**HTL Saalfelden**

**Systemplanung und Projektentwicklung**

****

**Projektdokumentation**

**2023 / 2024**

|  |  |
| --- | --- |
| **Projektbezeichnung** | MBot2 |
| **Projektteam** | Philipp Kirchtag, Sebastian Krallinger, Fabian Scharfetter, Eduard Voicescu |
| **Erstellt am** | 10.01.2024 |
| **Letzte Änderung am** | 21.02.2024 |
| **Status** | [in Bearbeitung] |
| **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 |

**Inhalt**

[1. Allgemeines / Projektübersicht 4](#_Toc164944490)

[1.1 Projektbeschreibung 4](#_Toc164944491)

[1.2 Projektteam und Schnittstellen 4](#_Toc164944492)

[2. Funktionale Anforderungen 4](#_Toc164944493)

[2.1 Use Cases 4](#_Toc164944494)

[2.1.1 Verbindung zwischen Roboter und Anwendung herstellen 4](#_Toc164944495)

[2.1.2 Sensordaten anzeigen 5](#_Toc164944496)

[2.1.3 Roboter steuern 5](#_Toc164944497)

[2.1.4 Sicherheitsmodus aktivieren/deaktivieren 5](#_Toc164944498)

[2.1.5 Netzwerkanbindung des Roboters 5](#_Toc164944499)

[2.1.6 Autonomer Linienfolgemodus 5](#_Toc164944500)

[2.1.7 LED-Farben einstellen 6](#_Toc164944501)

[2.1.8 Status der Lichtsensoren im Linienfolgemodus anzeigen 6](#_Toc164944502)

[2.1.9 Darstellung des mBot im 3D-Raum basierend auf dem Lagesensor 6](#_Toc164944503)

[3. Projektplanung 6](#_Toc164944504)

[3.1 Variantenbildung 7](#_Toc164944505)

[3.2 Machbarkeitsstudie 7](#_Toc164944506)

[3.3 Allgemeine Planungsinformationen 7](#_Toc164944507)

[3.4 Projektumfeldanalyse 7](#_Toc164944508)

[4. Softwarearchitektur 7](#_Toc164944509)

[4.1 Aktivitätsdiagramme 8](#_Toc164944510)

[4.1.1 Aktivitätsdiagramm 1 Name 8](#_Toc164944511)

[4.1.2 Aktivitätsdiagramm n Name 8](#_Toc164944512)

[4.2 Sequenzdiagramme 8](#_Toc164944513)

[4.2.1 Sequenzdiagramm 1 Name 8](#_Toc164944514)

[4.2.2 Sequenzdiagramm n Name 8](#_Toc164944515)

[4.3 Komponentendiagramme 8](#_Toc164944516)

[4.4 Verteilungsdiagramme 9](#_Toc164944517)

[4.5 Softwarekomponenten / Programme 9](#_Toc164944518)

[4.5.1 SW Programme 9](#_Toc164944519)

[5.5.2 SW Komponenten 9](#_Toc164944520)

[4.6 Anwendungsfalldiagramm 10](#_Toc164944521)

[5. Projektdurchführung 11](#_Toc164944522)

[5.1 Sprint 1 11](#_Toc164944523)

[5.1.1 Sprintplanung 11](#_Toc164944524)

[5.1.2 Sprint Demo 11](#_Toc164944525)

[5.1.3 Sprint Retrospektive 11](#_Toc164944526)

[5.1.4 Sprint Zusammenfassung 12](#_Toc164944527)

[5.2 Sprint 2 13](#_Toc164944528)

[5.2.1 Sprintplanung 13](#_Toc164944529)

[5.2.2 Sprint Demo 13](#_Toc164944530)

[5.2.3 Sprint Retrospektive 13](#_Toc164944531)

[5.2.4 Sprint Zusammenfassung 14](#_Toc164944532)

[5.3 Sprint 3 15](#_Toc164944533)

[5.3.1 Sprintplanung 15](#_Toc164944534)

[5.3.2 Sprint Demo 15](#_Toc164944535)

[5.3.3 Sprint Retrospektive 15](#_Toc164944536)

[5.3.4 Sprint Zusammenfassung 16](#_Toc164944537)

[5.4 Sprint 4 17](#_Toc164944538)

[5.4.1 Sprintplanung 17](#_Toc164944539)

[5.4.2 Sprint Demo 17](#_Toc164944540)

[5.4.3 Sprint Zusammenfassung 17](#_Toc164944541)

[5.5 Sprint 5 18](#_Toc164944542)

[5.5.1 Sprintplanung 18](#_Toc164944543)

[5.5.2 Sprint Demo 18](#_Toc164944544)

[5.5.3 Sprint Zusammenfassung 18](#_Toc164944545)

[6. Installation / Software deployment 19](#_Toc164944546)

[7. Projektabschluß 19](#_Toc164944547)

[7.1 Projektzusammenfassung 19](#_Toc164944548)

[7.2 Attachments 19](#_Toc164944549)

# Allgemeines / Projektübersicht

## 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.

## Projektteam und Schnittstellen

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Rolle(n)** | **Name** | **Telefon** | **E-Mail** | **Team** |
| Projektleiter | Fabian Scharfetter | 0664 1900355 | fabian.scharfetter@htl-saalfelden.at | 3 |
| Projektmitarbeiter | Philipp Kirchtag | 0664 75039947 | philipp.kirchtag@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 |

# Funktionale Anforderungen

## Use Cases

### 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.

### 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.

### 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.

### 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.

### 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).

### 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.

### 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.

### 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.

### 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.

# Projektplanung

In diesem Kapitel sollen grundlegende Fragen geklärt werden bevor mit der Projektdurchführung begonnen wird. Obwohl es klar ist, das dieses Projekt mit Hilfe von agilen PM Methoden durchgeführt wird, sollen vorab grundsätzliche Entscheidungen geklärt werden wie z.B. verwendete Programmiersprache, Variantenbildung, Betriebssystem, ...

Manche dieser Punkte können auch auf Grund der gegebenen Anforderungen als „gesetzt“ hingenommen werden.

## Variantenbildung

Auf Basis der Projektanforderungen, welche Varianten für die Umsetzung wurden ausgearbeitet und welche wurde schließlich gewählt und warum. Welche Varianten wurden verworfen und aus welchem Grund.

## 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.

Gab es einzelne Punkte, die vorher grob getestet wurden um zu Zeigen das die Umsetzung grundsätzlich möglich ist.

z.B. Kommunikation mit einem MQTT-Server mit Hilfe der Programmiersprache Python, ...

## Allgemeine Planungsinformationen

Andere Planungsinformationen welche nicht durch andere Kapitel abgedeckt sind werden hier eingetragen.

## Projektumfeldanalyse

Führen Sie eine Analyse des Projektumfeldes durch. Welche vergleichbaren Produkte gibt es bereits am Markt. Wie erfolgt die Abgrenzung zu diesen bereits bestehenden Produkten? Wer sind die relevanten Stakeholder des Projektes.

# Softwarearchitektur

In diesem Kapitel soll der Aufbau der Software/Hardware beschrieben werden.

aus welchen Komponenten besteht das SW-Produkt.

Wie interagieren die einzelnen Komponenten miteinander.

Auf welcher Hardware läuft das System bzw. handelt es sich vielleicht sogar um ein verteiltes System. Wie kommunizieren diese Komponenten miteinander.

## Aktivitätsdiagramme

### Aktivitätsdiagramm 1 Name

Beschreibung der entsprechenden Aktivität und einfügen des Aktivitätsdiagramm

### Aktivitätsdiagramm n Name

Beschreibung der entsprechenden Aktivität und einfügen des Aktivitätsdiagramm

## Sequenzdiagramme

### Sequenzdiagramm 1 Name

Beschreibung der entsprechenden Sequenz und Einfügen des Sequenzdiagramm

### Sequenzdiagramm n Name

Beschreibung der entsprechenden Sequenz und Einfügen des Sequenzdiagramm

## Komponentendiagramme

Aus welchen Komponenten besteht die Software und welche Schnittstellen bieten diese an.

## Verteilungsdiagramme

Ein Bild, das Text, Diagramm, Reihe, Plan enthält.

Automatisch generierte Beschreibung

(11.04.2024)

Zeigt an wie die einzelnen Teile der Software auf die Hardwarekomponenten verteilt sind und wie die Hardwarekomponenten miteinander verbunden sind.

Auf welchem Rechner läuft welcher Software. Wie sind diese über ein Netzwerk miteinander verbunden.

## Softwarekomponenten / Programme

### SW Programme

Auflistung aller verwendeten SW Programme die bei der Umsetzung des Projektes verwendet worden sind. inkl. Angabe der Versionsnummer

z.B.: Visual Studio 2022,...

### 5.5.2 SW Komponenten

Auflistung aller verwendeten SW Komponenten welche für den Betrieb der SW benötigt werden. z.B.: Java Version, Apache Webserver, DotNet Framework, SW Library XY

inkl. Versiosnummer, Hersteller, Bezugsquelle (Downloadlink, ...) und SW-Lizenz (GPL, LGPL, Apache License, ...

## Anwendungsfalldiagramm

Ein Bild, das Diagramm, Text, Zeichnung, Reihe enthält.

Automatisch generierte Beschreibung

# Projektdurchführung

## Sprint 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

### Sprint Demo

Die User Stories Steuerung PC sowie Satusanzeige 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 einiger Probleme nicht abgeschlossen und wird in den nächsten Sprint übernommen.

### 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

### Sprint Zusammenfassung

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

Burndownchart:

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

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

## Sprint 2

### 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

### 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.

### 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

### 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:

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

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

## Sprint 3

### 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

### 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.

### 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

### Sprint Zusammenfassung

Es wurden keine weiteren User Storys hinzugefügt.

Ein Bild, das Text, Reihe, Diagramm, Zahl enthält.

Automatisch generierte Beschreibung

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

## Sprint 4

### Sprintplanung

Dauer: 18.04.2024 – 08.05.2024

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: 34

### Sprint Demo

D

### Sprint Zusammenfassung

E

## Sprint 5

### Sprintplanung

D

### Sprint Demo

D

### Sprint Zusammenfassung

E

# Installation / Software deployment

Anleitung welche Schritte notwendig sind um das fertige SW Produkt zu installieren und in Betrieb zu nehmen.

# Projektabschluß

## Projektzusammenfassung

Zusammenfassung der Projektdurchführung. Was lief gut/schlecht. Welche Erkenntnisse wurden während der Durchführung des Projektes gewonnen. Was würde man, nun anders machen bzw. wieder gleich machen?

## Attachments

Tabellarische Auflistung der Projektdateien.

z.B.: ZIP-File mit dem Quellcode, Projektpräsentationen, ...