zeigte das Projekt gut wie die Kommunikation zwischen Server, Client und Gerät funktioniert und implementiert werden kann.

Bei der Durchführung liefen einige Sachen gut, so war es eine große Hilfe, dass wir vor Projektstart ein Demoprojekt erstellten und uns mit den Funktionen des mBots auseinandersetzten und experimentierten. Auch das Team funktionierte gut und jeder konnte jedem helfen, dies wurde auch bei den Sprint Reviews bewiesen, diese liefen alle gut. Im Vergleich zum Projekt aus dem dritten Jahrgang ist der Umgang mit GitHub positiv hervorzuheben und es traten keine großen Fehler durch die Versionsverwaltung auf.

Jedoch muss man auch negative Punkte nennen, der größte ist wohl die Tatsache, dass das Arbeitspaket mit der höchsten Priorität nicht fertiggestellt werden konnte. Weiters war die Schätzung der Arbeitspakete nicht sehr hilfreich und ungenau, dass wiederum führte auch zu Verzögerungen und Unklarheiten.

Alles in allem war das Projekt trotzdem ein Erfolg da die Theorie in der Praxis umgesetzt werden konnte und es ein erstes Projekt mit Hardware war. Man erlernt die Kommunikation zwischen den verschiedenen Komponenten und die Ansteuerung und Auswertung verschiedenster Sensorendaten.