­­­­

Projektdokumentation

Frontend der Sportfestverwaltungssoftware des ATIW Berufskollegs

Mirco Böckmann  
**Deinighauser Str.45  
44357 Dortmund  
mirco.boeckmann@education-siemens.com**

**David Wennemaring  
Bleichstraße 38  
33102 Paderborn  
david.wennemaring@education-siemens.com**

Fachinformatiker Anwendungsentwicklung

Inhaltsverzeichnis

[1 Projektdefinition 1](#_Toc499797469)

[1.1 Ausgangssituation 1](#_Toc499797470)

[1.2 Ist-Analyse 1](#_Toc499797471)

[1.3 Soll-Analyse 1](#_Toc499797472)

[1.4 Zieldefinition 1](#_Toc499797473)

[1.5 Abgrenzungskriterien 2](#_Toc499797474)

[2 Projektplanung 2](#_Toc499797475)

[2.1 Zeitplanung 2](#_Toc499797476)

[2.2 Ressourcenplanung 4](#_Toc499797477)

[2.3 Kostenplanung 4](#_Toc499797478)

[2.4 Planung der Umsetzung 4](#_Toc499797479)

[2.5 Qualitätsplan 7](#_Toc499797480)

[2.6 Testplan 8](#_Toc499797481)

[3 Projektrealisierung 8](#_Toc499797482)

[3.1 Analyse 8](#_Toc499797483)

[3.2 Planung 8](#_Toc499797484)

[3.3 Realisierung 8](#_Toc499797485)

[3.4 Testphase 8](#_Toc499797486)

[3.5 Projektdokumentation 8](#_Toc499797487)

[3.6 Projektabschluss 8](#_Toc499797488)

[4 Abweichungen 8](#_Toc499797489)

[4.1 Zeitliche Abweichungen 8](#_Toc499797490)

[4.2 Finanzielle Abweichungen 8](#_Toc499797491)

[5 Reflexion 8](#_Toc499797492)

[5.1 Ausblick 8](#_Toc499797493)

[5.2 Persönliches Fazit 8](#_Toc499797494)

[6 Benutzerhandbuch 8](#_Toc499797495)

[7 Verzeichnisse 9](#_Toc499797496)

[7.1 Abbildungsverzeichnis 9](#_Toc499797497)

[7.2 Tabellenverzeichnis 9](#_Toc499797498)

[8 Anhang 10](#_Toc499797499)

[8.1 Grobe Skizze der Kundenwünsche 10](#_Toc499797500)

[8.2 Pflichtenheft 10](#_Toc499797501)

# 

# Projektdefinition

## Ausgangssituation

Im ATIW Berufskolleg wird jährlich ein Sportfest mit allen Klassen, welche sich zu dem Zeitpunkt in Paderborn befinde veranstaltet. Zur Verwaltung der Daten welche für diese Veranstaltung anfallen wird eine Webapplikation verwendet. Diese beinhaltet beispielsweise welcher Schüler an welcher Disziplin teilnimmt und wie viele Punkt pro Disziplin zu erhalten sind.

## Ist-Analyse

Derzeit existieren zwei Lösungen für die Verwaltung des Sportfestes. Eine alte Version, welche, aufgrund ihrer geringen Usability überarbeitet werden sollte und die Lösung, welche die alte Version ablösen sollte. Die neue Umsetzung wurde bereits zu einem Vorherigen Zeitpunkt angefangen zu entwickeln, allerdings konnte sie nicht voll funktionsfähig bis zum Erreichen der Deadline beendet werden. Ein Hauptteil der Applikation welcher dabei nicht funktioniert ist das eintragen von Ergebnissen, da es Definitionsuneinigkeiten mit der REST-Schnittstelle gab.

## Soll-Analyse

Nach Abschluss des Projektes soll eine neue Form der Seite für Sportarten existieren. Dazu sollen alle neuen Vorgaben bezüglich der Schnittstellen eingebunden werden. Dadurch wird sich ebenfalls die Seite zur Erstellung beziehungsweise Bearbeitung von Disziplinen verändern.

In weiteren Schritten werden Fehler in der Usability behoben um den Workflow zu vereinfachen.

Das Projekt soll letztendlich auf einem Server lauffähig sein und alle benötigten Funktionalitäten für ein Sportfest abdecken.

## Zieldefinition

### Muss-Kriterien

Die Erfüllung der folgenden Kriterien ist Voraussetzung für den erfolgreichen Abschluss des Projektes. Sie stellen die grundlegenden Anforderungen an den Funktionsumfang des Gesamtsystems dar. Ihre Umsetzung hat höchste Priorität.

1. Eintragen der Ergebnisse vereinheitlichen

Zum Zeitpunkt der Ist-Analyse wird noch zwischen Einzel- und Teamsportarten unterschieden. Diese Unterscheidung wird aufgelöst um die Oberfläche und den Workflow zu vereinfachen und den redundanten Code so gering wie möglich zu halten.

1. Anbindung der Restschnittstellen

Ziel ist es alle für das Sportfest relevanten Schnittstellen in die App einzubinden, sofern diese zur Verfügung stehen (Siehe Anhang). Unter Vorbehalt für technisch bedingte Änderungen.

1. Anlegen von Regeln an Schnittstelle anpassen

Die Art und Weise wie Regeln erstellt werden hat sich mit Absprache mit dem Team des Backends geändert. Dadurch bedingt muss auch die entsprechende Oberfläche zum Erstellen und Editieren der Regeln angepasst werden.

### Kann-Kriterien

Über die Muss-Kriterien hinaus gibt es weitere Wünsche des Kunden, welche die Qualität des Systems für den Nutzer verbessern würden. Aufgrund des begrenzten Zeitrahmens kann ihre Umsetzung im Rahmen dieses Projektes nicht garantiert werden. Für den Fall, dass gegen Ende des Durchführungszeitraumes noch Zeit zur Verfügung steht, wird die Anwendungen um folgende Funktionalitäten erweitert. Die Aspekte sind anhand der Priorität, die durch den Kunden festgelegt wurde, absteigend sortiert.

1. Beheben von Grafikfehlern

Während der Ist-Analyse sind Probleme mit der Grafik aufgetreten. Diese behindern nicht die Funktionalität der Anwendung, weshalb deren Beseitigung zu den Kann-Zielen zugeordnet wurde.

1. Update auf neueste Version des Material-Designs

Zwischen dem Start des Projekts im 4. Block des ATIWs und dem 5. Block gab es grundlegende Änderungen beim Material-Design welches hier verwendet wird. Auch hier ist die Funktionalität der Anwendung durch die veraltete Version nicht eingeschränkt.

## Abgrenzungskriterien

Da das Backend und dessen Schnittstellen parallel zu diesem Projekt entwickelt werden, fällt die Funktionalität und Bereitstellung dieser nicht in unser Aufgabenfeld.

# Projektplanung

## Zeitplanung

Für das gesamte Projekt ist eine Bearbeitungsdauer von 70 Stunden eingeplant. Auf dieser Grundlage wurde die Zeitplanung erstellt. Abweichungen der Summen von der Phasenplanung im Projektantrag lassen sich durch die Zuweisung von Ressourcen erklären. Im Antrag wurde der Gesamtaufwand pro Aufgabenpaket eingeschätzt, während im Gantt-Diagramm die voraussichtliche Dauer unter Berücksichtigung parallel zu bearbeitender Vorgänge berechnet wird.

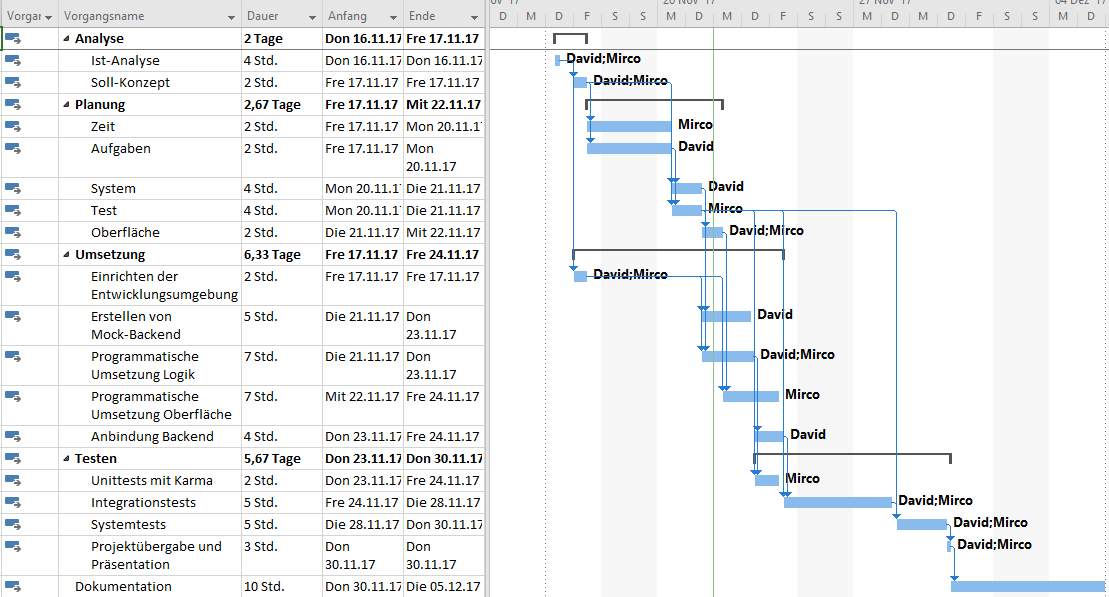


Abbildung : GANTT-Diagramm zur Zeitplanung

## Ressourcenplanung

In diesem Projekt sind drei Ressourcen vorgesehen. Die Ressourcen sind die beiden Entwickler David Wennemaring und Mirco Böckmann und der Projektkunde Jürgen Ewald. Die beiden Entwicklerressourcen sind hierbei an allen Schritten beteiligt, während der Projektkunde nur bei der Umsetzung und bei der Abnahme eingesetzt wird.

Damit belaufen sich die geschätzten Arbeitszeiten wie folgt:

* Mirco Böckmann: 70 Std.
* David Wennemaring: 70 Std.
* Jürgen Ewald: 4 Std.

## Kostenplanung

Damit die Stundensätze für das Projekt übersichtlicher sind, wird jeweils ein Pauschalsatz für die Ressourcen verrechnet. In diesen Sätzen sind jeweils schon Lohnnebenkosten und mögliche anfallende Nebenkosten eingerechnet. Dabei fallen bei den Auszubildenden jeweils ein Satz von ca. 15€ pro Stunden an. Für den Projektkunden wird mit einem Satz von 70€ pro Stunde gerechnet.

Tabelle : Kostenplanung

|  |  |  |  |
| --- | --- | --- | --- |
| Ressource | Stundensatz | Geplante Stunden | Gesamtbetrag |
| Mirco Böckmann | 15€/h | 70 Std. | 1.050 € |
| David Wennemaring | 15€/h | 70 Std. | 1.050 € |
| Jürgen Ewald | 70€/h | 4 Std. | 280 € |
| Summe |  |  | 2.380 € |

## Planung der Umsetzung

### Entwicklungsumfeld

Da das Projekt übernommen wurde, werden die bereits verwendeten Technologien wiederverwendet. Dies bedeutet, dass das Angular2- Framework für das Erstellen der Oberfläche verwendet wird. Für das Schreiben des Quelltextes wird Visual Studio Code verwendet, da dieser sowohl TypeScript, als auch HTML Syntax-Highlighting unterstützt.

Um an einem Mock-System zu arbeiten, während das richtige Back-End parallel von einem anderen Team entwickelt wird, wird ein Tomcat-Server aufgesetzt. Auf diesem läuft dann die Work-in-Progress Version des Frontends.

### Produktfunktionen

Die Funktionen, welche für die überarbeiteten Oberflächen anfallen, sind in dem folgenden Anwendungsfalldiagram (Abbildung 2) visualisiert. Dabei wurden Funktionen, welche Stammdaten und Anmeldungen importieren übersichtlichkeitshalber außer Acht gelassen.

Die in dem Diagramm dargestellten Elementen werden durch mehrere Oberflächen umgesetzt, wobei ein Anwendungsfall nicht für eine Oberfläche steht. Die effektiven Oberflächen, welche mit diesem Diagramm abgedeckt werden sind:

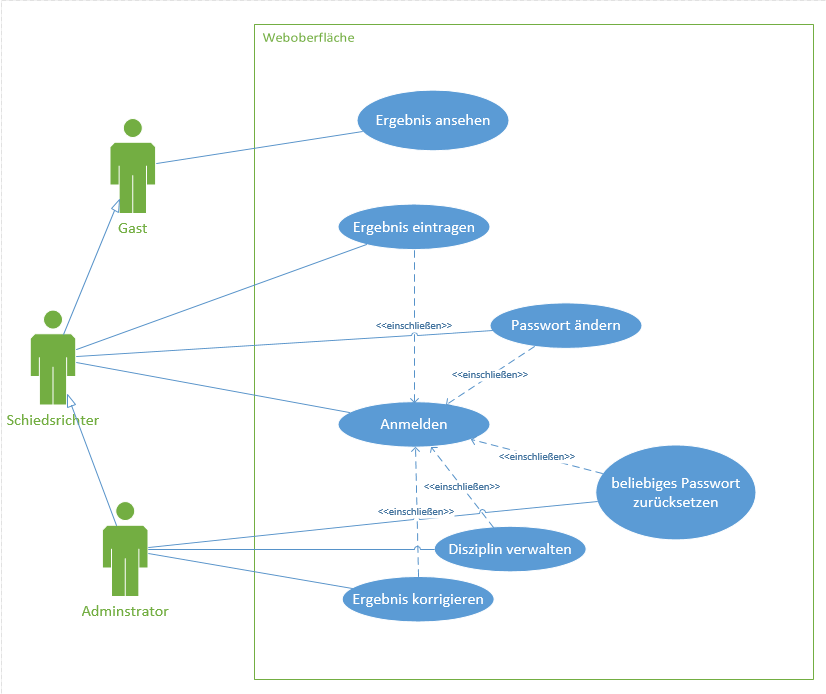


Abbildung : Anwendungsfalldiagramm für die Benutzung der Webseite

Tabelle : Zuordnung von Oberflächen und ihre Funktionalitäten

|  |  |
| --- | --- |
| Seite | Funktionen |
| Überall | * Anmelden * Passwort ändern |
| Disziplin | * Ergebnis ansehen * Ergebnis eintragen * Ergebnis korrigieren |
| Disziplin verwalten | * Disziplin verwalten |
| Benutzer verwalten | * Beliebiges Passwort zurücksetzen |

Der Inhalt der Seiten ist darauf ausgelegt, sich je nachdem ob der User angemeldet ist und wenn ja, welcher Rolle dieser besitzt, zu verändern. Damit ist gemeint, dass ein neuer Bereich auf der „Disziplin“-Oberfläche erscheint, wenn ein Schiedsrichter, oder ein Administrator angemeldet ist. Andere Seiten sind dafür ohne die dafür notwendigen Berechtigungen gar nicht erreichbar, zum Beispiel ist für das Erreichen der „Admin“-Oberfläche eine Anmeldung als Administrator notwendig.

### Programmablauf

Die Hauptfunktionalität ist des Sportfestes ist das Eintragen der Ergebnisse von Disziplinen. Dies geschieht, indem sich der User anmeldet und dann über das Menüband die gewünschte Disziplin auswählt. Daraufhin kann über ein Dropdown-Menü der gerade angetretene Schüler bzw. die gerade angetretene Klasse ausgewählt werden. Neben dem Teilnehmer stehen dann Felder, in welchen der Teilnehmer Leistungen erbringen kann. Diese können per Knopfdruck wieder entfernt werden, falls ein Fehler gemacht wurde. Je nach Disziplin kann dann nach jedem Spiel, oder nach jedem Teilnehmer das Ergebnis gespeichert werden.

### Planung der Benutzeroberflächen

Wie in verhingen Kapiteln erwähnt, liegt der Fokus des Projektes auf der Überarbeitung von bestimmten Seiten. Darunter die wichtigste Oberfläche; das Eintragen von Ergebnisse bei Disziplinen.

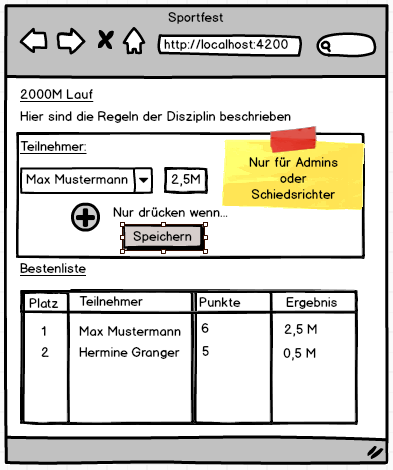


Abbildung : Skizze der Oberfläche für Disziplinen mit Individualleistungen

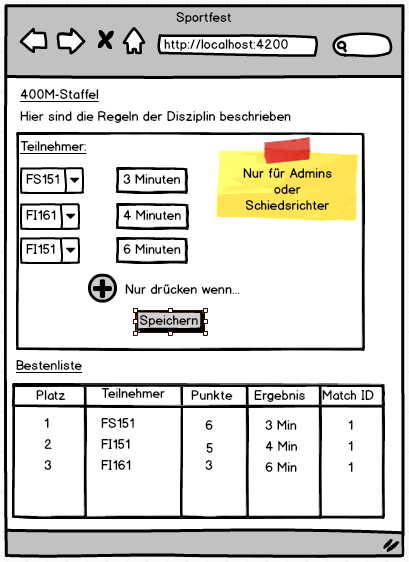


Abbildung : Skizze der Oberfläche für Disziplinen mit Klassenleistungen

Die Oberfläche für die Disziplin soll es in zwei Versionen geben. Eine Versionen für Disziplinen mit Individualleistungen (Abbildung 3) und eine Version für Disziplinen mit Klassenleistungen (Abbildung 4). Wie bereits in erwähnt gibt es einen Bereich auf beiden Seiten, welcher ausschließlich für Benutzer mit genug Rechten einsehbar ist. In diesem sollen die Leistungen des jeweiligen Teilnehmers gespeichert werden. Darunter befindet sich eine Bestenliste, für die aktuell ausgewählte Disziplin. Bei Disziplinen, in denen die Teilnehmer direkt gegeneinander antreten soll ebenfalls eine „Match ID“ angezeigt werden. Diese ist dafür da die Konkurrenten, die gegeneinander angetreten sind, einander zuzuordnen.

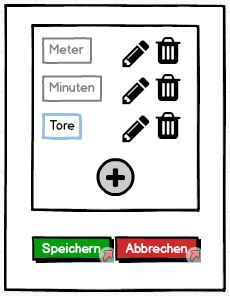


Abbildung : Skizze der Oberfläche für das Verwalten von Einheiten

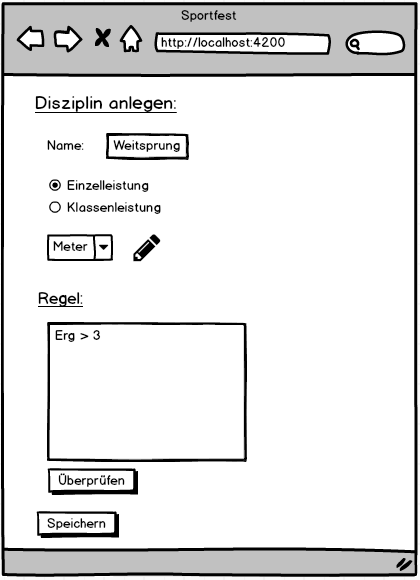


Abbildung : Skizze der Oberfläche für das Anlegen für Disziplinen

Bei der Oberfläche für das Anlegen von Disziplinen (Abbildung 5) soll es möglich sein die Disziplin komplett individuell anzulegen. Deshalb soll vom Namen, über die Art und Einheit der Leistung bis hin zu den Regeln für die Punkteverteilung alles anpassbar sein. Die Einheiten sollen über einen Bearbeiten-Button in einem Pop-up verwaltet werden können (Abbildung 6). In diesem extra Menü sollen die Attribute der Einheit verändert werden könne, wie zum Beispiel, dass das Datenformat eine Zeichenkette ist. Ebenfalls könne in diesem Fenster Einheiten gelöscht oder erstellt werden.

### Planung der Programmstruktur

GGF

## Qualitätsplan

Aufgrund der Tatsache, dass sich die typischen Qualitätsmerkmale wechselseitig beeinflussen, bietet es sich an im Voraus eine Einschätzung ihrer Relevanz für das Endprodukt vorzunehmen. Es werden die Qualitätsmerkmale des Standards ISO 9126 herangezogen.

Tabelle : Bewertung der Qualitätsmerkmale

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | Sehr wichtig | Wichtig | Teilweise wichtig | Unwichtig |
| Funktionalität | X |  |  |  |
| Zuverlässigkeit |  | X |  |  |
| Benutzbarkeit |  | X |  |  |
| Effizienz |  |  |  | X |
| Wartbarkeit |  |  | X |  |
| Portabilität |  | X |  |  |

Aus Übersichtlichkeitsgründen wird an dieser Stelle nicht genauer auf die Bedeutung jedes Merkmals für dieses Projekt eingegangen. Eine genauere Erläuterung befindet sich im beiliegenden Pflichtenheft.

## Testplan

Im generellen sollen die Qualitätsmerkmale mit den drei folgenden Testarten sichergestellt werden. Dazu soll noch unabhängig die Funktionalität der Oberfläche während der Entwicklung überprüft werden. Dieser GUI-Test geschieht dabei parallel zur Entwicklung.

1. Unit-Tests

Diese Tests werden mit dem Karma-Testframework durchgeführt und mindestens für die neuen Features implementiert.

1. Integrationstests (in Zusammenarbeit mit Sportfest BE)

Bei diesem Test wird das Frontend mit dem Backend verbunden, um mögliche Differenzen zu erkennen und zu korrigieren.

1. Systemtests (in Zusammenarbeit mit Sportfest BE)  
   Hier wird ein AccessPoint mit dem Server, auf welchem das Front- und das Backend laufen, aufgebaut. Mit diesem kann sich dann verbunden werden. Damit wird die ordnungsgemäße Funktionalität während des Sportfests getestet und sichergestellt.

# Projektrealisierung

|  |  |  |  |
| --- | --- | --- | --- |
| Analyse | Soll-Stunden | Ist-Stunden | Differenz |
| 14.11.2017 - 15.11.2017 | 6,0 | 6,0 | 0,0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Ist-Analyse | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 4 | 3 | +1,0 | +1,0 | ✓ |
| Das Projekt startet mit der Analyse des letzten Entwicklungsstandes des Sportfest-Frontends. Daraus ergab sich automatisch die Technologie, welche im weiteren Verlauf der Entwicklung zu verwenden ist, wie Angular 2.  *Begründung für die Differenz:*  Da die Technologie und auch schon der ungefähre Stand der Applikation bekannt war, viel die Dauer der Analyse kürzer aus, als zu Anfang erwartet. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Soll-Zustand | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2 | 3 | -1,0 | 0,0 | ✓ |
| Durch das Wissen, auf welchem Stand sich die Applikation zum dem Zeitpunkt befunden hat, wurden der Soll-Zustand definiert. In dem Soll-Zustand sind ebenfalls die Muss- und Kann-Ziele inbegriffen (siehe 1.4 Zieldefinition). Bei der Zielsetzung wurde sich an den Projektzielen des ersten Entwicklungszyklus orientiert und mehrheitlich übernommen.  *Begründung für die Differenz:*  Neben dem Aussehen und den Funktionalitäten der Applikation, musste sich ebenfalls Gedanken um die benötigten Schnittstellen gemacht werden. Dies hat, auf Grund der Vielzahl der Schnittstellen länger gedauert als erwartet. | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Planung | Soll-Stunden | Ist-Stunden | Differenz |
| 16.11.2017 – 20.11.2017 | 14,0 | 15,0 | -1,0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Zeit | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2,0 | 2 | 0,0 | 0,0 | ✓ |
| Als Grundlage wurden alle anfallenden Aufgaben gesammelt und per Brainstorming mit einer Zeit gewichtet. Danach wurde sich im Team auf einen realistischen Wert geeinigt. Danach wurden die Aufgaben zeitlich sinnvoll aufgeteilt. Dabei wurden Arbeitsschritte, welche parallel laufen können parallel gelegt, um möglichst Zeiteffektiv zu arbeiten. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Aufgaben | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2,0 | 1 | -1,0 | -1,0 | ✓ |
| Die Planung der Aufgaben bestand hauptsächlich aus dem Aufteilen der Ressourcen auf die verschiedenen Arbeitsschritte. Dabei lag speziell ein Augenmerk auf die Aufgaben auf parallele ausgeführte werden sollten und mit welcher Kraft sie am besten gleichzeitig bearbeitet werden können.  *Begründung für die Differenz:*  Die Aufgaben konnten aufgrund der minimalen Know-How Differenz leicht verteilt werden, weshalb sich schnell geeinigt werden konnte, wer welche Aufgaben übernehmen wird. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| System | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 4,0 | 6 | +2,0 | +1,0 | ✓ |
| Nachdem die benötigten Schnittstellen auftragsintern definiert waren mussten diese schließlich mit dem Projektteam des Backends synchronisiert werden. Daraus folgte dann ein Datenmodell, mit welchen beide Teams weiterarbeiten konnten. Ebenfalls wurden bei dieser Planung die programmtechnischen Schnittstellen definiert bzw. festgelegt, welche existierenden Schnittstellen übernommen werden können.  *Begründung für die Differenz:*  Im Laufe der Besprechung mit dem Backend-Team hat es Aufgrund von, für uns, unvorhergesehenen Technologie wechseln des Backend-Teams länger gedauert sich auf neues Datenmodell und die damit verbundenen Schnittstellen zu einigen, als zunächst erwartet. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Test | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 4,0 | 4,0 | 0,0 | +1,0 | ✓ |
| Bei den Tests sollten möglichst viele Fehler aufgedeckt werden, egal welcher Herkunft. Deshalb wurden Unit-tests für die neuen Komponenten angesetzt. Für die Anbindung an des Backend wurde ein Integrationstest geplant, um möglich Fehler in Definitionslücken aufzudecken. Als nächstes wurde ein Systemtest geplant, welcher ebenfalls mit dem Backend zusammen stattfinden sollte. Dabei sollte das System in einer Generalprobe im Freien getestet werden. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Oberfläche | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2,0 | 2,0 | 0,0 | +1,0 | ✓ |
| Nach dem, in vorherigen Arbeitsschritten, alles Technische spezifiziert wurde, wurde im letzten Schritt der Planung festgelegt, welche Oberflächen benötigt werden. Dazu wurden bereits vorhandene Oberflächen begutachtet und überprüft, ob diese weiterverwendet werden können, ob diese mit Modifikationen den neuen Anforderungen entsprechen, oder ob sie komplett neu geschrieben werden müssen. Dazu wurden Skizzen für die einzelnen Oberflächen angelegt (siehe 2.4.4 Planung der Benutzeroberflächen) | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Realisierung | Soll-Stunden | Ist-Stunden | Differenz |
| 15.11.2017 – 23.11.17 | 25,0 | 32,0 | +7,0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Einrichtung der Entwicklungsumgebung | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2,0 | 7 | +5,0 | +6,0 | ✓ |
| Die Phase der Realisierung begann mit dem Aufsetzen der Entwicklungsumgebung. Dazu wurden auf den Entwicklungs-PCs NodeJS, ein Tomcat, Git und MariaDB installiert. Da Visual Studio-Code bereits vorinstalliert war, musste dies nicht extra eingerichtet werden. Auf dem Tomcat sollte dabei eine alte Version des Backends laufen. Dazu wurde MariaDB als Datenbank verwendet. Nach der Einrichtung konnte mit zwei Shell-Skripten sowohl der Tomcat, mit allen benötigten Diensten als auch ein Continuous-Build Service für die Angular Oberfläche gestartet werden.  *Begründung für die Differenz:*  Die Verzögerung bestand im Grunde aus zwei Teilen. Der erste Teil war, dass Installationsschwierigkeiten aufgetreten sind, aufgrund fehlender Rechte auf den Benutzer PCs und des erschwerten Downloads der Software, aufgrund der Firmenproxy. Darüber hinaus war der Zugriff auf den Tomcat über das Netzwerk nicht möglich, er musste also für jeden Entwickler einzeln eingerichtet werden. Der zweite Teil war das Einrichten des alten Backends. Aufgrund der lückenhaften Dokumentation für das aufsetzten der Backend Version, hat das Einrichten und Füllen der Datenbank wesentlich länger gedauert, da viele Schritte zur Behebung der Fehler nicht einfach zu identifizieren waren. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Erstellen eines Mock-Backend | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 5,0 | 5 | 0,0 | +6,0 | ✓ |
| Damit das Frontend entsprechend der neuen Definitionen Daten anzeigen kann, werden Objekte und Daten „gemockt“. Um diese Daten im Frontend nutzen zu könne mussten dazu ebenfalls Methoden „gemockt“ werden, welche den Zugriff auf die Datenbank simulieren. Es wurden nur solche Daten „gemockt“, die vom alten Datenmodell abweichen. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programtechnische Umsetzung der Logik | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 7,0 | 8 | +1,0 | +7,0 | ✓ |
| Die Logik konnte parallel zu den Mock-Daten entwickelt werden, da hierfür nicht zwingend Daten benötigt werden. Wichtiger an dieser Stelle war wie die Daten aussehen. Anhand dessen konnten schon Funktionen für die Oberfläche implementiert werden. Sobald eine Mock-Schnittstelle fertig war, konnte sie angebunden werden. Funktionen die zur Logik gehören sind z.B.: Überprüfung der Rechte eines Nutzers, Speichern der eingetragenen Ergebnisse, Validitätsprüfung der Daten  *Begründung für die Differenz:*  Sich ändernde Schnittstellen und neue Überlegungen zur Verbesserung der Übersichtlichkeit mit Blick auf die Wartbarkeit haben die Umsetzung der Logik leicht verzögert. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Programtechnische Umsetzung der Oberflächen | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 7,0 | 9 | +3,0 | +9,0 | ✓ |
| Anhand der vorher definierten grafischen Templates, konnte die tatsächliche Seite nachgebaut werden. Parallel mit der Logik wurde die Oberfläche Stück für Stück erweitert.  *Begründung für die Differenz:*  Der Funktionsumfang welcher erwartet wurde, war nicht so leicht umzusetzen wie zunächst angenommen. Vor allem beim Vereinheitlichen der Oberfläche für die Disziplinen sind Komplikationen aufgetreten. Nicht-nachvollziehbares Verhalten der html-Elemente hat den Implementierungsprozess zusätzlich erschwert.  Darüber hinaus hat sich die Umstellung auf die neuste Version von Angular-Material länger gedauert als zunächst angenommen. Die Versionsunterschiede bezüglich der Verwendung der Elemente. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Anbindung des Backends | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 4,0 | 4 | 0,0 | +9,0 | ✓ |
| Das Backend stand zum Ende der Implementationsphase leider nicht zur Verfügung. Alle gleichbleibenden Schnittstellen sind weiterhin erreichbar. Dies umfasst hauptsächlich die Benutzerverwaltung. Darüber hinaus wurde das Frontend so weit wie möglich vorbereitet um später mit dem Backend zu interagieren. Dazu wurden alle Methoden die das Backend ansprechen bereits implementiert und die im Frontend verwendete Klassenstruktur dem des Backends angeglichen. Eine Tatsächliche Anbindung an das Backend hätte die geplante Zeit überschritten. | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Testphase | Soll-Stunden | Ist-Stunden | Differenz |
| 21.11.17 – 27.12.2017 | 12,0 | 3,0 | -9,0 |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Unit-Test mit Karma | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 2,0 | 3 | +1,0 | +8,0 | ✓ |
| Da Angular ein eigenes Testframework mitliefert, wurde versucht damit die Unit-Tests durchzuführen. Letztendlich ist der Versuch gescheitert, weil alles was in einer Klasse verwendet wird auch in die Klasse der Unit-Tests importiert werden muss. Der Import der das Scheitern verursachte war das http Modul von Angular. Dieses konnte im Zusammenhang mit den Unit-Tests nicht verwendet werden. Um dies zu umgehen, hätte ein eigener Service geschrieben werden müssen der das Backend „mockt“, dieses allerdings nie anspricht. Unsere Mockdaten wurden allerdings in den bestehenden Service integriert, es war also eine Kombination aus echtem und „mock“-Backend. Der Aufwand um tatsächlich alle Daten in einem Service zu „mocken“ wäre in diesem Zeitrahmen nicht möglich gewesen. Aus diesem Grund wurde auf automatische Unit-Tests verzichtet und nur händische verwendet. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Integrationstest | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 5,0 | 0 | -5,0 | +3,0 |  |
| Aufgrund des nicht verwendbaren Backends zum Ende der Implementationsphase konnten keine Integrationstests durchgeführt werden. | | | | | |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Systemtest | | | | | |
| Datum | **Soll-Std.** | **Ist-Std.** | **Diff.** | **Kumulierte Diff.** | **abgeschlossen** |
|  | 5,0 | 0 | -5,0 | -2,0 |  |
| Aufgrund des nicht verwendbaren Backends zum Ende der Implementationsphase konnten keine Integrationstests durchgeführt werden. | | | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Projektabschluss | Soll-Stunden | Ist-Stunden | Differenz |
| 28.11.17 – 28.11.2017 | 3,0 | 3,0 | 0,0 |
| Der Projektabschluss umfasste eine umfangreiche Präsentation der Implementation, sowie der neu hinzugefügten Seiten. Besprochen wurden außerdem die Limitierungen für den Benutzer und eine Erläuterung des Einflusses von Mockdaten auf die Präsentation. Hauptsächlich die für den Nutzer unerwarteten Mockdaten waren ein Disskussionspunkt. Anschließend wurden noch Verbesserungsvorschläge und Aufwand um diese zu realisieren besprochen. | | | |

|  |  |  |  |
| --- | --- | --- | --- |
| Projektdokumentation | Soll-Stunden | Ist-Stunden | Differenz |
| 29.11.2017 – 01.12.2017 | 10,0 | 10,0 | 0,0 |
| Die Projektdokumentation lief wie erwatet. | | | |

# Abweichungen

## Zeitliche Abweichungen

Die Hauptgründe für die zeitlichen Abweichungen sollen an dieser Stelle noch einmal aufgeführt werden.

1. Unterschätzen der Restriktionen im Entwicklungsumfeld

Das Aufsetzen von Projekten im privaten Umfeld macht normalerweise keine Probleme. Der Zugriff auf andere Computer wird im erforderlichen Rahmen nicht verhindert und kein Proxy steht zwischen dem Computer und dem Internet. Das diese Einschränkungen in dieser Art existieren war teilweise nicht klar, da Sie in einem vorherigen Block noch nicht aufgetreten sind. Außerdem wurde unterschätzt, dass sie die Arbeit so stark beeinflussen und verlangsamen.

1. Html und der Kampf mit CSS

Immer wieder sind Probleme mit der Darstellung aufgetreten die auch auf den zweiten Blick nicht unbedingt nachvollziehbar waren. Aufgrund fehlenden Wissens hat an dieser Stelle häufig nur Try-and-Error geholfen.

1. Fehlendes Backend

Da das Backend zum Ende der Implementationsphase nicht zur Verfügung stand, bestand keine Chance Integrations und Systemtests durchzuführen. Die Zeit welche dadurch frei wurde, wurde genutzt um Verzug der vorherigen Schritte auszugleichen und die Kann-Ziele zu realisieren.

## Finanzielle Abweichungen

Die Finanziellen Abweichungen sind nicht vorhanden, da ausschließlich Personalkosten entstanden sind. Mit Blick auf die hundertprozentige Fertigstellung des Projektes müsste allerdings noch etwas investiert werden. Wenn das Backend fertig ist, muss dieses noch angebunden werden wozu noch weitere Stunden anfallen werden. Die Ausgaben für die Vollendung halten sich allerdings in Grenzen, da das Projekt zum größten Teil fertiggestellt ist. Wir rechnen mit zusätzlichem Zeitaufwand von x Stunden

# Reflexion

## Ausblick

## Persönliches Fazit

Aufgrund der gewonnen Erfahrungen wird es möglich sein zukünftige Projekte besser abzuschätzen. Vor allem was die Entwicklung

# Benutzerhandbuch

# Verzeichnisse

## Abbildungsverzeichnis

[Abbildung 1: GANTT-Diagramm zur Zeitplanung 3](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801346)

[Abbildung 2: Anwendungsfalldiagramm für die Benutzung der Webseite 5](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801347)

[Abbildung 3: Skizze der Oberfläche für Disziplinen mit Individualleistungen 6](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801348)

[Abbildung 4: Skizze der Oberfläche für Disziplinen mit Klassenleistungen 6](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801349)

[Abbildung 6: Skizze der Oberfläche für das Verwalten von Einheiten 7](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801350)

[Abbildung 5: Skizze der Oberfläche für das Anlegen für Disziplinen 7](file:///S:\_Public\Projekt\Sportfest%20FE\Projektdokumentation.docx#_Toc499801351)

## Tabellenverzeichnis

[Tabelle 1: Kostenplanung 4](#_Toc499801387)

[Tabelle 2: Zuordnung von Oberflächen und ihre Funktionalitäten 5](#_Toc499801388)

[Tabelle 3: Bewertung der Qualitätsmerkmale 7](#_Toc499801389)

# Anhang

## Grobe Skizze der Kundenwünsche

## Pflichtenheft