

Projektdokumentation - Football App

Gruppe 1 – Lagger,

Sarkovic,

Weiler,

Wutti

SYP-PRE, 4BHIFS 2016/17

05.06.2017

**Inhaltsverzeichnis**

[1. Aufgabenstellung 4](#_Toc484448119)

[2. Story Map 5](#_Toc484448120)

[3. Akzeptanzkriterien User Stories 6](#_Toc484448121)

[3.1 Akzeptanzkriterium User Story - 'Spieler anlegen' 6](#_Toc484448122)

[3.2 Akzeptanzkriterium User Story - 'Positionen verwalten' 7](#_Toc484448123)

[3.3 Akzeptanzkriterium User Story - 'Spieler entfernen' 7](#_Toc484448124)

[3.4 Akzeptanzkriterium User Story - 'Spiel anlegen' 8](#_Toc484448125)

[3.5 Akzeptanzkriterium User Story - 'Login' 8](#_Toc484448126)

[4. Modellierung 9](#_Toc484448127)

[4.1 Klassendiagramm 9](#_Toc484448128)

[4.2 Entity Relationship Diagram 10](#_Toc484448129)

[5. Web Service Dokumentation 11](#_Toc484448130)

[5.1 HTTP Errors 11](#_Toc484448131)

[5.2 Player 11](#_Toc484448132)

[5.2.1 Add Player 11](#_Toc484448133)

[5.2.2 Remove Player 12](#_Toc484448134)

[5.2.3. Get Player 12](#_Toc484448135)

[5.2.3 Get All Players 13](#_Toc484448136)

[5.2.4 Update Player 13](#_Toc484448137)

[5.3 Match 14](#_Toc484448138)

[5.3.1 Add Match 14](#_Toc484448139)

[5.3.2 Get All Matches 14](#_Toc484448140)

[5.3.3 Get Matches 15](#_Toc484448141)

[5.3.4 Update Match 15](#_Toc484448142)

[5.4 Statistic 16](#_Toc484448143)

[5.4.1 Add Statistic 16](#_Toc484448144)

[5.4.2 Get Statistic 16](#_Toc484448145)

[5.4.3 Get All Statistics 17](#_Toc484448146)

[5.5 Position 18](#_Toc484448147)

[5.5.1 Add Position 18](#_Toc484448148)

[5.6 Occupation 18](#_Toc484448149)

[5.6.1 Add Occupation 18](#_Toc484448150)

[5.6.2 Remove Occupation 19](#_Toc484448151)

[5.7 Participation 19](#_Toc484448152)

[5.7.1 Add Participation 19](#_Toc484448153)

[5.7.2 Update Participation 20](#_Toc484448154)

[6 Softwarearchitektur 21](#_Toc484448155)

[7 Aufgabenverteilung 22](#_Toc484448156)

[8 Prozessfortschritt 23](#_Toc484448157)

# Aufgabenstellung

Beginn des Projektes 08.03.2017

Geplantes Ende 14.06.2017

Das Projekt wurde im Rahmen des SYP-PRE und POS Unterrichts durchgeführt. Die drei Wochenstunden SYP-PRE wurden hauptsächlich für die Planung des Projektablaufes genutzt. Das verwendete Vorgehensmodell ist SCRUM und wird mittles dieses Projektes gelehrt.

Die zwei praxisorientierten Wochenstunden POS wurden hauptsächlich für Programmierarbeiten genutzt. Die grundlegenden Androidkentnisse wurden in der Theoriestunde vermittelt und nach und nach durch stätiges Arbeiten am Projekt individuell spezialisiert und verbessert.

Das Projekt ist in fünf zwei-wöchige Sprints aufgeteilt. Die nachfolgende Dokumentation enstand im Laufe dieser zehn Wochen und wird nun zusammengefüft und teilweise ergänzt.

Die Football App soll eine Möglichkeit bieten Spiele statistisch zu erfassen und auszuwerten.

Es gibt mehrer Spiele von welchen jeder über einen Account verfügt. Diese, die über Adminrechte verfügen, können ein neues Spiel anlegen, wobei zwei Teams eingeteilt werden und das Austragungsdatum festgelegt wird. Nach Beendigung des Spiels kann der Admin den Spielstand festlegen. Jeder Spieler kann über mehrere Positionen verfügen, eine davon hat er eine während des Spiels inne.

# Story Map

# Akzeptanzkriterien User Stories

## 3.1 Akzeptanzkriterium User Story - 'Spieler anlegen'

Ein User mit Adminrechten kann einen neuen Spieler (Account) anlegen. Dies erfolgt durch die Eingabe eines gültigen Benutzernamens und Passworts (+ Passwort bestätigen). Das Passwort darf hierbei nicht kürzer als fünf Zeichen sein und der gewünschte Benutzername darf nicht bereits genutzt werden. Zusätzlich kann angegeben werden, ob der neu angelegte Account über Adminrechte verfügen soll.

Der neue Benutzer kann sich direkt nach dem Anlegen mit einem anderen Android Smartphone anmelden und hat Zugriff auf alle ihm gewährten Funktionalitäten.

Allgemeine Vorbedingung: Der gerade angemeldete User besitzt Adminrechte.

Testfälle:

**Vorbedingung** Der Username wird bereits von einem Account verwendet.

**Aktion** Der Name eines bereits existierenden Users wird eingegeben.

**Ergebnis** Es erscheint ein Informationsdialog, das Eingabefeld des Namens wird

geleert.

**Aktion** Ein Name und ein Passwort wird mit nicht zulässigen Zeichen

(Sonderzeichen, Umlaute) eingegeben.

**Ergebnis** Es erscheint ein Informationsdialog, der Text im Eingabefeld wird rot

markiert.

**Aktion** Ein Passwort welches die Mindestlänge (5 Zeichen) unterschreitet wird

eingegeben.

**Ergebnis** Es werden die Mindestanforderungen bekannt gegeben.

**Aktion** Es wird ein Passwort zur Bestätigung angegeben, das nicht dem zuvor

eingegebenen entspricht.

**Ergebnis** Es erscheint ein Informationsdialog, die Eingabefelder betreffend Passwort

werden geleert.

**Aktion** Es wird nichts eingegeben.

**Ergebnis** Es erscheint ein Informationsdialog, der User auffordert etwas einzugeben.

**Aktion** Es wird ein Username sowie ein Passwort eingegeben. Das Passwort stimmt

mit der Bestätigung überein und der Username wird noch nicht verwendet.

Die Checkbox für Adminrechte wird angewählt.

**Ergebnis** Ein neuer User mit dem eingegebenen Namen und Passwort wird angelegt.

Zusätzlich hat dieser Adminrechte.

## 3.2 Akzeptanzkriterium User Story - 'Positionen verwalten'

Ein User kann seine eigenen Positionen verwalten, d.h. festlegen, welche Positionen für ihn spielbar sind. Wenn der User Adminrechte besitzt, kann er die Positionen aller Spieler verwalten.

Bei Spielen an denen der jeweilige User teilnimmt, kann er nur an einer Position spielen die er in seinem Profil als spielbar markiert hat.

Allgemeine Vorbedingung: Der gerade angemeldete User befindet sich in seinem Profil oder

besitzt Adminrechte.

Testfälle:

**Aktion** Keine Position wird als spielbar markiert.

**Ergebnis** Es erscheint ein Informationsdialog, der User wird aufgefordert, zumindest

eine Position zu wählen.

**Aktion** Es wird eine Position als spielbar markiert.

**Ergebnis** In der Datenbank wird beim ausgewählten User die soeben markierte

Position als spielbar festgehalten.

**Aktion** Es werden mehrere Positionen als spielbar markiert.

**Ergebnis** In der Datenbank werden beim ausgewählten User die soeben markierten

Positionen als spielbar festgehalten.

## 3.3 Akzeptanzkriterium User Story - 'Spieler entfernen'

Ein User mit Adminrechten kann aus allen angelegten Spielern einen auswählen und diesen entfernen. Dieser User kann daraufhin nicht mehr in die App einsteigen.

Allgemeine Vorbedingung: Der gerade angemeldete User besitzt Adminrechte.

Testfälle:

**Aktion** Der angemeldete Admin versucht, sich selbst zu entfernen.

**Ergebnis** Es erscheint ein Informationsdialog, der User wird benachrichtigt, dass er

sich nicht selbst entfernen darf.

**Aktion** Es wird ein User ausgewählt der nicht der selbe Admin ist.

**Ergebnis** Es erscheint ein Informationsdialog, welcher den Admin nochmal fragt ob er sich sicher ist, dass er diesen User entfernen will.

## 3.4 Akzeptanzkriterium User Story - 'Spiel anlegen'

Ein User mit Adminrechten kann ein neues Spiel anlegen. Dies erfolgt durch die Eingabe eines gütligen Datums. Das angegebene Datum darf nich in der Zukunft liegen.

Allgemeine Vorbedingung: Der gerade angemeldete User besitzt Adminrechte.

Testfälle:

**Aktion** Das angegeben Datum liegt in der Zukunft.

**Ergebnis** Es erscheint ein Informationsdialog, der User wird aufgefordert, ein

gültiges Datum anzugeben.

**Aktion** Es wird ein gültiges Datum ausgewählt.

**Ergebnis** Ein neues Spiel mit dem angegebenen Datum wird angelegt.

## 3.5 Akzeptanzkriterium User Story - 'Login'

Ein User, der bereits über einen Account verfügt, möchte sich anmelden, um auf die für ihn freigeschalteten Funktionalitäten Zugriff zu bekommen. Der Funktionsumfang hängt von der Berechtigungsstuge ab (Admin, Standard).

Nach dem Schließen der Applikation muss der User sich erneut mit seinen Logindaten anmelden.

Allgemeine Vorbedingung: Der Benutzer verfügt bereits über einen Account. Er verfügt außerdem über

eine Internetanbindung.

Testfälle:

**Vorbedingung** Es gibt keinen Zugriff auf das Internet

**Aktion** Ein Benutzername und ein Passwort werden eingegeben.

**Ergebnis** Es erscheint ein Dialog am unteren Bildschirm der den User

darauf hinweist, dass es ein Problem mit der Datenbankverbindung gibt.

**Aktion** Beide Eingabefelder (Benutzername, Passwort) werden nicht ausgefüllt, es

wird sofort mit Login bestätigt.

**Ergebnis** Der User wird darauf hingewiesen, dass er jeweils mindestens ein Zeichen eingeben muss.

**Aktion** Es wird ein Username sowie ein Passwort eingegeben. Der User bestätigt

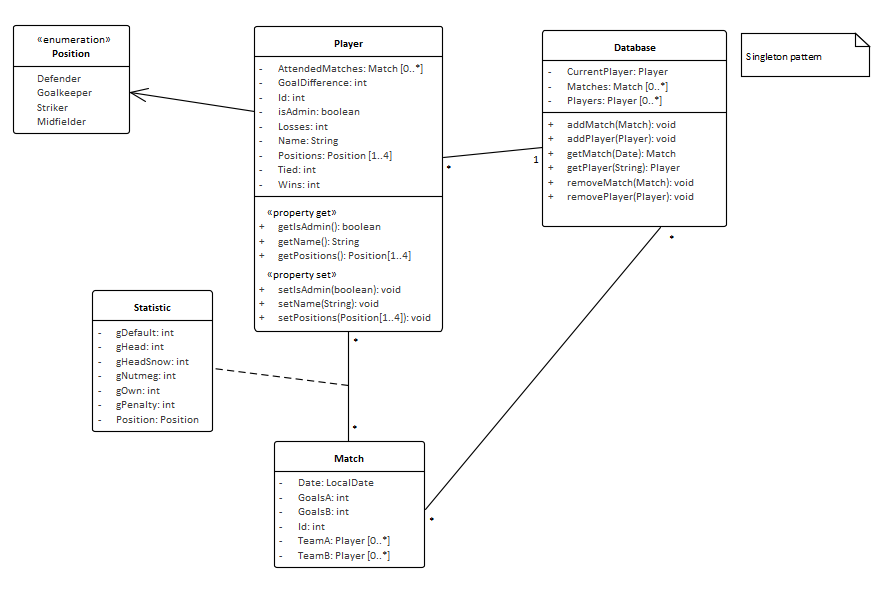
durch klicken auf den Login-Button.

**Ergebnis** Man ist bis zum Schließen der App im System angemeldet und hat Zugriff

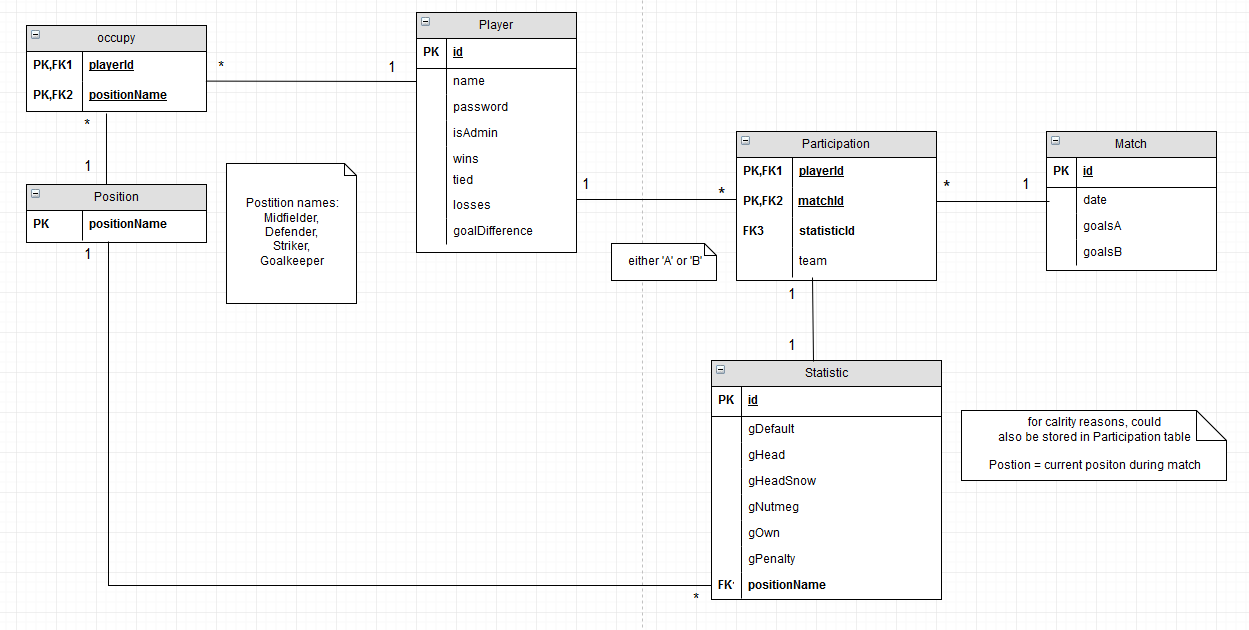
auf alle gewährten Funktionen.

# Modellierung

## Klassendiagramm



## Entity Relationship Diagram



# Web Service Dokumentation

Die Web Service Dokumentation wird aus der von der 5BHIFS Jahrgang 2016/17 zur Verfügung gestellten Dokumentation übernommen.

## HTTP Errors

|  |  |
| --- | --- |
| 400 | Webservice expects other data to be sent. |
| 404 | No webservice implemented for this path. |
| 405 | A method was requested which is not supported by the webserver. |
| 415 | Webservice expects JSON-format, request sends another format. |
| 500 | Should not happen, something on the webserver went wrong. |

## Player

### 5.2.1 Add Player

Request

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| POST | /player |

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | PARAMETER | VALUE | |
| POST | JSON | {                  } | "goalDifference":0,  "id": -1, //set automatically  "isAdmin":0,  "losses":0,  "name":"Zweistein",  "password":"twice",  "tieds":0,  "wins":0 |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | {                  } | "goalDifference":0,  "id":14,  "isAdmin":0,  "losses":0,  "name":"Zweistein",  "password":"twice",  "tieds":0,  "wins":0 |

### 5.2.2 Remove Player

Request

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| DELETE | /player |

|  |  |  |
| --- | --- | --- |
| **TYPE** | **Name** | **VALUE** |
| @QueryParam | id | 14 |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | "ok" |

### 5.2.3. Get Player

Request

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| POST | /player/auth |

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | PARAMETER | VALUE | |
| POST | JSON | {                  } | "goalDifference":0,  "id": -1,  "isAdmin": 0,  "losses":0,  "name":"Andi",  "password":"sichersten",  "tieds":0,  "wins":0 |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | {                  } | "goalDifference":1,  "id":1,  "isAdmin":1,  "losses":2,  "name":"Andi",  "password":"sichersten",  "tieds":2,  "wins":1 |

### Get All Players

Request

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| GET | /player |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | [{ | "goalDifference":1,  "id":1,"isAdmin":1,  "losses":2,  "name":"Andi",  "password":"sichersten",  "tieds":2,  "wins":1 |
|  | },  {                  }] | "goalDifference":1,  "id":2,  "isAdmin":1,  "losses":2,  "name":"Thomas",  "password":"gay",  "tieds":2,  "wins":1 |

### Update Player

Request

|  |  |
| --- | --- |
| METHOD | URL |
| PUT | /player |

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | PARAMETER | VALUE | |
| PUT | JSON | {                  } | "goalDifference":345,  "id":1,  "isAdmin":1,  "losses":23,  "name":"Andi",  "password":"sichersten",  "tieds":34,  "wins":345 |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | "ok" |

## Match

### 5.3.1 Add Match

Request

|  |  |
| --- | --- |
| METHOD | URL |
| POST | /match |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| POST | JSON | {  "firmDate":"2017-04-06",  "goalsA":66,  "goalsB":6,  "id":-1 //automatically set  } |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | {  "firmDate":"2017-04-06", | |
|  | } | "goalsA":66,  "goalsB":6,  "id": 6 |

### 5.3.2 Get All Matches

Request

|  |  |
| --- | --- |
| METHOD | URL |
| GET | /match |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | [{ | "firmDate":"2017-03-20",  "goalsA":1,  "goalsB":0,  "id":5 |
|  | },  {          }] | "firmDate":"2017-04-06",  "goalsA":66,  "goalsB":6,  "id":6 |

### Get Matches

Request

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URL** |  |
| GET | /match/{playerId} |  |
|  |  |  |
| **TYPE** | **PARAMETER** | **VALUE** |
| @PathParam | playerId | 2 |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | [{ | "firmDate":"2010-01-20",  "goalsA":4,  "goalsB":4,  "id":1 |
|  | },  { | "firmDate":"2017-01-13",  "goalsA":3,  "goalsB":4,  "id":2 |
|  | }] | |

### Update Match

Request

|  |  |
| --- | --- |
| METHOD | URL |
| PUT | /match |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| PUT | JSON | {  "firmDate":"2010-01-20",  "goalsA":66,  "goalsB":6,  "id":1  } |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | "ok" |

## 5.4 Statistic

### 5.4.1 Add Statistic

Request

|  |  |
| --- | --- |
| METHOD | URL |
| POST | /statistic |

|  |  |  |  |
| --- | --- | --- | --- |
| TYPE | PARAMETER | VALUE | |
| POST | JSON | {                  } | "id" : -1, //(set automatically)  "gDefault" : 0,  "gHead" : 0,  "gHeadSnow" : 0,  "gNutmeg" : 0,  "gOwn" : 0,  "gPenalty" : 0,  "positionName" : "Sturm" |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | { | |
|  | } | "id" : 5,  "gDefault" : 0,  "gHead" : 0,  "gHeadSnow" : 0,  "gNutmeg" : 0,  "gOwn" : 0,  "gPenalty" : 0,  "positionName" : "Sturm" |

### 5.4.2 Get Statistic

Request

|  |  |
| --- | --- |
| METHOD | URL |
| GET | /statistic/{matchID}/{playerId} |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| @PathParam | matchID | 2 |
| @PathParam | playerId | 2 |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | {                  } | "gDefault":0,  "gHead":0,  "gHeadSnow":2,  "gNutmeg":0,  "gOwn":0,  "gPenalty":0,  "id":2,  "positionName":"attacker right" |

### 5.4.3 Get All Statistics

Request

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URL** |  |
| GET | /statistic/{matchID} |  |
|  |  |  |
| **TYPE** | **PARAMETER** | **VALUE** |
| @PathParam | matchID | 2 |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | [{ | "gDefault":0,  "gHead":0,  "gHeadSnow":2,  "gNutmeg":0,  "gOwn":0,  "gPenalty":0,  "id":2,  "positionName":"attacker right" |
|  | },  {                  }] | "gDefault":3,  "gHead":1,  "gHeadSnow":0,  "gNutmeg":6,  "gOwn":4,  "gPenalty":0,  "id":3,  "positionName":"attacker left" |

## Position

### Add Position

Request

|  |  |  |
| --- | --- | --- |
| **METHOD** | **URL** |  |
| POST | /position |  |
|  |  |  |
| **TYPE** | **PARAMETER** | **VALUE** |
| POST | JSON | {  "position": "attacker right"  } |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | {  "position": "attacker right"  } |

## Occupation

### 5.6.1 Add Occupation

Request

|  |  |
| --- | --- |
| METHOD | URL |
| POST | /occupation |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| POST | JSON | {  "playerId":1,  "positionName":"defender right"  } |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | {  "playerId":1,  "positionName":"defender right"  } |

### 5.6.2 Remove Occupation

Request

|  |  |
| --- | --- |
| **METHOD** | **URL** |
| DELETE | /occupation |

|  |  |  |
| --- | --- | --- |
| **TYPE** | **Name** | **VALUE** |
| @QueryParam | playerId | 1 |
| @QueryParam | positionName | "defender right" |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | "ok" |

## 5.7 Participation

### 5.7.1 Add Participation

Request

|  |  |
| --- | --- |
| METHOD | URL |
| POST | /participation |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| POST | JSON | {  "playerId" : 1,  "matchID" : 2,  "statisticId" : 3,  "team" : "A"  } |

Response

|  |  |  |
| --- | --- | --- |
| Status | Response | |
| 200 | {  "matchID":1, | |
|  | } | "playerId":2,  "statisticId":3,  "team":"A" |

### 5.7.2 Update Participation

Request

|  |  |
| --- | --- |
| METHOD | URL |
| PUT | /participation |

|  |  |  |
| --- | --- | --- |
| TYPE | PARAMETER | VALUE |
| PUT | JSON | {  "playerId" : 1,  "matchID" : 2,  "statisticId" : 3,  "team" : "X"  } |

Response

|  |  |
| --- | --- |
| Status | Response |
| 200 | "ok" |

# Softwarearchitektur

# Aufgabenverteilung

Die Aufgaben wurden in Programmiertasks eingeteilt, welche später vom Bearbeiter wiederum in kleinere Arbeitspakete aufgeteilt worden sind.

|  |  |
| --- | --- |
| Lagger |  |
| Sarkovic |  |
| Weiler |  |
| Wutti |  |

# Prozessfortschritt

Der Prozessfortschritt wurde mit Daily Scrums und Sprintzielen protokolliert. Darüber hinaus wurden Soll-Ist Tabellen nach jedem Sprint erstellt und im Team besprochen.







