



Softwareprojekt-Engineering, SS 2025

Lastenheft - LEGO Roboter Projekt

Lastenheft der Projektgruppe G12 im 2. Semester

Gruppe 12

Autor

Tim Peko

Korrekturleser

Tim Wahlmüller

Teammitglieder

Moritz Kieselbach
Tim Wahlmüller
Tim Peko
Alexander Kranl
Alexandra Usuanlele

Inhaltsverzeichnis

| | |
|--|----|
| Einleitung | 5 |
| 1 Einführung in das Projekt | 6 |
| 1.1 Stakeholder | 6 |
| 1.1.1 Kursleiter FH-Prof. DI Dr. Herwig Mayr | 6 |
| 1.1.2 Partnergruppe G11 | 6 |
| 1.1.3 Andere Projektgruppen | 6 |
| 1.1.4 Koordinationsgruppe | 6 |
| 1.2 Zeitlicher Rahmen | 6 |
| 2 Ausgangssituation (Istzustand) | 7 |
| 3 Aufgabenstellung (Sollzustand) | 8 |
| 3.1 Grundanforderungen | 8 |
| 3.1.1 Stapel | 8 |
| 3.2 Rahmenbedingungen | 8 |
| 3.2.1 Hardwarekomponenten | 8 |
| 4 Schnittstellenanforderungen | 10 |
| 4.1 Physische Schnittstellen | 10 |
| 4.1.1 Lego-Blöcke | 10 |
| 4.1.2 Aufbau | 10 |
| 4.1.2.a Steinebereich | 10 |
| 4.1.2.b Zielbereich | 11 |
| 4.1.2.c Sonstiges | 11 |
| 4.1.3 Markierungen | 11 |
| 4.2 Menschliche Interaktion | 11 |
| 4.3 Programmierung | 11 |
| 5 Systemanforderungen | 12 |
| 5.1 Hardware | 12 |
| 5.2 Software | 12 |

| | |
|---|----|
| 5.3 Implementierung | 12 |
| 6 Betriebsanforderungen | 13 |
| 6.1 Bedienung | 13 |
| 6.2 Untergrund | 13 |
| 6.3 Beleuchtung | 13 |
| 6.4 Betriebsdauer | 13 |
| 6.5 Fehlerbehandlung | 13 |
| 6.5.1 Typische Fehlerfälle | 13 |
| 7 Qualitätsanforderungen | 15 |
| 7.1 Genauigkeit | 15 |
| 7.2 Geschwindigkeit | 15 |
| 7.3 Zuverlässigkeit | 15 |
| 7.4 Sicherheit | 15 |
| 7.5 Ressourceneffizienz | 15 |
| 8 Anforderungen an die Projektentwicklung | 16 |
| 8.1 Entwicklungsmethodik | 16 |
| 8.2 Dokumentation | 16 |
| 8.3 Tests | 16 |
| 8.4 Kommunikationswege | 16 |
| 8.4.1 Kursleiter | 16 |
| 8.4.2 Partnergruppe G11 | 17 |
| 8.4.3 Gruppeninterne Kommunikation | 17 |
| 9 Szenario-Beschreibung | 18 |
| 10 Vorprüfung | 19 |
| 10.1 Machbarkeit | 19 |
| 10.2 Minimalität | 19 |
| 10.3 Eindeutigkeit | 19 |

| | |
|-------------------------------------|----|
| 10.4 Plausibilität | 19 |
| 10.5 Modifizierbarkeit | 19 |
| 10.6 Überprüfbarkeit | 19 |
| 11 Abnahmekriterien | 20 |
| 11.1 Dokumente | 20 |
| 11.2 Wettbewerb bzw. Testtage | 20 |
| Anhang | 21 |
| Begriffe | 21 |
| Teammitglieder der Gruppe G12 | 21 |

Einleitung

Im Rahmen des Sommersemesters 2025 wird im Modul Software-Projekt Engineering 2 ein Projekt durchgeführt. Dieses Projekt dient als Übung, um die Kursteilnehmer mit klassischem Projektmanagement für Softwareprojekte vertraut zu machen und sie auf organisatorische Strukturen und Konventionen in der freien Wirtschaft vorzubereiten.

Das **Projektteam** in dieser Projektausführung ist die Gruppe **G12**. Dessen **Partnergruppe** ist **G11**. Die **Partnergruppen** müssen sich gemeinsame *Ressourcen* teilen. In diesem Projekt wird das ein Lego-Mindstorms-Roboter sein. Unter Industriebedingungen könnte dies ein größerer Rechenserver sein, dessen *Ressourcen* auf mehrere Teams verteilt werden müssen.

Die konkreten Teammitglieder des **Projektteams G12** werden im Anhang aufgeführt.

1 Einführung in das Projekt

Dieses Projekt umfasst die Entwicklung eines Lego-Mindstorms-Roboters, der in der Lage sein muss, Lego-Blöcke von einem erhöhten *Steinebereich* auf einen niedrigeren *Zielbereich* zu transportieren und dabei nach Farben sortiert zu stapeln. Der Roboter muss autonom arbeiten und seine Aufgabe ohne menschliches Eingreifen erfüllen können.

1.1 Stakeholder

Stakeholder sind Personen, die an einem Projekt direkt oder indirekt interessiert sind und dessen Ergebnisse beeinflussen können.

Im Rahmen dieses Projekts wurden folgende *Stakeholder* identifiziert:

1.1.1 Kursleiter FH-Prof. DI Dr. Herwig Mayr

Der **Kursleiter** FH-Prof. DI Dr. Herwig Mayr stellt die Aufgabenstellung, die Ziele des Projekts sowie die benötigten *Ressourcen* und Rahmenbedingungen bereit. Er fungiert zugleich als Auftraggeber des Projekts.

1.1.2 Partnergruppe G11

Die **Partnergruppe G11** arbeitet parallel an derselben Aufgabenstellung und teilt die *Ressourcen* mit der Gruppe **G12**. Dies erfordert eine enge Kooperation zur gemeinsamen Nutzung des Lego-Mindstorms-Roboters und eine sorgfältige Abstimmung der Arbeitszeiten und -abläufe zwischen den Gruppen.

1.1.3 Andere Projektgruppen

Die anderen Projektgruppen arbeiten an der gleichen Aufgabenstellung unter vergleichbaren Umständen und Voraussetzungen. Der Erfolg jeder einzelnen Ausführung des Projektes wird in einem *Wettbewerb* gegen die anderen Projektgruppen entschieden.

1.1.4 Koordinationsgruppe

Die **Koordinationsgruppe** vertritt die Gesamtheit aller Projektgruppen und fungiert als offizielle Schnittstelle zwischen diesen und dem **Kursleiter**. Sie definiert in etwa Bewertungskriterien, genaue Umgebungsmaße oder spricht allgemeine Probleme mit dem **Kursleiter** an.

1.2 Zeitlicher Rahmen

Das Projekt folgt einem strukturierten Zeitplan mit drei definierten Meilensteinen, auch *Meilensteintage* genannt:

- 2025-05-08: Der erste Meilenstein am 08.05.2025 markiert die Fertigstellung einer lauffähigen *Alpha-Version* des Roboters. Diese Version sollte die grundlegende Funktionalität demonstrieren, muss aber noch nicht alle Anforderungen vollständig erfüllen.
- 2025-05-22: Der zweite Meilenstein am 22.05.2025 beinhaltet die Fertigstellung einer *Beta-Version*, die bereits alle funktionalen Anforderungen erfüllt, aber noch Optimierungsbedarf bei nicht-funktionalen Anforderungen haben kann.
- 2025-06-18: Der dritte und finale Meilenstein am 18.06.2025 ist der Tag des finalen *Wettbewerbs* und zugleich der Abgabetermin für das Projekt. Zu diesem Zeitpunkt muss das Projekt vollständig fertiggestellt sein und alle funktionalen sowie nicht-funktionalen Anforderungen erfüllen.

2 Ausgangssituation (Istzustand)

Aktuell existiert weder eine Lösung noch ein Prozess für das im Projekt geschilderte Problem. Es ist unabhängig und separiert von anderen Prozessen in der Anwendungsdomäne und kann daher nicht auf ein bestehendes System aufbauen bzw. dieses erweitern.

Der für das Projekt zu verwendende Lego-Mindstorms-Roboter wird aus einem vergangenen Projekt übernommen. Seine Grundkonfiguration ist bereits vorhanden, muss jedoch an die spezifischen Anforderungen dieses Projekts angepasst werden.

3 Aufgabenstellung (Sollzustand)

Im Rahmen dieses Projekts ist eine automatisierte Lösung für das Stapeln von Lego-Steinen nach Farbe zu entwickeln. Der zu entwickelnde Roboter muss in der Lage sein, Lego-Blöcke anhand ihrer Farbe zu erkennen, diese von einem erhöhten *Steinebereich* zu greifen und auf einem *Zielbereich* nach Farben sortiert zu stapeln. Diese Aufgabe muss autonom erledigt werden.

3.1 Grundanforderungen

3.1.1 Stapel

Die gebildeten Stapel dürfen nur genau eine Farbe beinhalten. Sie müssen stabil stehen bleiben, ansonsten gibt es keine Präzisionsanforderungen. Eine maximale Stapelhöhe ist nicht vorgegeben, wobei ein möglichst hoher Stapel anzustreben ist.

| Messgröße | Wertvorgabe |
|----------------------------|--|
| Höhe | Minimal: 1 Block Maximal: unbegrenzt |
| Abstand zu anderen Stapeln | Minimal: 0 cm |
| Grundfläche | Genau: 1x1 Block |
| Anzahl Farben | Genau: 1 |

Tabelle 1: Stapeldimensionen

Die einzelnen Blöcke eines Stapels dürfen in jeglicher Rotation platziert werden. Ein von der *Arbeitsfläche* heruntergefallener Block gilt als verloren und darf nicht wieder aufgehoben werden.

3.2 Rahmenbedingungen

Die folgenden Rahmenbedingungen sind für die Durchführung des Projekts festgelegt. Nicht explizit definierte Bedingungen können von der Projektgruppe in Absprache mit dem **Kursleiter** selbst festgelegt werden.

3.2.1 Hardwarekomponenten

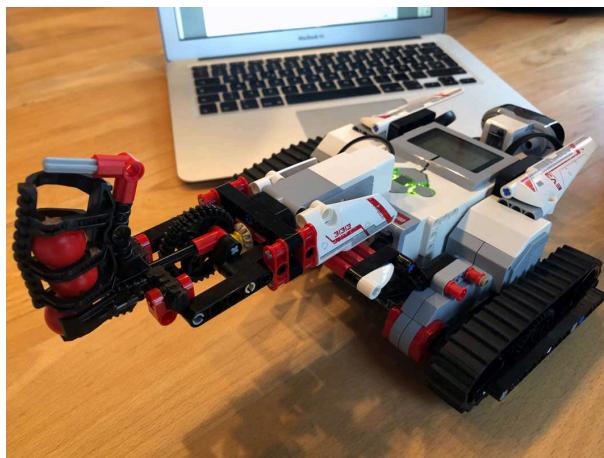


Abbildung 1: Reale Darstellung eines Lego-Mindstorms-Roboters

Als Basis für den Roboter dient ein Lego-Mindstorms-EV3-Mono-Brick, der mit Batterien betrieben wird. Folgende Peripheriegeräte sind ebenfalls verfügbar:

- 3x Motoren
- 1x Ultraschallsensor
- 1x Farbsensor

- 1x Gyrosensor

Zusätzlich steht ein Lego-Technik-Set zur Verfügung, das für den Aufbau des Roboters genutzt werden kann.

Das Hinzufügen neuer Teile oder Komponenten ist nicht erlaubt. Der **Kursleiter** bietet auf Anfrage die Möglichkeit, kaputte oder fehlende Teile zu ersetzen.

4 Schnittstellenanforderungen

Dieses Kapitel beschreibt die Schnittstellen, die der Roboter unterstützen muss, um seine Aufgabe erfüllen zu können.

4.1 Physische Schnittstellen

Der Roboter muss physisch mit seiner Umgebung interagieren können. Die Konstruktion des Roboters muss robust genug sein, um seine Aufgaben zuverlässig ausführen zu können, auch wenn leichte Variationen in der Umgebung oder den Eigenschaften der Blöcke auftreten.

4.1.1 Lego-Blöcke

Die Lego-Blöcke haben die gleiche Form und Größe.

| Größe | Wert |
|--------|----------|
| Höhe | ~3,15 cm |
| Breite | ~3,3 cm |
| Länge | ~3,3 cm |

Tabelle 2: Lego-Blöcke Dimensionen

Sie unterscheiden sich nur in der Farbe.

Die konkreten Farben werden am *Alpha-Testtag* basierend auf der Zuverlässigkeit der Sensoren festgelegt. Es kann von $3 \leq n \leq 10$ Farben ausgegangen werden.

Die Blöcke auf dem *Steinebereich* sind niemals verdeckt, bedeutet es gibt keine Situation, in der ein Block nur teilweise oder von hinten durch einen anderen Block verdeckt wird.

4.1.2 Aufbau

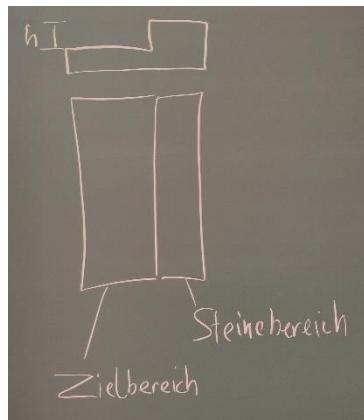


Abbildung 2: Skizze des Umgebungsaufbaus vom **Kursleiter**

4.1.2.a Steinebereich

Die Lego-Blöcke befinden sich auf einem erhöhten, ebenen, geraden, rechteckigen *Steinebereich* (z.B. ein Holzbalken), der auf einem Holztisch platziert ist.

Die Blöcke sind in einer ungefähr geraden Linie angeordnet, wobei nicht von einer exakten geraden Linie ausgegangen werden kann. Die Reihenfolge der Blöcke ist zufällig und folgt keinem Muster.

Ansonsten können auch sonst keine genauen Angaben über die Position der Blöcke gemacht werden.

4.1.2.b Zielbereich

Der rechteckige *Zielbereich* befindet sich neben dem *Steinebereich* auf demselben Untergrund, auf dem sich auch der Roboter befindet. Er ist also auch gleichzeitig die *Arbeitsfläche* und wird so groß gewählt, dass genügend Platz für den Roboter und die zu bildenden Stapel ist.

Es ist zu beachten, dass bei größeren *Arbeitsflächen* möglicherweise mehrere Tische zusammengestellt werden, was zu leichten Unebenheiten führen kann.

4.1.2.c Sonstiges

Folgende Aspekte müssen erst noch genau festgelegt werden:

- Beschaffenheit des Tisches
- Dimensionen des Tisches, *Steinebereich* und *Zielbereich*
- Höhdifferenz zwischen *Steinebereich* und *Zielbereich*

4.1.3 Markierungen

Es gibt keine Markierungen auf dem Tisch, die der Roboter als Orientierung verwenden könnte. Diese dürfen auch nicht vom **Projektteam** angebracht werden.

4.2 Menschliche Interaktion

Der Roboter ist bereits vorprogrammiert und muss nur noch analog, manuell von einem Benutzer gestartet werden. Danach läuft der Roboter autonom ohne menschlichen Eingriff. Beim *Wettbewerb* wird das Programm von einem *Tutor* gestartet.

Der Roboter muss kein Feedback an den Benutzer geben. Es ist keine Statusanzeige oder Ton bei Schlüssel- oder Endereignissen (wie Beendigung des Durchlaufs) zu implementieren.

4.3 Programmierung

Für die Programmierung wird eine Standard-EV3-Umgebung vorausgesetzt, wobei PyBricks oder vergleichbare Programmierumgebungen eingesetzt werden können. Die Wahl der Programmiersprache liegt im Ermessen der Projektgruppe und sollte auf Basis der verfügbaren Kenntnisse und der Eignung für die Aufgabe getroffen werden.

5 Systemanforderungen

Die Systemanforderungen definieren die Hardware- und Softwarevoraussetzungen sowie die an das zu entwickelnde System.

5.1 Hardware

Für die Umsetzung des Projekts wird ein Lego-Mindstorms-EV3-Roboter mit allen notwendigen Komponenten benötigt. Zur Programmierung des Roboters wird benötigt:

1. eine Micro-SD-Karte vom Typ SDHC (4GB - 32GB)
2. einen Laptop für die Entwicklung der Programme
3. ein Mini-USB-Kabel zur Übertragung

Die zulässigen Maße für den zusammengebauten Roboter werden später genauer festgelegt. Es ist jedoch zu erwarten, dass diese praktisch irrelevant sind, solange der Roboter durch die Tür des *Audimax* passt.

5.2 Software

Es wird eine geeignete Entwicklungsumgebung zur Programmierung des Lego-Mindstorms EV3 benötigt. Die Wahl der konkreten Programmiersprache liegt im Ermessen der Projektgruppe und sollte auf Basis der verfügbaren Kenntnisse und der Eignung für die Aufgabe getroffen werden.

5.3 Implementierung

Es besteht keine Anforderung an den konkreten Ablauf eines Durchlaufs. Es muss lediglich die grundlegende Funktionalität umgesetzt werden.

6 Betriebsanforderungen

Die Betriebsanforderungen beschreiben die Bedingungen, unter denen der Roboter betrieben werden soll, sowie Anforderungen an seine Laufzeit und Fehlerbehandlung.

6.1 Bedienung

Der Roboter muss seine Aufgabe ohne zusätzliche manuelle Eingriffe ausführen können. Davon ausgenommen sind:

- Aufsetzen der Betriebsumgebung
- Positionierung des Roboters
- Programmstart

6.2 Untergrund

Der Roboter ist darauf ausgelegt, unter folgenden Untergrundbedingungen zu arbeiten. Die Funktionsfähigkeit muss auf anderen Untergründen nicht gewährleistet werden.

- Holz (Tisch)
- Grundsätzlich eben
 - Kleine Spalten zwischen zusammengeschobenen Tischen sind zulässig
- Horizontal unverformt
 - Nicht korvex
 - Nicht konkav
- Nicht rutschig
- Nicht relevant schief (gemessen nach Wasserwaage)

6.3 Beleuchtung

Die Umgebung ist aktiv beleuchtet. Die Beleuchtung des *Audimax* wird so eingestellt, dass eine möglichst hohe Helligkeit erreicht wird. Da die Vorhänge allerdings manchmal geschlossen und geöffnet werden, ist eine gewisse Variation der Beleuchtung zu erwarten.

6.4 Betriebsdauer

Der Roboter muss in der Lage sein, seinen Arbeitsauftrag einmalig vollständig durchzuführen. Es gibt keine explizite Anforderung an eine ununterbrochene Betriebsdauer über diesen einzelnen Durchlauf hinaus.

6.5 Fehlerbehandlung

Der Roboter muss tendenziell nicht in der Lage sein, im Fehlerfall automatisch einen Neustart auszuführen. Es muss nicht auf verschiedene Fehlerarten (Prozess-, Umgebungs-, Bedienungsfehler) reagiert werden.

Je nach Fehlerart und -grad ist dies dennoch sinnvoll, um seine Aufgabe erfüllen zu können. Die Implementierung der Fehlerbehandlung erfolgt daher nach den besten Möglichkeiten anhand der Ressourcen des durchführenden **Projektteams**.

6.5.1 Typische Fehlerfälle

Zu den typischen Fehlerfällen, die berücksichtigt werden sollten, gehören:

- **Fehlgeschlagene Farberkennung:** Der Roboter kann die Farbe des Blocks unabhängig von der Ursache nicht erkennen.

-
- **Vorgabe:** Jeder Block ist eindeutig erkennbar, es kann also keine Fehlerbehandlung für diesen Fall stattfinden.
 - **Heruntergefallener Block:** Ein Block fällt während des Transports herunter.
 - Mögliche Maßnahme: Es sollte versucht werden, mit dem nächsten Block fortzufahren, anstatt zu versuchen, den verlorenen Block wiederzufinden.
 - **Kollision mit Hindernissen:** Der Roboter droht mit Hindernissen zu kollidieren.
 - Mögliche Maßnahme: Der Roboter fährt aus der Kollisionsrichtung aus.
 - Mögliche Maßnahme: Der Roboter führt einen Notstopp durch.
 - **Instabilität des Stapels:** Es wird erkannt, dass ein Stapel instabil wird.
 - Mögliche Maßnahme: Der oberste Lego-Block wird neu positioniert.
 - Mögliche Maßnahme: Es wird ein neuer Stapel begonnen.

Detaillierte Anforderungen zur Fehlerbehandlung sind innerhalb des **Projektteams** abzustimmen und sollten in das *Pflichtenheft* aufgenommen werden.

7 Qualitätsanforderungen

Die Qualitätsanforderungen definieren die nicht-funktionalen Aspekte des Systems, die für einen erfolgreichen Betrieb erforderlich sind. Allgemein kann angenommen werden, dass Prioritäten Genauigkeit = Zuverlässigkeit > Geschwindigkeit > Ressourceneffizienz gelten.

Zusätzlich hat das **Projektteam** im Vorsemester bereits eigene Qualitätsmerkmale definiert. Diese müssen im Rahmen dieses Projekts überprüft werden. Ausmaß und Priorität wurden zum aktuellen Zeitpunkt noch nicht festgelegt.

7.1 Genauigkeit

Die Genauigkeit beim Stapeln der Lego-Blöcke muss ausreichend sein, damit die Stapel nicht umfallen und stabil stehen bleiben, auch nachdem der Roboter seine Aufgabe beendet hat. Dies erfordert eine präzise Steuerung der Bewegungen des Roboters und eine genaue Positionierung der Blöcke beim Stapeln.

7.2 Geschwindigkeit

Der Roboter bekommt einen fix definierten Zeitslot beim *Wettbewerb* zur Verfügung. Dieser Zeitslot darf nicht überschritten werden und wird später genauer bekanntgegeben.

7.3 Zuverlässigkeit

Der Roboter darf eine Fehlerrate von 100% haben, anzustreben ist eine Fehlerrate von 0%. Die akzeptable Fehlertoleranz wird durch den *Wettbewerb*, also den Vergleich zu anderen Projektgruppen, entschieden. Die absolute Fehlerrate hat keinen Einfluss auf die Bewertung des Projekterfolgs.

Die Robustheit des Systems gegenüber leichten Variationen in der Umgebung oder den Eigenschaften der Blöcke ist ein wichtiger Aspekt der Zuverlässigkeit und sollte durch umfangreiche Tests sichergestellt werden.

7.4 Sicherheit

Es sind keine Sicherheitsfunktionen wie etwa ein Notstopp-Knopf oder eine Pause-Funktion erforderlich.

Es muss jedoch verhindert werden, dass der Roboter selbst von der *Arbeitsfläche* fällt. Dieses Ereignis stellt einen Fehldurchlauf dar.

7.5 Ressourceneffizienz

Es gibt keine Limits für den Ressourcenverbrauch in einem einzelnen Durchlauf. Der Roboter muss lediglich in der Lage sein, seinen Arbeitsauftrag einmalig vollständig durchzuführen.

8 Anforderungen an die Projektentwicklung

Dieses Kapitel beschreibt die methodischen und organisatorischen Anforderungen an die Durchführung des Projekts.

8.1 Entwicklungsmethodik

Die Entwicklung orientiert sich an einem klassischen (nicht agilen) Vorgehensmodell. Im Rahmen des Projekts sind ein *Lastenheft* sowie ein *Pflichtenheft* zu erstellen. Zusätzlich müssen ein Organigramm sowie Stellenbeschreibungen des **Projektteams** erarbeitet werden, um die Verantwortlichkeiten und Kommunikationswege klar zu definieren.

8.2 Dokumentation

Neben den im obigen Abschnitt 8.1 angeführten Dokumenten müssen einzelne Dokumente im Rahmen der Lehrveranstaltung erstellt werden. Die konkreten Anforderungen an die Dokumente werden dabei näher definiert. Zusätzliche Dokumentation erfolgt im Eigeninteresse des **Projektteams**, um ein möglichst reibungsloses Projektmanagement zu gewährleisten.

Informationen, die erst zu einem späteren Zeitpunkt bekannt gegeben werden, sollten im *Pflichtenheft* aufgenommen werden, können aber auch separat festgehalten werden.

Alle Dokumente, die im Rahmen dieses Projekts erstellt werden, müssen sich an folgende Richtlinien halten:

- Konsistentes Layout
- Befolgen des Namensschemas
- Korrekte Versionierung x.y
 - x Inkrement für offiziell publizierte oder abgegebene Versionen
 - y Inkrement für im **Projektteam** intern publizierte Versionen
- Deckblatt bei mehr als 3 Seiten
- Korrekturlesung durch ein anderes Teammitglied

8.3 Tests

Die Tests werden vom **Projektteam** manuell in einer selbst nachgestellten Umgebung durchgeführt. Die Exzessivität der Tests sollte vom **Projektteam** selbstständig so gewählt werden, dass eine qualitative Umsetzung des Projekts sichergestellt werden kann.

8.4 Kommunikationswege

Es gibt bevorzugte Kommunikationswege für die verschiedenen *Stakeholder*.

8.4.1 Kursleiter

| Kommunikationsweg | Zeitlicher Rahmen | Unter welchen Umständen |
|---------------------|---|--|
| Mündlich | Unmittelbar zu Vorlesungen oder Übungen | Bei schnellen Fragen oder Anliegen |
| E-Mail | <ul style="list-style-type: none">• Rund um die Uhr• Antwortet meist innerhalb von 24h | <ul style="list-style-type: none">• Bei weniger dringenden Anliegen• Bei umfangreicheren Themen |
| Koordinationsgruppe | Hauptsächlich zu geplanten Besprechungen | Bei Anliegen (fast) aller Projektgruppen betreffend |

Tabelle 3: Kommunikationswege mit **Kursleiter**

Der **Kursleiter** ist grundsätzlich mit der Festlegung zusätzlicher Kommunikationswege einverstanden. Dies erfolgt auf Absprache und wird von der Projektgruppe **G12** initiiert.

Der **Kursleiter** versucht, typisches Kommunikationsverhalten von realen Auftraggebern zu imitieren. Das beinhaltet:

- Manchmal zu beschäftigt für eine Antwort
- Schlechtere Antworten bei unpräzisen Fragen
- ...

8.4.2 Partnergruppe G11

Grundsätzlich ist die Kommunikation mit der **Partnergruppe G11** der behandelnden Gruppe **G12** überlassen. Es muss sich hierbei selbst koordiniert werden.

8.4.3 Gruppeninterne Kommunikation

Die gruppeninterne Kommunikation unterliegt dem gruppeneigenen Ermessen. Es gibt hierbei kein Standardprozedere oder andere Vorgaben.

9 Szenario-Beschreibung

Um die Funktionsweise des zu entwickelnden Roboters zu veranschaulichen, wird im Folgenden ein exemplarischer Ablauf der Aufgabe beschrieben:

1. **Initialisierung:** Der Roboter wird eingeschaltet und positioniert sich zwischen dem *Steinebereich* und dem *Zielbereich*. Er kalibriert seine Sensoren und bereitet sich auf die Aufgabe vor.
2. **Erkennung der Blöcke:** Der Roboter wendet sich dem *Steinebereich* zu und scannt die dort aufgereihten Lego-Blöcke. Er identifiziert die Positionen und Farben der einzelnen Blöcke.
3. **Greifvorgang:** Der Roboter fährt zum ersten Block, positioniert seinen Greifmechanismus und hebt den Block von dem *Steinebereich*.
4. **Transport:** Mit dem gegriffenen Block bewegt der Roboter seinen Arm in Richtung des *Zielbereichs*.
5. **Farbbasierte Entscheidung:** Basierend auf der erkannten Farbe des Blocks entscheidet der Roboter, an welcher Position auf dem *Zielbereich* der Block abgelegt werden soll, um nach Farben sortierte Stapel zu bilden.
6. **Ablage und Stapeln:** Der Roboter platziert den Block präzise auf der *Zielbereich*, entweder als Grundstein eines neuen Stapels oder auf einem bereits vorhandenen Stapel der gleichen Farbe.
7. **Wiederholung:** Der Roboter dreht sich zurück zum *Steinebereich*, um den nächsten Block zu greifen. Die Schritte 3-6 werden wiederholt, bis alle Blöcke transportiert und gestapelt sind.
8. **Abschluss:** Nach dem Transport des letzten Blocks fährt der Roboter in eine definierte Endposition und signalisiert die erfolgreiche Beendigung der Aufgabe.

Dieser Ablauf verdeutlicht die Komplexität der Aufgabe und die verschiedenen Teilschritte, die der Roboter bewältigen muss. Die genaue Implementierung kann je nach gewähltem Konstruktions- und Programmieransatz variieren.

Dies stellt keine geforderte Ablaufreihenfolge dar, sondern eine mögliche Umsetzung der Aufgabe.

10 Vorprüfung

Das Projekt wurde einer theoretischen Vorprüfung unterzogen, deren Ziel es war, die generelle Machbarkeit zu überprüfen. Aufgrund der überschaubaren Komplexität des Projekts wurde von einer vollständigen prototypischen Vorprüfung abgesehen. Stattdessen wurden die folgenden Aspekte theoretisch evaluiert:

10.1 Machbarkeit

Die Machbarkeit des Projekts wurde als realistisch eingestuft. Alle erforderlichen Komponenten (EV3 Brick, Sensoren, Motoren) stehen zur Verfügung, und der Roboter kann mit den vorhandenen Laptops programmiert werden. Ein Vergleich verschiedener Programmiersprachen und EV3-Umgebungen wurde bereits durchgeführt, um die optimale Kombination für dieses Projekt zu identifizieren. Zudem wurden ähnliche Projekte in früheren Semestern und von anderen Gruppen erfolgreich umgesetzt, was die grundsätzliche Machbarkeit bestätigt.

10.2 Minimalität

Die Aufgabenstellung wurde auf die wesentlichen Anforderungen reduziert: Erkennung verschiedener Farben, Transport von Lego-Steinen und Stapeln nach Farben. Diese Reduktion auf das Wesentliche trägt zur Klarheit der Aufgabe bei und erhöht die Wahrscheinlichkeit einer erfolgreichen Umsetzung.

10.3 Eindeutigkeit

Die Aufgabenstellung und Anforderungen wurden klar definiert und lassen wenig Interpretationsspielraum. Dies erleichtert die Umsetzung. Das Bewertungssystem für den Projekterfolg ist ebenfalls unmissverständlich definiert.

10.4 Plausibilität

Die Anforderungen wurden als plausibel und mit den vorhandenen *Ressourcen* umsetzbar bewertet. Die zur Verfügung stehenden technischen Mittel sind für die gestellte Aufgabe angemessen.

10.5 Modifizierbarkeit

Das Projekt erlaubt Anpassungen und Modifikationen, sowohl in der Hardware (Roboterkonfiguration) als auch in der Software. Dies ermöglicht eine flexible Reaktion auf unvorhergesehene Herausforderungen während der Entwicklung.

10.6 Überprüfbarkeit

Der Projekterfolg kann durch den finalen *Wettbewerb* eindeutig ermittelt werden. Die definierten funktionalen und nicht-funktionalen Anforderungen bieten klare Kriterien für die Überprüfung der Projektumsetzung.

11 Abnahmekriterien

Diese Projektausführung muss seine Anforderungen besser im Vergleich zu anderen Projektgruppen erfüllen. Es werden dafür die nachfolgenden Kriterien verwendet, wobei sich diese im Laufe der Zeit noch erweitern und konkretisieren können. Sie werden durch die **Koordinationsgruppe** und den **Kursleiter** festgelegt.

11.1 Dokumente

Geforderte Dokumente sind:

- *Lastenheft*
- *Pflichtenheft*
- *Organigramm*
- *Stellenbeschreibungen*

Zusätzlich gibt es individuell geforderte Dokumente.

Prinzipiell müssen alle Dokumente die im Abschnitt 8.2 angegebenen Kriterien erfüllen.

11.2 Wettbewerb bzw. Testtage

Genaue Kriterien werden erst am Tag des *Wettbewerbs* und am *Alpha-Testtag* bzw. *Beta-Testtag* entsprechend bekannt gegeben. Grob gestaltet sich die Bewertung wie folgt, wobei erstere Aspekte stärker gewichtet werden:

1. Erreichte Punkte steigen exponentiell mit der Höhe der Stapel
2. Die Einhaltung des zugeordneten Zeitslots entscheidet bei Gleichstand

Punkte können beim Testen in der offiziellen *Testumgebung* unter Aufsicht von *Tutoren* an Testtagen erzielt werden. Das geschieht unter den für diesen Tag festgelegten Kriterien.

Anhang

Begriffe

| | |
|-------------------------|--|
| Steinebereich | Fläche, auf der die Steine initial liegen; Fläche, von der die Steine entnommen werden |
| Zielbereich | Fläche, auf der die Steine nach Farben sortiert gestapelt werden |
| Arbeitsfläche | Fläche, auf der sich der Roboter befindet und er seine Aufgabe erfüllen muss; Vorraussichtlich die gleiche Fläche wie der <i>Zielbereich</i> |
| Alpha-Version | Erste lauffähige Version des Roboters mit grundlegender Funktionalität |
| Alpha-Testtag | Tag des ersten Testens der Alpha-Version in der Testumgebung |
| Beta-Version | Funktionsfähige Version des Roboters, die noch Optimierungsbedarf bei nicht-funktionalen Anforderungen hat |
| Beta-Testtag | Tag des Testens der Beta-Version in der Testumgebung |
| Meilensteintag | Tag, an dem die Projektgruppen ihre aktuelle Lösung vorstellen und testen; Tage des Wettbewerbs, Alpha- und Beta-Testens |
| Wettbewerb | Tag des finalen Wettbewerbs; Alle Projektgruppen treten mit ihrer Lösung gegeneinander an |
| Pflichtenheft | Dokument, das die vom Auftragnehmer umzusetzenden Anforderungen detailliert beschreibt |
| Lastenheft | Dokument, das die Gesamtheit der Anforderungen des Auftraggebers an die Lieferungen und Leistungen eines Auftragnehmers enthält |
| Ressourcen | Mittel, die zur Durchführung des Projekts benötigt werden, wie Hardware, Software, Zeit und Personal |
| Stakeholder | Personen oder Gruppen, die ein Interesse am Projektverlauf oder -ergebnis haben und diese direkt oder indirekt beeinflussen können |
| Projektteam | Die behandelnde Projektgruppe G12 , die diese hiermit dokumentierte Projektausführung umsetzt |
| Betriebsumgebung | Die Umgebung, in der der Roboter seine Aufgabe erfüllen muss; Umgebung beim Wettbewerb |
| Testumgebung | Umgebung, in der die Funktionalität des Roboters getestet wird; Umgebung offiziell von Kursleitung bereitgestellt |
| Audimax | Hörsaal, in dem typischerweise die Vorlesungen und Übungen des Kurses stattfinden; Ort des finalen Wettbewerbs |
| Tutor | Höhersemestrige Studenten, die den Kursleiter im Rahmen der Lehrveranstaltung unterstützen |

Teammitglieder der Gruppe G12

- Moritz Kieselbach
- Tim Wahlmüller
- Tim Peko
- Alexander Kranl
- Alexandra Usuanlele