**[*Intelligente Parkplatzerkennung mit künstlichen neuronalen Netzwerken*]**

**Projektmanagement-Plan**

1. Projektorganisation 3
   1. Rollen & Zuständigkeiten 3
   2. Projektstrukturplan 3
2. Projektführung 5
   1. Rahmen- und Projektplan 5
3. Projektunterstützung 7
   1. Tools für Entwicklung 7
   2. Konfigurationsmanagement 7
   3. Tools zur Projektkommunikation und zum Dokumentenaustausch 7
4. Qualitäts- und Testplan 8
   1. Definition of Ready und Definition of Done 8
   2. Testdesign & Testautomatisierung 8
   3. Testfälle 8
5. Anhänge 9

Versionen:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Rev. | Datum | Autor | Bemerkungen | Status |
| 0.1 | 14.03.2019 | Felix Willrich | 1. Entwurf + Eintragen aller Informationen | Abgeschlossen |
| 0.2 | 20.03.2019 | Felix Willrich | Erste Bearbeitung der Rollen | In Bearbeitung |

# Projektorganisation

## Rollen & Zuständigkeiten

Zu Beginn des Projektes wurden die Stärken eines jeden Gruppenmitgliedes erörtert. Nachdem die Fähigkeiten, welche innerhalb der Gruppe zu Verfügung stehen, erfasst wurden, wurde ein Projektziel mit einem entsprechenden Produkt definiert. Aufbauend auf die Ziele dieses Projektes, wurde folgende Rollenverteilung innerhalb des Projektes vorgenommen:

|  |  |  |
| --- | --- | --- |
| Name | Rolle(n) | Beschreibung |
| Felix Willrich | Teamleiter, Projektmanagement, Entwickler | Projektleiter vom Projekt, Ansprechpartner vom Kunden,  Entwickler für Teilbereiche |
| Frederik Rieß | Co-Teamleiter, Chefentwickler | Co-Teamleiter vom Projekt,  Koordinierung der Entwicklung |
| Pit-Aurel Ehlers | Entwickler | Entwickler für Teilbereiche |
| Jascha Schmidt | Entwickler | Entwickler für Teilbereiche |

## Projektstrukturplan

**Ziel:**

**Erfassen von Aufgaben und ihren Abhängigkeiten**

**Vorbereitung für Aufwands- und Zeitplan**

Gliederung des Projekts in plan- und kontrollierbare Elemente. Strukturierung des Projekts nach einem Gesichtspunkt, z.B. welche Komponenten/Teilprodukte sind zu erstellen, oder nach Phasen/Projektaktivitäten oder Iterationen –

*<Zerlegen Sie Ihr Projekt bis zum ersten Release (14.11.) in Teilaufgaben und schließlich soweit, bis Sie Arbeitspakete ableiten können. >*

Ein Arbeitspaket ist eine geschlossene Aufgabenstellung, die zu einem festgelegten Zeitpunkt mit definiertem Ergebnis und Aufwand umgesetzt werden kann,

Einem Arbeitspaket sind zugeordnet

* ein (Haupt)Verantwortlichen
* inhaltliche Beschreibung
* die benötigten Ressourcen
* die Abhängigkeiten zu anderen Arbeitspaketen
* Zuordnung zu einem Meilenstein

 Beispiel Projektstrukturplan

Bildergebnis für Projektstrukturplan

Abbildung 1: Beispiel Projektstrukturplan, © Projekte-leicht-gemacht

# Projektführung

## Rahmen- und Projektplan

Der grobe Ablaufplan ist durch die Terminvorgaben des Teamprojekts in wesentlichen Teilen festgelegt:

* Architektur: 24.10.
* 1. Release: 14.11.
* 2. Release: 28.11.
* 3. Release: 19.12.

So etwas ist bei größeren, agilen Projekten nicht unüblich und wird „Rahmenplan“ oder „Timebox der Projektführungsebene“ genannt.

*<Passend zu diesem Rahmenplan (s. auch Folie 6-7 in der Einführungsveranstaltung) sollen Sie nun einen Projektplan erstellen.>*

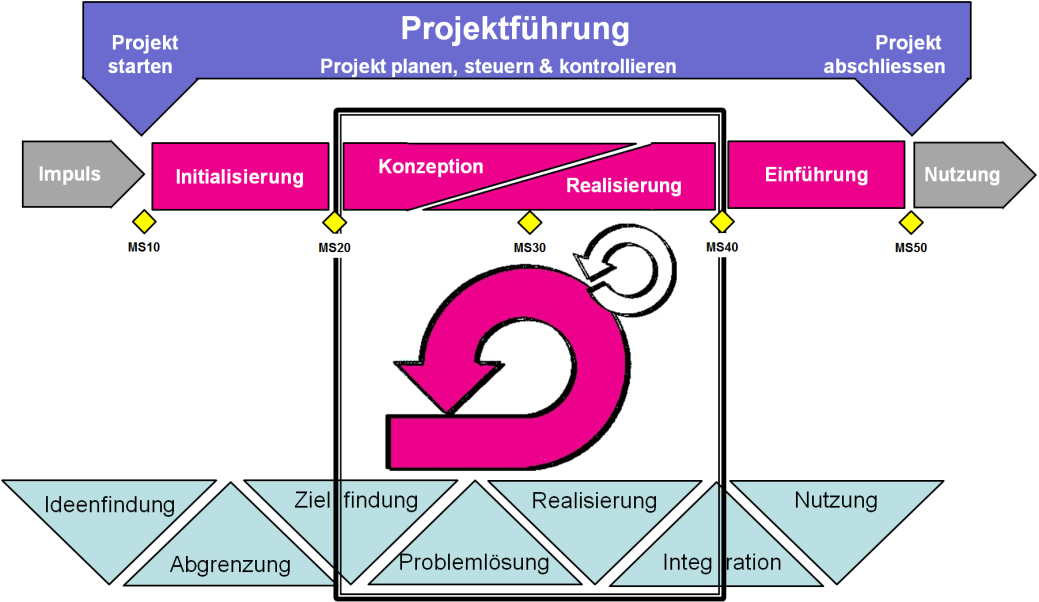


Abbildung 2: SoDa-Prozessmodell nach Hofstätter, Jud basierend auf Jenny03

Durch die Vorgabe des ersten Meilensteins „Architektur“ ist eine Initialisierungsphase vorgegeben, in der wesentliche vorbereitende Aktivitäten stattfinden sollen zu den Themen

* Architekturentwurf Ihrer Software
  + Komponenten und ihre Beziehungen
  + Externe und wichtige interne Schnittstellen
  + Kern des Datenmodells
* Infrastrukturplanung
  + Netzwerkdienste, die eingebunden werden sollen (Drucker, Mail,…)
  + Hardware (z.B. spezielle Controller,…)
  + Betriebssysteme, Middleware,
  + Software-Frameworks oder Protokolle, die genutzt werden soll (z.B. MySQL, LDAP…)
* Qualität (bezogen auf das Gesamtprojekt)
  + Anforderungen an IT-Sicherheit
  + UX
  + Performanz
  + …
* Technologie-Festlegung
  + IDEs
  + SW-Frameworks
  + Austausch und Kommunikation zu Entwicklungsartefakten (u.a. Konfigurationsmanagement, Issue-Board)
  + Testframework
  + …
* Vorbereiten der Projekt-Infrastruktur
  + Kommunikation im Projekt (Wiki, Email-Verteiler,…)
  + Austausch von Dokumenten
  + ..

Nach der Initialisierungsphase wird agil vorgegangen in Form von Scrum-Sprints, die dann jeweils eine Planungs-, eine Konzeptions-, ein Umsetzungs- und eine Auslieferungs/Einführungsphase beinhalten.

*<Ihre Aufgabe ist es, einen ersten Projektplan zu erstellen, der die Arbeitspakete der Initialisierungsphase und des ersten Sprints bis zum14.11.. beinhaltet. Für den weiteren Projektverlauf (2. Und 3. Sprint) genügt eine grobe Planung, anhand der Sie abschätzen können, ob und wie Sie Ziele des Projekts bis zum Projektende erreichen können. Für den Projektplan können Sie mit Netzplantechnik oder mit Balkendiagrammen (GANTT-Diagramm) arbeiten.>*

# Projektunterstützung

## Tools für Entwicklung

Während des ersten Projektmeetings wurden folgende Tools für die Entwicklung beschlossen:

|  |  |
| --- | --- |
| Name | Beschreibung |
| Python 3.X | Grundlage jeglicher Programmierung |
| Keras | Paket für maschinelles Lernen in Python geschrieben |
| NumPy | Paket, um das Rechnen mit Matrizen und Vektoren zu vereinfachen, Geschrieben in NumPy und wird passiv mitgenutzt |
| Tensorflow | Keras stützt sich auf Tensorflow |
| Jupyter Notebook | Sozusagen die IDE |
| Foto Datenbank | <https://web.inf.ufpr.br/vri/databases/parking-lot-database/>  Grundlage zum Anlernen |

## Konfigurationsmanagement

In unserem Projektmeeting wurde festgelegt, dass wir keine Konfigurationen einheitlich einsetzen werden. Wir versuchen eine möglichst breite Masse an Einstellungen auszutesten um das bestmöglichste Ergebnis zu erreichen.

## Tools zur Projektkommunikation und zum Dokumentenaustausch

Während des ersten Projektmeetings wurden folgende Medien beschlossen, die zur Kommunikation und zum Austausch dienen sollen:

|  |  |  |
| --- | --- | --- |
| Name | Link(ggf.) | Beschreibung |
| Github | https://github.com/Scantraxx123/Parkplatzerkennung | Austausch aller Dokumente und Code |
| Slack |  | Kommunikation mit Kunden |
| Whatsapp |  | Kommunikation untereinander |

# Qualitäts- und Testplan

## Definition of Ready und Definition of Done

Definition of Ready (DoR): Was muss ein Backlog Item mindestens erfüllen, damit es im nächsten Sprint Planning ausgewählt werden kann?

* Alle Teammitglieder müssen die Aufgabe verstanden haben
* Die Aufgabe wird als sinnvoll betrachtet
* Die Aufgabe unterstützt das Sprintziel
* Aufwand der Aufgabe wurde verhältnismäßig kalkuliert
* Ziel ist realistisch

Definition of Done (DoD): Was muss ein Backlog Item erfüllen, damit Sie am Ende eines Sprints als vom Product Owner als fertig deklariert werden kann?

* Alle Teammitglieder sehen die Aufgabe als erledigt
* Item wurde getestet
* Ergebnis wird als hinreichend betrachtet
* Das Ergebnis ist in der Versionsverwaltung (Github) hochgeladen
* Ergebnis wurde von anderen Entwicklern überprüft
* Ergebnis ist im Gesamtkonzept mit eingebaut

## Testdesign & Testautomatisierung

Neben funktionalen Tests ergeben sich häufig projektspezifische Aspekte, die speziellere Tests erforderlich machen. (Mobile) Benutzbarkeitstests, Sensor- und Aktortest, Robustheit gegen Sensorfehler oder Kommunikationsverlust,… gehören in diese Kategorie.

Nachdem die CNN angelernt wurde, lässt man wird ein Set mit Testdaten durchlaufen mit denen man die Erkennungsrate des CNN ableiten kann. Dieses Funktionen bring das Keras Framework mit, sodass keine weiteren Frameworks benötigt werden. Als Grundlage für die Testdaten werden die gleichen Daten wie für das Anlernen benutzt. Außerdem überprüfen wir, ob es möglich ist sich eigene Testbilder in großer Zahl zu generieren.

## Testfälle

<noch nicht notwendig> (nur Testfälle für nicht automatisierte Tests, die nicht als «definition of done» formuliert sind)

# Anhänge

<später > (Sprintpläne / Sprintreview-Protokolle / Meilensteinberichte)