

# Freie Universität Berlin

Bachelorarbeit am Institut für Informatik der Freien Universität Berlin

Arbeitsgruppe Software Engineering

## Entwicklung und Einführung eines webbasierten Abrechnungssystems für Übungsleiter eines Sportvereins

Paul Harfenmeister

Matrikelnummer: 4454000

[paul.harfenmeister@fu-berlin.de](mailto:paul.harfenmeister@fu-berlin.de)

Eingereicht bei: Prof. Dr. Lutz Prechelt

Berlin, 1. Februar 2019



## **Zusammenfassung**

In dieser Arbeit wird ein auf Papier und Unterschriften basierter Abrechnungsprozess eines Sportvereins digitalisiert. Dazu wird ein webbasiertes Mehrbenutzersystem entworfen und implementiert, mit dem die Rollen Übungsleiter, Abteilungsleitung und Mitarbeiter ihre bisherigen Aufgaben online erledigen können. Als Webframework wird das in Python entwickelte Django genutzt. In der Entwicklung werden moderne Softwareentwicklungstechniken wie eine Versionskontrolle, automatische Tests, Continuous Integration sowie Continuous Deployment angewendet. Ebenfalls Teil dieser Arbeit ist die produktive Einführung des Systems in dem Sportverein, welche sehr erfolgreich verlaufen ist. Durch das Abrechnungssystem werden alle Beteiligten zeitlich entlastet. Alle Personen konnten durch Nutzung der Webseite ihre Aufgabe schneller erledigen. Die Software soll über die Arbeit hinaus weiter entwickelt werden und im hinteren Teil ist ein Ausblick auf die zukünftigen Weiterentwicklungen und Ausbaustufen zu finden.



## **Eidesstattliche Erklärung**

Ich versichere hiermit an Eides Statt, dass diese Arbeit von niemand anderem als meiner Person verfasst worden ist. Alle verwendeten Hilfsmittel wie Berichte, Bücher, Internetseiten oder ähnliches sind im Literaturverzeichnis angegeben, Zitate aus fremden Arbeiten sind als solche kenntlich gemacht. Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungskommission vorgelegt und auch nicht veröffentlicht.

1. Februar 2019

Paul Harfenmeister



# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung</b>	<b>9</b>
<b>2</b>	<b>Die Domäne Sportverein</b>	<b>10</b>
2.1	Der Sportverein . . . . .	10
2.2	Der bisherige Abrechnungsprozess . . . . .	11
2.2.1	Die Abrechnung . . . . .	12
2.2.2	Nachteile des Abrechnungsprozesses . . . . .	12
<b>3</b>	<b>Anforderungen und Rahmenbedingungen für den neuen Abrechnungsprozess</b>	<b>15</b>
<b>4</b>	<b>Entwicklungsumgebung</b>	<b>16</b>
4.1	Wahl des Frameworks . . . . .	16
4.2	Entwicklungsumgebung und Setup . . . . .	17
4.3	Versionsverwaltung . . . . .	18
4.4	Durchführung von Tests und Coverage . . . . .	18
4.5	Continuous Integration . . . . .	19
4.6	Continuous Deployment . . . . .	19
4.7	Verwendete Datenbank . . . . .	20
<b>5</b>	<b>Das neue Abrechnungssystem</b>	<b>22</b>
5.1	Die Entwicklung . . . . .	22
5.1.1	Modellierung der Daten . . . . .	22
5.1.2	Die Benutzer- und Zugriffsverwaltung . . . . .	25
5.1.3	Aufbau der Seite und Funktionen . . . . .	26
5.2	Die Einführung des Systems im Verein . . . . .	27
<b>6</b>	<b>Ausblick</b>	<b>30</b>
<b>7</b>	<b>Zusammenfassung</b>	<b>32</b>
	<b>Literaturverzeichnis</b>	<b>33</b>
<b>A</b>	<b>Anhang</b>	<b>34</b>
A.1	PDF-Abrechnungsformular . . . . .	34





## 1 Einleitung

Heutzutage gibt es für viele Anliegen und Problemstellungen webbasierte Lösungen im Internet. Man schließt Handyverträge online ab, nutzt Online-Banking und kommuniziert mit seiner Krankenkasse über ein Online-Portal. In der Regel führt das zu einer schnelleren, kostengünstigeren und komfortableren Verwaltung für den Nutzer.

Im Gegensatz dazu erfolgt der Abrechnungsprozess für Übungsleiter in meinen Sportverein immer noch größtenteils auf Basis von wanderndem Papier. Dieser Prozess wird jeden Monat von mehr als 100 Personen genutzt, um die geleisteten Trainingsstunden abzurechnen. Pro Abrechnung sind jedes Mal mindestens drei Personen daran beteiligt. Vereinzelt nutzen einige Personen bereits die Möglichkeiten der Digitalisierung, aber es existiert kein einheitlicher Standard. Dementsprechend treten große Unterschiede in der Qualität der Abrechnungen, also auch im zeitlichen Arbeitsaufwand für die beteiligten Personen auf.

Entsprechend unterschiedlich sind die Meinungen zum Abrechnungsprozess bei den Beteiligten, die im Rahmen einer Umfrage erhoben wurden. Von 69 Befragten gaben nur fünf an, dass sie sehr zufrieden sind, 44% sind immerhin zufrieden, 36% sind weder zufrieden noch unzufrieden. 11% sind unzufrieden oder sehr unzufrieden.

Ziel dieser Arbeit ist, den Abrechnungsprozess vollständig zu digitalisieren und anschließend produktiv einzuführen. Dabei sollen möglichst viele Techniken der modernen Softwareentwicklung verwendet werden.

Es soll eine Webseite mit Benutzerverwaltung entworfen werden, die alle Anforderungen des alten Abrechnungsprozesses abbildet und vereinheitlicht. Dabei sollen die Aspekte von der Sichtung der Anforderungen über die Modellierung der Daten, automatisches Testen und Deployment sowie die eigentliche Einführung im Verein betrachtet werden.

Die erste Version der Software konzentriert sich auf die minimalen grundlegenden Anforderungen. Damit ist ein schneller Start und Einsatz des Systems möglich. Es soll im Rahmen dieser Arbeit und darüber hinaus weiter entwickelt und ausgebaut werden. Die zukünftigen Erweiterungsmöglichkeiten werden teilweise bei grundlegenden Designentscheidungen mit berücksichtigt.

## 2 Die Domäne Sportverein

Für das Verständnis des Problems und der Umsetzung wird zunächst in die Struktur und Prozesse des Sportvereins eingeführt.

### 2.1 Der Sportverein

Der Turn- und Sportverein Lichterfelde von 1887 e.V. ist ein Mehrspartenverein mit Sitz in Berlin-Lichterfelde. Der Verein hat 2.500 Mitglieder in zwölf Abteilungen. Einer Abteilung ist mindestens eine Sportart wie zum Beispiel Turnen, Schwimmen oder Leichtathletik zugeordnet und gibt der Abteilung ihren Namen.

Pro Abteilung gibt es mehrere Trainingsangebote und Gruppen, die von einem oder mehreren Übungsleitern geleitet und betreut werden. Ein Übungsleiter ist entweder ein Trainer mit einer Lizenz oder ein Helfer ohne Lizenz. Im weiteren Verlauf werden die beiden Varianten unter Übungsleiter zusammengefasst und nur, wenn es einer Differenzierung bedarf, von Trainer oder Helfer gesprochen. Der Großteil der über 150 Übungsleiter engagiert sich nebenberuflich im Rahmen von einer bis zehn Stunden pro Woche im Verein. Darüber hinaus hat der Verein einige selbstständige Übungsleiter, die zusätzlich in mehreren Vereinen Training anbieten.

Der Übungsleiter erhält für die geleisteten Trainingsstunden ein Entgelt, welches nach Ablauf eines Monats durch eine Übungsleiterabrechnung abgerechnet wird. Pro Jahr werden durch die Übungsleiter circa 1.500 Abrechnungen mit einem Gesamtbetrag von 125.000 Euro eingereicht.

Der Verein wird durch den gewählten, ehrenamtlichen Vorstand nach innen und außen vertreten und geleitet. Er ist für die Geschäftsführung und Einhaltung von Gesetzen und Ordnungen verantwortlich. Zum Zeitpunkt dieser Arbeit bilden drei Mitglieder des Vereins den Vorstand, wobei der Autor Vorstandsvorsitzender ist.

Dem Vorstand und Verein steht eine Geschäftsstelle für die Verwaltung zur Verfügung. Die Geschäftsstelle wird durch eine hauptamtliche Vollzeitkraft geleitet und durch weitere Mitarbeiter unterstützt. Dabei handelt es sich um Absolventen eines Freiwilligen Sozialen Jahres (FSJ), eine duale Studentin und einen Werksstudenten. Alle zusammen werden im weiteren Verlauf der Arbeit als Mitarbeiter bezeichnet.

In jeder Abteilung wird von den Mitgliedern für maximal zwei Jahre eine Abteilungsleitung gewählt. Sie besteht aus mindestens drei Personen mit den Positionen Abteilungsleiter, stellvertretender Abteilungsleiter und Kassenwart. Die Abteilungsleitung kümmert sich um die sportlichen und organisatorischen Belange ihrer Abteilung. Sie verwaltet ein eigenes Budget, das in einen Teil für sportliche, organisatorische Belange und einen Teil für die Übungsleiterentgelte aufgeteilt ist.

Jeder Übungsleiter hat einen Vertrag. Dieser wird zwischen dem Verein (vertreten durch den Vorstand), einer Abteilung (vertreten durch ein Mitglied der Abteilungsleitung) und dem Übungsleiter geschlossen. Die Abteilungsleitung legt das Entgelt (Stundensatz) und den Umfang für den Vertrag fest.

Somit sind die wichtigsten Rollen Vorstand, Mitarbeiter, Abteilungsleitung und Übungsleiter für den weiteren Verlauf der Arbeit definiert.

## 2.2 Der bisherige Abrechnungsprozess

Der zu ersetzende Abrechnungsprozesses hat viele verschiedene Schritte, die in der nachfolgenden Grafik aufgeführt sind.

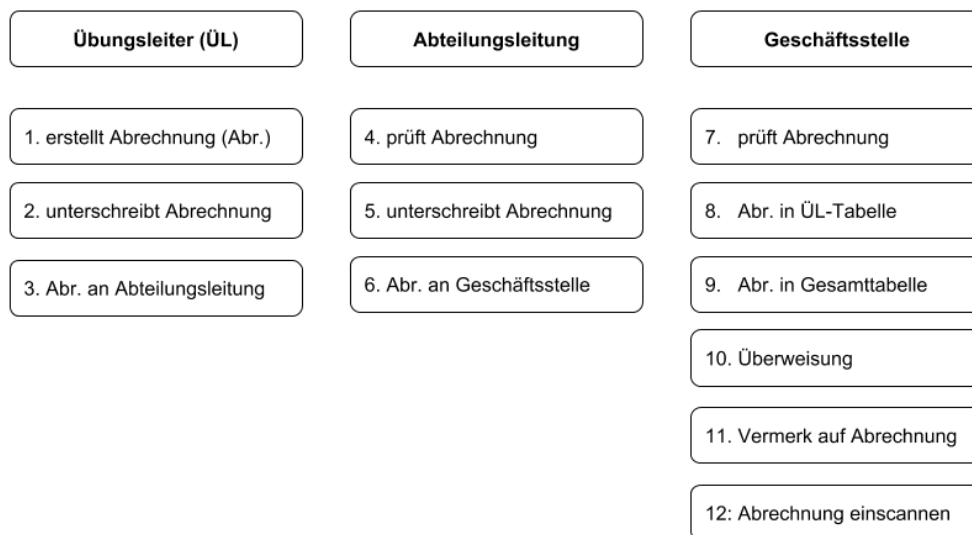


Abbildung 1: derzeitiger Abrechnungsprozess

Nach Ablauf eines Monats erstellt der Übungsleiter seine Abrechnung und unterschreibt diese. Anschließend wird die Abrechnung an die Abteilungsleitung zur Prüfung und Unterschrift gegeben. Dann geht die Abrechnung weiter zur Geschäftsstelle, wo sie durch einen Mitarbeiter ebenfalls noch einmal auf alle Angaben überprüft wird. Der Mitarbeiter vermerkt dann die geleisteten Stunden und das Entgelt in einer Tabelle für jeden Übungsleiter. Zusätzlich wird das Entgelt in einer Gesamttabelle der Abteilung vermerkt. Anschließend erstellt der Mitarbeiter eine Überweisung im Bankprogramm, unterschreibt die Abrechnung und vermerkt das Überweisungsdatum. Zum Schluss wird die Abrechnung zur digitalen Archivierung eingescannt.

## 2. Die Domäne Sportverein

### 2.2.1 Die Abrechnung

Für die Abrechnung stand den Übungsleitern bisher ein ausfüllbares PDF-Formular zur Verfügung (siehe Anhang: "PDF-Abrechnungsformular"). Dort sind durch den Übungsleiter im Kopfbereich folgende Angaben einzugeben:

- Jahr der Abrechnung
- Monat der Abrechnung
- Vorname und Nachname
- Abteilung
- Gruppen und Sportstätten
- Lizenz-Nummer (wenn vorhanden)
- Gültigkeit der Lizenz (wenn vorhanden)

Im Mittelteil der Abrechnung werden die geleisteten Trainingsstunden angegeben. Dazu gehören Wochentag, Tag im Monat, Trainingsbeginn und -ende, die Dauer und Gruppe/ Sportstätte. Zusätzlich muss angegeben werden, ob es sich um eine Wettkampfbetreuung oder ein normales Training handelt. Früher gab es die einheitliche Regelung, dass eine bis vier Stunden Wettkampfbetreuung als eine Zeitstunde abgerechnet werden dürfen, bei mehr als vier Betreuungsstunden als zwei Zeitstunden.

### 2.2.2 Nachteile des Abrechnungsprozesses

Wie schon deutlich wurde, ist der Abrechnungsprozess sehr aufwendig und komplex. Diese Feststellung wird von den Antworten einer Umfrage zu den Meinungen zum derzeitigen Abrechnungsprozesses gestützt. An der Umfrage haben 61 Übungsleiter, 16 Abteilungsleitungen und ein Mitarbeiter teilgenommen. Ein Übungsleiter braucht im Schnitt etwas mehr als 12 Minuten für die Erstellung der Abrechnung. Jeder fünfte Übungsleiter benötigt sogar 20 Minuten oder mehr. Ebenso variieren die Anzahl der Tage von der Erstellung der Abrechnung bis zum Geldeingang auf dem Konto des Übungsleiters. Der Durchschnitt liegt bei etwas mehr als zehn Tagen, wobei einige auch bis zu 30 Tage auf ihr Geld warten.

#### **Übungsleiter**

Die Umfrage hat ergeben, dass 75% der Übungsleiter das ausfüllbare PDF-Formular für die Erstellung der Abrechnung verwenden und das Formular gut finden. Die restlichen Übungsleiter füllen das Formular per Stift aus. Allerdings treten einige Probleme schon beim Öffnen des PDF-Formular auf. So wird beispielweise das PDF nach dem Herunterladen auf einem Apple-Gerät meist erst in der Vorschau geöffnet. In der Folge funktionieren die Formeln in dem Formular nicht mehr und es kann

nicht verwendet werden. Ein weiterer negativer Punkt ist die zwingende Verwendung vom Adobe Acrobat Reader, damit die Formeln im Formular funktionieren. Einige Übungsleiter wollen den AcrobatReader nicht einsetzen oder haben auf Linux-Systemen nur eine alte nicht unterstützte Version. Des Weiteren wird für das Erstellen der Abrechnung ein PC oder Laptop benötigt, da das Formular mit dem Handy nicht funktioniert.

Wenn das PDF-Formular fertig ausgefüllt ist, muss es unterschrieben werden. Einige Übungsleiter fügen die Unterschrift digital ein und versenden somit gleich die PDF-Datei an die Abteilungsleitung. Der Großteil der Nutzer druckt die ausgefüllte Abrechnung aus und unterschreibt. Ein weiteres Hindernis ist auch, dass einige Übungsleiter gar keinen Drucker zu Hause haben und die Abrechnungen dann woanders ausdrucken müssen.

Am aufwendigsten ist es, die Abrechnung der Abteilungsleitung zukommen zu lassen und deswegen gibt es hier auch die meisten Varianten. Einige Übungsleiter senden die Abrechnung per Post, andere übergeben sie in der Halle. Gerne wird die Abrechnung auch wieder eingescannt und per Mail versendet. Immer häufiger werden Abrechnungen einfach mit dem Handy abfotografiert und per Messenger als Bild versendet. Entsprechend schlecht ist dann die Qualität, vor allem weil die Abrechnung in der Regel noch einmal ausgedruckt werden muss.

### **Abteilungsleitung**

Von den Abteilungsleitungen werden vor allem die langen Dienstwege sowie die schlechte Qualität einiger Abrechnungen als negative Punkte genannt. Weitere Punkte sind, dass die Abrechnungen auf unterschiedliche Weise zu ihnen kommen (per Post, persönlich, per Mail, ...) und ein gutes System dafür benötigt wird. Bei der Übergabe in der Sportstätte gibt es auch zwei Varianten. Einige Übungsleiter unterschreiben die Abrechnung direkt und der Übungsleiter muss sie dann zur Geschäftsstelle bringen oder die Abteilungsleitung nimmt sie mit und lässt sie der Geschäftsstelle zukommen.

Insgesamt ist es eine Menge Papier, das durch die Hände und Sporttaschen wandert und so kann auch mal etwas verloren gehen. Das ist in der Vergangenheit leider schon mehrmals vorgekommen.

### **Mitarbeiter/Geschäftsstelle**

Die Mitarbeiter der Geschäftsstelle sind die letzte Station im Abrechnungsprozess. Die häufigsten Fehler auf Abrechnungen sind falsch angegebene oder ausgerechnete Stunden, vor allem bei händischen Abrechnungen. Oft fehlen auch Angaben wie Name, Monat oder Lizenznummer. Immer mal wieder kommt es auch vor, dass eine Unterschrift fehlt, weil der Übungsleiter die Abrechnung direkt an die Geschäftsstelle geschickt hat.

## *2. Die Domäne Sportverein*

Insgesamt sind für den Mitarbeiter sehr viele Schritte zu erledigen und verschiedene Tabellen und Dateien zu bearbeiten. Das dauert natürlich lange und ist fehleranfällig.

### **3 Anforderungen und Rahmenbedingungen für den neuen Abrechnungsprozess**

Bevor mit der Umsetzung des digitalen Abrechnungsprozesses begonnen wird, müssen einige Anforderungen ermittelt und die Rahmenbedingungen geklärt werden. Die erste Frage war, ob der Verein seinen Abrechnungsprozess überhaupt ändern möchte. Diese Entscheidung wurde durch die Funktion des Autors im Vorstand des Vereins sehr schnell getroffen und die Entwicklung eines eigenen Abrechnungssystems unterstützt. Auch durch die Mitarbeiter der Geschäftsstelle wurde dieser Vorstoß begrüßt und unterstützt.

Eine weitere Frage war, ob im Abrechnungsprozess auf die Unterschriften verzichtet werden kann. Eine Nachfrage beim betreuenden Steuerbüro hat bestätigt, dass dies möglich ist. Zur Absicherung soll der Übungsleiter am Ende eines Jahres die erhaltende Gesamtsumme per Unterschrift bestätigen.

Eine wichtige Anforderung und Motivation für das Abrechnungssystem ist, dass der Abrechnungsprozess für alle Beteiligten schneller, einfacher und komfortabler wird. Dadurch sollen aber nicht die Kontrollinstanzen von Abteilungsleitung und Mitarbeiter wegfallen.

Ebenfalls einen sehr hohen Stellenwert hat die Sicherheit der Daten und die Beachtung der Datensparsamkeit. So dürfen Übungsleiter nur ihre eigenen Abrechnungen einsehen und bearbeiten und Abteilungsleitungen nur die ihrer Abteilung.

Wichtig ist weiterhin, dass das Erstellen, Einreichen und Prüfen von Abrechnungen auf einem Smartphone funktioniert. Aus Erfahrung nutzen vor allem ganze junge und eher ältere Mitglieder lieber ein Smartphone als einen Laptop oder PC. Zahlen von der Vereinswebseite zeigen, dass mittlerweile über 50% der Nutzer die Seite mit einem Smartphone besuchen. Tendenziell wird diese Zahl in den nächsten Jahren weiter steigen.

Eine weitere Anforderung ist, dass das System sehr stabil und möglichst wartungsarm funktionieren soll. Der Prozess mit dem PDF-Abrechnungsformular ist fast zehn Jahre alt und davor wurde jahrelang mit Durchschlagbögen gearbeitet. Das System sollte also mindestens auch zehn Jahre oder mehr eingesetzt werden können. Zu berücksichtigen ist hierbei, dass es durch die große Anzahl von Ehrenamtlichen und durch regelmäßige Wahlen spätestens nach zwei Jahren zu Personalwechseln in der Vereinsverwaltung und -führung kommen kann. Dementsprechend müssen Zugriffsberechtigungen und Serverzugänge angepasst oder weitergegeben werden.

## 4 Entwicklungsumgebung

Um eine möglichst gute Wartbarkeit, Stabilität und Weiterentwicklungsmöglichkeit zu erreichen, wurde im Rahmen dieser Arbeit auf moderne Softwareentwicklungstechniken wie Versionsverwaltung, automatische Tests, Testabdeckung, Continuous Integration und Continuous Deployment gesetzt. In den nächsten Kapiteln wird der Einsatz und Konfiguration ausführlicher beschrieben und auf bestimmte Entscheidungen näher eingegangen.

### 4.1 Wahl des Frameworks

Der zeitliche Rahmen einer Bachelorarbeit ist für die Entwicklung einer kompletten Verwaltungswebseite sehr knapp bemessen. Die zeitliche Komponente muss also bei der Wahl der eingesetzten Webtechnologie berücksichtigt werden. Da in der Vergangenheit gute Erfahrungen mit dem Webframework Angular gemacht wurden, kam die Nutzung dieses Frameworks auch für die Erstellung des Abrechnungssystems in Frage. Angular ist ein Framework für die Erstellung von Single-Page-Anwendungen. Diese zeichnen sich dadurch aus, dass die Webseite dynamisch beim Client im Browser erzeugt wird und mit einem Backend hauptsächlich Daten über das Datenformat JSON ausgetauscht werden. Bei diesem Ansatz müsste aber sowohl das Backend, als auch das Frontend separat entwickelt werden. Auf der Seite des Backends fehlt dem Autor aber die ausreichende Programmiererfahrung, was zu einer erhöhten Einarbeitungszeit führen würde. Ein weiterer kritischer Punkt ist, dass Angular ein sich sehr schnell entwickelndes Framework ist, von dem pro Jahr zwei neue Releases veröffentlicht werden. Die Releases werden nur für 18 Monate mit Sicherheitsupdates versorgt und in der Vergangenheit gab es häufiger Inkompatibilitäten zwischen den Versionen. Mit Blick auf eine möglichst lange Nutzungsdauer und geringen Wartungsaufwand sind das keine optimalen Voraussetzungen.

Die Wahl des Frameworks ist letztendlich auf das in Python geschriebene Django gefallen. Mit Django wurden ebenfalls bereits gute Erfahrungen gemacht und schon mehrere Projekte verwirklicht. Django ist ein sehr stabiles und ausgereiftes Framework, das immer noch aktiv weiter entwickelt und gepflegt wird. Es bietet ein objektorientiertes Datenmodell als Abstraktionsschicht zur Datenbank. Damit lassen sich datenintensive Anwendungen sehr schnell modellieren, ohne dass man sich mit SQL oder den Datenbankspezifika beschäftigen muss. Weiterhin beinhaltet Django bereits eine komplette Benutzerverwaltung sowie ein einfach zu konfigurierendes Admininterface. Ebenfalls mit dabei sind einfache Werkzeuge zum Testen sowie ein Entwicklungsserver. Fehlende Funktionalität lässt sich durch unzählige Pakete erweitern.

Der Releasezyklus von Django beträgt circa acht Monate, wobei jede Version für mindestens 15 Monate gepflegt wird. Alle zwei Jahre erscheint eine Version mit Langzeit-



unterstützung (LTS), die für drei Jahre mit Sicherheitsupdates versorgt wird. Somit bleibt zwischen zwei LTS-Versionen ein Jahr Zeit für ein Update, was den Aufwand und die Dringlichkeit für ein Update verringert.

Das Abrechnungssystem wurde initial mit der Version 2.0 von Django erstellt. Diese wird bis zum April 2019 unterstützt. Obwohl schon die Version 2.1 zur Verfügung stand, wurde dennoch die Vorgängerversion gewählt. Das Ziel dieses Ansatzes ist es, die Aktualisierung auf die nächste Version so zu dokumentieren, dass sie später auch durch eine andere Person als den Entwickler durchgeführt werden kann. Im zeitlichen Rahmen dieser Arbeit wird diese Möglichkeit nur berücksichtigt, nicht aber durchgeführt. Die Aktualisierung und dessen Dokumentation soll nach Abgabe der Arbeit erfolgen.

### 4.2 Entwicklungsumgebung und Setup

Entwickelt wurde die Anwendung auf der täglich genutzten Linuxdistribution Arch Linux, die sich durch sehr aktuelle Paketquellen auszeichnet. Als Datenbank wurde zur Entwicklung das dateibasierte SQLite3 eingesetzt. Somit ist die Konfiguration eines Datenbankservers entfallen und hat Zeit gespart. Die Wahl von Arch Linux hatte allerdings auch einen Nachteil. Durch ein Update von SQLite3 auf Version 3.26 funktionierten einige Datenbankmigrationen mit Django nicht mehr. Dabei handelte es sich um einen Bug von Django, der erst im Januar mit einer neuen Version von Django behoben wurde. Bis dahin musste die lokale SQLite3 Version auf 3.25 herunter gesetzt werden.

Anfangs wurde auf die Python Version 3.7 gesetzt, die auch auf dem späteren Server unterstützt wird. Allerdings stellte sich schnell heraus, dass diese Version zwar auf dem Server verfügbar ist, sich aber keine Pakete mit dem Python Paketmanager (pip) pip3.7 installieren ließen. Es gab ein Problem mit OpenSSL, wodurch letztendlich doch auf Python 3.6 gewechselt wurde.

Um die Pakete von Python in einer bestimmten Version zu nutzen und keine Kollisionen mit systemweit installierten Abhängigkeiten zu bekommen, wird eine Python Umgebung (virtuelenv) erzeugt und genutzt. Die Erstellung der Umgebung inklusive der Installation der Abhängigkeiten ist in einem Bash-Skript zusammengefasst. Damit ist ein schnelles und einfaches Starten für zukünftige Entwickler oder beim Aufsetzen einer neuen Umgebung möglich.

Die verwendeten Python Pakete werden in einer Textdatei gespeichert. Da für verschiedene Schritte der Entwicklung unterschiedliche Pakete benötigt werden, wurde sich für eine Aufteilung auf mehrere Dateien entscheiden. Somit werden immer nur die Pakete heruntergeladen und installiert, die auch im entsprechenden Schritt benötigt werden. Pakete, die in allen Schritten verwendet werden, sind in der Datei

## 4. Entwicklungsumgebung

`base.txt` aufgeführt. Pakete, die für das Testen gebraucht werden, landen in `ci.txt`, die die `base.txt` inkludiert. Für die lokale Entwicklungsumgebung wird noch ein bisschen mehr benötigt. Diese Datei heißt `dev.txt` und inkludiert die `ci.txt`. Zum Schluss gibt es noch die Datei `prod.txt` für die Serverumgebung mit dem Paket `unicorn`. Diese inkludiert wiederum nur die `base.txt`.

### 4.3 Versionsverwaltung

Die Verwendung eines Systems zur Versionsverwaltung ist bei Softwareprojekten mittlerweile Standard. In diesem Projekt wird dafür Git verwendet. Ihre richtige Stärke entwickeln solche Systeme mit einem Webdienst wie GitHub oder Gitlab mit weiteren Funktionen. Bei dieser Arbeit wird der Service von `gitlab.com` genutzt. Gitlab ist kostenlos nutzbar, bietet private Repositories und die Möglichkeit, automatische Tests durchzuführen.

Im Projekt werden ein `master` und ein `develop` Branch verwendet. Auf dem `master` Branch soll sich zu jeder Zeit nur Code befinden, der produktiv eingesetzt werden kann. Auf dem `develop` Branch befindet sich Code in der Entwicklung. Lokal wird noch mit Zweigen für neue Funktionen gearbeitet, die aber im Rahmen der Entwicklung lokal in den `develop` Branch zusammengeführt werden.

### 4.4 Durchführung von Tests und Coverage

Damit die Software stabil und sicher funktioniert, werden automatische Tests durchgeführt. Diese unterstützen auch das Ziel der guten Wartbarkeit und Möglichkeiten der einfachen Anpassung und Überarbeitung von Code. Es werden einerseits Unittests auf der Ebene der Datenmodels durchgeführt. Damit werden Funktionen der Models und das richtige Anlegen in der Datenbank getestet. Ebenfalls überprüft werden benötigte Felder und Abhängigkeiten.

Andererseits werden ganzheitliche Tests der Requests durchgeführt. Dabei werden einerseits die Funktionalität und Logik auf der Serverseite als auch die Templates getestet. Ebenfalls getestet werden hier die Zugriffsberechtigung auf bestimmte Seiten und Eigenschaften, je nach Benutzerberechtigungen. Beim Schreiben eines solchen Testfalls wurde beispielsweise ein Fehler gefunden, der es ermöglicht hätte, dass Übungsleiter auf fremde Abrechnungen zugreifen können.

Um zu überprüfen, ob die wichtigsten Codestellen von den Tests abgedeckt werden, wird `coverage.py` eingesetzt. Dieses Tool ermittelt die Testabdeckung, das heißt welche Zeilen durch die Tests ausgeführt wurden oder nicht. Dies ist vor allem hilfreich beim Erstellen der Tests, damit man möglichst alle Zeilen abdeckt. Es hilft aber auch dabei, nicht verwendeten Code ausfindig zu machen, den man gegebenenfalls entfernen kann.

Bei Django gibt es eine eigene Template-Syntax, in der auch Logik wie Bedingungen und Schleifen vorhanden sind. HTML-Templates werden allerdings vom `coverage.py` nicht berücksichtigt. Dafür existiert das Paket namens `django_coverage_plugin`, welches auch die Templates mit überprüft. Leider wurde dieses erst recht spät entdeckt und integriert. Dementsprechend gab es eine große Anzahl von Stellen in den Templates, die von den Tests nicht abgedeckt oder nicht benötigt wurden.

Zwischenzeitlich lag die Testabdeckung bei 99,8%. Innerhalb der letzten Tage mussten allerdings noch schnell einige Funktionen hinzugefügt werden, die aus Zeitgründen nicht mehr unter Test gestellt werden konnten. Zum Abgabezeitpunkt liegt die Testabdeckung mit Durchführung aller 185 Tests bei 93,7% und soll im Anschluss wieder auf über 99% gesteigert werden.

### 4.5 Continuous Integration

Einer der ersten Schritte nach der Initialisierung der Django-Anwendung war die Einrichtung automatischer Testausführung bei Gitlab. Mit jedem Einchecken und Veröffentlichenden einer Änderung bei Gitlab werden dort automatisch die Tests ausgeführt und die Testabdeckung ermittelt.

Die Tests werden bei Gitlab in sogenannten Runnern ausgeführt. Ein Runner ist eine Docker Container Umgebung, die ausgeführt wird. Bei Gitlab können vordefinierte Docker Abbilder oder selbst erstellte Abbilder verwendet werden. Der Nachteil von vordefinierten Umgebungen ist, dass bei jedem erneuten Ausführen erst alle Software und Abhängigkeiten installiert werden müssen. Das dauert einerseits etwas länger und erzeugt andererseits auf den Servern der Paketanbieter entsprechende Last. Der zeitliche Aspekt ist insofern relevant, dass es nur ein begrenztes Kontingent von 2000 Minuten Laufzeit pro Monat bei Gitlab gibt. Außerdem möchte man nicht allzu lange auf das Ergebnis von den Tests warten.

Aus diesen Gründen wurde ein eigenes Docker Abbild erzeugt. Dort werden einmal alle Abhängigkeiten installiert und das Abbild anschließend bei Gitlab ins Registry hochgeladen. Jetzt muss im Testprozess nur noch das Abbild geladen werden und die Testausführung kann schneller starten. Zu beachten ist dadurch aber, dass mit jeder Veränderung der Pakete auch das Abbild neu gebaut und hochgeladen werden muss. Das ist in der Entwicklung allerdings recht selten vorgekommen.

### 4.6 Continuous Deployment

Wenn alle Tests erfolgreich bestanden sind, wird im nächsten Schritt der Code auf den Server kopiert. Es gibt für dieses Projekt zwei Webserver:

- <https://abrechnungen.tusli.de>

#### 4. Entwicklungsumgebung

- <https://staging.abrechnungen.tusli.de>

Der erste Webserver wird für das produktive System verwendet. Der Staging-Server ist zum Testen und Ausprobieren von neuen Funktionen gedacht. Wenn die Tests auf dem git Branch `develop` erfolgreich sind, wird der Code auf den Staging-Server kopiert. Wenn der Code von `develop` in den `master` zusammengeführt wird und alle Tests dort ebenfalls erfolgreich sind, wird der Code auf das produktive System kopiert.

Per `ssh` und `rsync` werden die Dateien auf den entsprechenden Webserver kopiert. Eine Herausforderung war die Herstellung der Verbindung zum Server. Dafür wurde ein SSH-Schlüsselpaar erzeugt und der öffentliche Schlüssel auf dem Webserver hinlegt. Der private Schlüssel zum Webserver sollte aber weder in der Versionsverwaltung, noch im Docker Abbild integriert werden, da bei einer zukünftigen Öffnung des Git-Repositories für weitere Entwickler jeder Zugang zum Server bekommen könnte. Gitlab bietet dafür die Möglichkeit, Umgebungsvariablen für das Projekt zu definieren. Diese werden dann zur Laufzeit in den Docker Container reingereicht und können verwendet werden. Genauso wird mit den SMTP-Zugangsdaten für den Mailversand vorgegangen.

Nachdem die Dateien auf den Webserver kopiert wurden, wird per `ssh` auf dem Server das `deploy.sh` Skript aufgerufen. Mit diesem Skript werden alle installierten Pakete überprüft und gegebenenfalls nachinstalliert. Anschließend wird im Skript für den Start des WSGI-Servers `gunicorn` der richtige Pfad ersetzt. Dann werden die statischen Dateien (`css`, `js`, ...) in das richtige Verzeichnis kopiert und die Datenbankmigrationen aufgerufen. Zum Schluss wird der Webservice neu gestartet, damit die aktuelle Version ausgeliefert wird.

#### 4.7 Verwendete Datenbank

Für die Entwicklung und das Testen wird SQLite3 als Datenbank verwendet. Für das produktive System war die Umstellung auf MySQL geplant. Allerdings wurde festgestellt, dass Django eine MySQL Version von mindestens 5.5 benötigt, auf dem Webserver aber nur 5.1 zur Verfügung steht. Aus diesem Grund wurde erst einmal auf die Verwendung von MySQL verzichtet, weil dadurch ein Umzug auf einen anderen Webserver notwendig gewesen wäre. Damit hätten alle Deployment-Schritte angepasst werden müssen und auch die Einarbeitung in das System eines neuen Webserver wäre etwas länger geworden.

Diese Abhängigkeit hätte schon zu Beginn der Entwicklung auffallen und bei der Wahl des Servers berücksichtigt werden können. Auf der anderen Seite stehen die Erfahrungen mit dem System des Webserveranbieters, die zeitlich einen enormen Vorteil ausgemacht haben. Nach Abschluss der Arbeit soll allerdings ein Umzug auf einen

anderen Server durchgeführt werden. Dort ist dann der Einsatz von PostgreSQL geplant, welches von Django empfohlen wird.

## 5 Das neue Abrechnungssystem

In diesem Kapitel geht es um die eigentliche Entwicklung des neuen webbasierten Abrechnungssystems sowie um die Einführung des Produktivsystems.

### 5.1 Die Entwicklung

Die Entwicklung des Abrechnungssystems lässt sich in verschiedene Abschnitte gliedern. Den ersten Abschnitt bildet die Modellierung der Daten. Danach folgt die Benutzer- und Zugriffverwaltung. Zum Schluss werden der Aufbau der Seite und die implementierten Funktionen betrachtet.

#### 5.1.1 Modellierung der Daten

Die Modellierung der Datenstruktur mit Django ist sehr komfortabel und gut dokumentiert. Dafür muss allerdings die grundlegende Struktur der Daten und die Anforderungen an diese klar sein. Den Ausgangspunkt aller Daten im System bildet eine Person. Eine Person hat einen Vornamen und Nachnamen, die an vielen Stellen benötigt werden. Somit wurde zuerst die Klasse `Person` angelegt. Eine Klasse steht für eine Tabelle im Datenbankmodell mit den Feldern oder Eigenschaften als Spalten.

Eine `Abteilung` ist eine weitere grundlegende Klasse. Diese hat einen Namen und eine im Verein intern verwendete zweistellige Nummer.

Die nächste interessante Klasse ist ein `Trainer`. Ein `Trainer` ist gleichzusetzen mit einem Übungsleiter, im Code wird aber `Trainer` verwendet, weil es kürzer und ohne Umlaut ist. Ein `Trainer` hat eine Verknüpfung zu einer `Person` und einer `Abteilung`. Eine `Person` kann mit mehreren `Trainern` verknüpft sein, da ein Übungsleiter in mehreren Abteilungen tätig sein kann. Zusätzlich wird zu dem `Trainer` noch die Eigenschaft `von` und `bis` gespeichert. Damit wird erfasst, wann ein Übungsleiter im Verein angefangen hat und wann er seine Tätigkeit eventuell beendet hat. Diese Felder sind bereits mit hinzugefügt, werden allerdings in der weiteren Programmlogik noch nicht verwendet. In der Zukunft werden sie aber definitiv benötigt, weil Übungsleiter aufhören werden und dann auch keine Abrechnungen mehr einreichen dürfen.

Neben der Rolle der `Trainer` gibt es die `Abteilungsleitung`. Diese hat ebenfalls eine Verknüpfung zu einer `Person` und einer `Abteilung`. Zusätzlich gibt es noch das Feld `Funktion`, in dem die Funktion in der `Abteilungsleitung` erfasst wird. Beispiele dafür sind `Abteilungsleiter`, `Stellvertreter` oder `Kassenwart`. Auch hier kann eine `Person` in mehreren `Abteilungsleitungen` tätig sein. Ebenfalls sind hier die Felder für die Dauer in der `Abteilungsleitung` vorgesehen, da durch regelmäßig Wahlen auch hier Wechsel stattfinden können. Da die nächsten Wahlen aber erst im Oktober stattfinden, gibt es hier noch ein bisschen Zeit bis zur Implementierung.

Die dritte Rolle bilden die Mitarbeiter, die ebenfalls einer Person, nicht aber einer Abteilung zugeordnet sind. Wie bei der Abteilungsleitung wird hier die Funktion und die Dauer erfasst.

Mit einer Person können zusätzlich ein oder mehrere Trainerkonten verknüpft werden. Ein Trainerkonto besteht aus einer IBAN, BIC, einem Banknamen und einem optionalen Kontoinhaber, wenn der Name von der Person abweicht. Zusätzlich wird auch hier die Angabe der Gültigkeit erfasst. Diese ist wichtig, damit jede jemals verwendete Kontoverbindung des Trainers aus steuerrechtlichen Gründen erhalten bleibt.

Für die Abrechnung wird für jeden Übungsleiter ein vereinbarter Stundensatz benötigt. Im alten Abrechnungsprozess wurde dieser jedes Mal vom Übungsleiter in das Formular eingetragen. Dadurch musste die Abteilungsleitung bei jeder Abrechnung überprüfen, ob auch der richtige Stundensatz angegeben ist. Mit der Speicherung des vertraglich vereinbarten Stundensatzes kann diese Prüfung in Zukunft entfallen. Der Stundensatz beinhaltet eine Verknüpfung zum Trainer, den Betrag, eine Bezeichnung (Training oder Wettkampfbetreuung) sowie ebenfalls eine Gültigkeit. Die Gültigkeit wird in der aktuellen Implementierung bereits verwendet, da die Änderung des Stundensatzes regelmäßig vorkommt und von Anfang an benötigt wird

Somit kommen wir zum Hauptdatensatz, der Abrechnung. Eine Abrechnung ist einem Trainer und einer Abteilung zugeordnet. Die Zuordnung zu einer Abteilung wäre hier nicht notwendig, da der Trainer schon eindeutig einer Abteilung zugeordnet ist. In der Anwendung müssen die Abrechnungen aber häufig nach der Zugehörigkeit zu einer Abteilung gefiltert werden und somit wurde die Verknüpfung mit aufgenommen. Zu einer Abrechnung wird für die eindeutige Zuordnung zwingend das Jahr und der Monat benötigt. Zu einer Kombination von Jahr, Monat und Abteilung, darf ein Übungsleiter nur eine Abrechnung einreichen, was durch eine Bedingung in dem Modell garantiert wird. Ein weiteres Feld der Abrechnung ist der Abrechnungsbetrag. Für eine spätere Funktionalität wird eine Verknüpfung zu der Person gespeichert, die die Abrechnung angelegt hat. Damit soll es in Zukunft möglich sein, dass eine Person für einen anderen Trainer eine Abrechnung anlegen kann. Ein Anwendungsfall davon ist, wenn eine Abrechnung per Papier eingereicht und von der Abteilungsleitung in das System eingegeben wird. In diesem Fall entspricht dann die Person nicht dem Trainer.

Es werden zusätzlich folgende Statusinformation gespeichert:

- Zeitstempel, wann die Abrechnung durch den Übungsleiter eingereicht wurde
- Zeitstempel, wann die Abrechnung durch die Abteilungsleitung freigegeben wurde

## 5. Das neue Abrechnungssystem

- welcher Abteilungsleiter die Abrechnung freigegeben hat
- Zeitstempel, wann der Mitarbeiter die Abrechnung bestätigt hat
- welcher Mitarbeiter die Abrechnung bestätigt hat
- Zeitstempel, wann die Abrechnung zur Zahlung angewiesen wurde (Überweisung)
- welcher Mitarbeiter die Abrechnung zur Zahlung angewiesen hat

Als Bedingungen für das Speichern einer Abrechnung sind die nachfolgenden Regeln hinterlegt, die die Einhaltung der Reihenfolge des Freigabeprozesses garantieren.

- wer und wann für eine Aktion müssen immer zusammen gespeichert
- freigegeben kann nur gesetzt werden, wenn eingereicht gesetzt ist
- bestätigt kann nur gesetzt werden, wenn freigegeben gesetzt ist
- überwiesen kann nur gesetzt werden, wenn bestätigt gesetzt ist

Einige Übungsleiter im Verein haben mehrere Stundensätze. Je nach Gruppe rechnen sie unterschiedlich ab. Bisher war eine solche Abrechnung nur mit zwei Abrechnungsbögen möglich. Das neue System soll dies in einer Abrechnung ermöglichen. Des weiteren haben viele Abteilungen einen zusätzlichen Stundensatz für Wettkampfbetreuung, der sich mit dem alten Formular nicht sauber abrechnen ließ. Diese beiden Fälle werden jetzt mit der folgenden Modellierung möglich. Mit dem Anlegen einer Abrechnung wird für jeden zu diesem Zeitpunkt gültigen Stundensatz ein AbrechnungsStundensatz angelegt. Das ist eine Klasse mit einer Verknüpfung zu einer Abrechnung und einem Stundensatz. Pro Abrechnung kann es mehrere von diesen Zuordnungen geben. Zusätzlich wird die Anzahl der geleisteten Stunden und der Gesamtbetrag gespeichert. Der Gesamtbetrag errechnet sich aus der Multiplikation von Stunden und Stundensatz.

Mit der Abrechnung müssen durch die Übungsleiter die geleisteten Stunden angegeben werden. Diese werden als Klasse Training modelliert. Ein Training hat eine Verknüpfung zu einem AbrechnungsStundensatz, ein Datum, ein Trainingsbeginn und -ende sowie eine Dauer und Beschreibung.

Eine Alternative wäre gewesen, das Training jeweils direkt mit einer Abrechnung zu verknüpfen und zum Training den Stundensatz anzugeben. Das hätte zur Folge, dass man pro Training die Dauer mit dem Stundensatz multipliziert. Wenn man dieses Ergebnis dem Benutzer anzeigt, können Rundungsfehler auftreten. Nehmen wir zum Beispiel 1,5 Stunden und einen Stundensatz von 9,19 (Mindestlohn). Das macht als Ergebnis 13,785 und würde dem Nutzer kaufmännisch gerundet als 13,79 angezeigt werden. Wenn er in einem Monat zwei solche Trainingseinheiten absolviert hat, macht



das nach der Darstellung  $13,79 + 13,79 = 27,58$ . Intern mit einer höheren Genauigkeit gerechnet, ergibt es aber  $13,785 + 13,785 = 27,57$ . Es erscheint in der Darstellung ein Unterschied von einem Cent. Aus diesem Grund werden bei der genutzten Modellierung erst alle Stunden aufsummiert und anschließend einmal mit dem Stundensatz multipliziert. Zusätzlich wird intern mit sechs Nachkommastellen gearbeitet.

Unter anderem aufgrund des vorangegangenen Beispiel wurde bei allen Feldern und Berechnungen in Verbindung mit Geldbeträgen anstatt der `float` die Klasse `Decimal` von Python verwendet.

Zusammen ergeben diese Klassen das Datenmodell für die neue Abrechnungssoftware.

### 5.1.2 Die Benutzer- und Zugriffsverwaltung

Die Benutzer- und Zugriffsverwaltung ist eine wichtige Komponente für das Abrechnungssystem. Im System werden persönliche und finanzielle Daten gespeichert, die es zu schützen gilt. Dafür wird die Benutzerverwaltung von Django verwendet, die noch um das `django-allauth` Modul ergänzt wird. Im aktuellen Stand der Software werden die Funktionen für das Registrieren, das An- und Abmelden sowie die „Passwort vergessen“ Funktion vom System verwendet. Funktionen wie „Passwort ändern“ sind zwar vorhanden, aber noch nicht in der Oberfläche verlinkt.

Die Nutzung der Webanwendung ist nur mit einem Benutzeraccount möglich. Auf der Webseite kann sich allerdings jeder einfach registrieren. Damit noch kein Zugriff auf Funktionen des Systems möglich ist, müssen alle Benutzer händisch freigegeben werden, indem ihnen die Gruppe `member` zugeordnet wird. Für diese Zuordnung wird das Django Admininterface verwendet. Einen Zugang zu diesem hat derzeit nur der Ersteller der Software, sowie zwei Mitarbeiter der Geschäftsstelle. Alle anderen Nutzer können sich nicht im Admininterface anmelden.

Selbst wenn ein Benutzer der Gruppe `member` zugeordnet ist, kann er in dem System noch keine Funktionen nutzen. Dies ist erst möglich, wenn der Benutzer einer Person zugeordnet wird. Dieser Prozess erfolgt ebenfalls über das Admininterface.

Die Berechtigungen zum Zugriff auf bestimmte Funktionen werden anhand der zu der Person zugeordneten Rollen ermittelt. Eine Übungsleiter darf auf seine bisherigen Abrechnungen zugreifen, neue Abrechnungen erstellen und bearbeiten sowie Abrechnungen einreichen.

Eine Person mit einer Abteilungsleitung hat die Möglichkeit, alle Übungsleiter ihrer Abteilung zu sehen. Zusätzlich kann sie eingereichte Abrechnungen dieser Übungsleiter betrachten, jedoch keine noch nicht eingereichten Abrechnungen. Diese Person hat außerdem die Möglichkeit, Abrechnungen freizugeben oder abzulehnen.

## 5. Das neue Abrechnungssystem

Eine Person mit einem verknüpften Mitarbeiter sieht alle Übungsleiter und Abteilungen. Sie kann alle Abrechnungen der Übungsleiter einsehen, die bereits freigegeben sind. Sie kann außerdem Abrechnungen bestätigen oder ablehnen.

Somit sind die Berechtigungen fest an die Funktionen der Person im Verein geknüpft. Sie spiegeln die Möglichkeiten wider, die sie im alten Abrechnungsprozess auch schon hatten.

### 5.1.3 Aufbau der Seite und Funktionen

Die Seite ist so aufgebaut, dass für jede Rolle (Übungsleiter, Abteilungsleitung, Mitarbeiter) ein eigener URL-Baum existiert. Zu diesem Bereich gelangt der Benutzer über eine Liste von Links auf der Hauptseite. Wenn eine Person nur die Rolle Übungsleiter hat, sieht sie dafür auch nur den Link.

Auf der ersten Ebene des jeweiligen Bereichs wird eine Übersicht mit den wichtigsten Daten und Links dargestellt. Ein Übungsleiter sieht auf dieser Seite bereits seine letzten Abrechnungen und hat die Möglichkeit, eine neue Abrechnung anzulegen. Eine angelegte Abrechnung kann durch den Übungsleiter gespeichert werden und zu einem späteren Zeitpunkt weiter bearbeitet werden. Das ist eine Funktion, die sich einige Übungsleiter gewünscht haben und die mit dem alten Formular auch möglich war. Wenn die Abrechnung vom Übungsleiter fertig ausgefüllt ist, kann er diese durch einen Klick einreichen. Mit dem Einreichen wird die Abrechnung für die Abteilungsleitung sichtbar. Sollte der Übungsleiter jetzt einen Fehler feststellen und die Abrechnung noch einmal bearbeiten wollen, kann er die Einreichung zurückziehen. Damit kann die Abrechnung wieder ganz normal bearbeitet werden und muss erneut eingereicht werden.

Die Abteilungsleitung sieht eine Liste aller Übungsleiter ihrer Abteilung mit einer jeweiligen Verlinkung zur Detailansicht des Übungsleiters. In der Detailansicht werden die zum Übungsleiter hinterlegten Stundensätze sowie die letzten eingereichten Abrechnungen aufgelistet. Ein Klick auf eine Abrechnung führt zur Detailansicht der Abrechnung. Diese kann von der Abteilungsleitung nicht geändert, sondern nur freigegeben oder abgelehnt werden. Eine Freigabe sorgt dafür, dass ein Übungsleiter seine Abrechnung nicht mehr bearbeiten kann. Sollte die Abteilungsleitung einen Fehler nach der Freigabe bemerken, kann sie diese zurückziehen. Das hat genauso wie eine Ablehnen zur Folge, dass auch die Einreichung durch den Übungsleiter rückgängig gemacht wird. Die Idee ist, dass nach einer Ablehnung der Übungsleiter die Abrechnung noch einmal überarbeiten und erneut einreichen muss.

Die Abteilungsleitung hat zusätzlich eine Tabelle mit allen Trainern als Zeilen und den Monaten eines Jahres als Spalten. In den Zellen steht der Gesamtbetrag der Abrechnung des Übungsleiters zu einem Monat. Eine solche Tabelle wurde bisher in

der Geschäftsstelle geführt und den Abteilungsleitungen auf Nachfrage zur Verfügung gestellt. Mit dieser Tabelle ist für die Abteilungsleitung schnell sichtbar, welcher Übungsleiter seine Abrechnung schon eingereicht hat und wer noch fehlt. Zusätzlich wird der Betrag von nicht freigegebenen Abrechnungen rot und in Klammern dargestellt und freigegebene in grün ohne Klammern. Durch Klick auf die Zelle kommt man ebenfalls zu der Detailansicht der Abrechnung.

Die Darstellung für die Mitarbeiter ist ähnlich der Abteilungsleitung, nur dass sie alle Abteilungen sehen und auswählen können. Von dort aus können sie die Detailansicht eines Übungsleiters und die entsprechenden Abrechnungen erreichen. Diese Funktion wird eher für Zwecke der Recherche und Auskunft benötigt.

Für die eigentliche Aufgabe der Bearbeitung der Abrechnungen gibt es eine separate Ansicht. Dort werden alle Abrechnungen aufgelistet, die von der Abteilungsleitung freigegeben, von der Geschäftsstelle aber noch nicht bestätigt wurden. Die Aufgabe des Mitarbeiters ist es jetzt, jede Abrechnung dieser Liste zu prüfen und zu bestätigen. Dabei soll er vor allem darauf achten, wer die Abrechnung freigegeben hat. Damit soll verhindert werden, dass ein Abteilungsleiter, der gleichzeitig Trainer ist, seine eigene Abrechnung freigibt. Derzeit ist dieser Fall möglich, soll in Zukunft aber eventuell vom System automatisch unterbunden werden. Im alten Prozess ist es allerdings auch möglich, dass jemand seine eigene Abrechnung als Abteilungsleitung unterschreiben könnte. Der Mitarbeiter kann bei Einwänden zur Abrechnung diese wie die Abteilungsleitung ablehnen. Das hat wie bei der Abteilungsleitung zur Folge, dass alle Freigaben und Einreichungen zurückgesetzt werden.

Zusätzlich zu der Liste der zu bearbeitenden Abrechnungen existiert eine Ansicht der zu überweisenden Abrechnungen. Hier werden alle Abrechnungen angezeigt, die bestätigt, aber noch nicht als überwiesen gekennzeichnet sind. In dieser Ansicht hat der Mitarbeiter zwei Funktionen. Die erste Funktion ermöglicht das Herunterladen der angezeigten Abrechnungen als CSV-Datei. Diese Datei hat das Importformat für das im Verein verwendete Bankprogramm. Der Mitarbeiter importiert diese Datei und damit sind alle Überweisungen fertig. Jetzt benötigt er die zweite Funktion um alle angezeigten Abrechnungen als Überwiesen zu kennzeichnen. Damit ist der Prozess für den Mitarbeitern dann abgeschlossen und es müssen keine Abrechnungen mehr eingescannt oder abgeheftet werden.

## 5.2 Die Einführung des Systems im Verein

Ein entwickeltes System erfüllt erst seinen Nutzen, wenn es auch eingesetzt und verwendet wird. Das Ziel der Arbeit war, das System soweit zu entwickeln, dass es produktiv zum Einsatz kommt.

Die Übungsleiter und Abteilungsleitungen haben mit Versendung der Umfrage zum

## 5. Das neue Abrechnungssystem

alten Abrechnungsprozess erstmals von den Planungen und der Entwicklung eines webbasierten Abrechnungssystems erfahren. Die Rückmeldungen dazu waren zu einem großen Teil positiv, auch wenn es einzelne ablehnende Haltungen gab.

Mit wachsendem Funktionsumfang wurde der Stand der Software in regelmäßigen Abständen den Vorstandsmitgliedern und Mitarbeitern gezeigt. Anschließend konnten sich einzelne Benutzer auf dem Testsystem registrieren, was allerdings bis auf zwei Personen keiner wahrgenommen hat. Die zwei Personen sind mit dem System gut zurecht gekommen. Als nächstes sollte der Test auf weitere Personen ausgedehnt werden. Das System war mittlerweile so gut durch automatische und manuelle Tests überprüft, sodass direkt mit dem produktiven Einsatz gestartet wurde.

Die zukünftigen Nutzer wurden aufgefordert, sich einen Benutzerzugang im Produktivsystem anzulegen. Anschließend sollte eine Mail mit Angabe von Benutzername, Vorname und Nachname sowie Abteilung an den Entwickler geschickt werden. Durch die E-Mail können die Personen verifiziert werden, da die verwendeten E-Mailadressen der Trainer bekannt sind. Wenn eine Nachricht von einer nicht bekannten E-Mailadresse kommt, muss geprüft werden, ob es sich um einen berechtigten Nutzer handelt. Die Gefahr eines nicht erlaubten Nutzers hält sich in Grenzen, da nur die Stundensätze, nicht aber die Kontodaten einsehbar sind. Die Kontodaten sind auch nicht über das Frontend änderbar, sodass ein Betrug nicht möglich ist.

Die ersten Nutzer waren bereits nach einer halben Stunde registriert und vom Entwickler verknüpft. Kurze Zeit später waren die ersten Abrechnungen im System und die erste positive Rückmeldung mit konstruktiver Kritik im Mailpostfach.

*„Es ging sehr schnell mit der Abrechnung. Ich empfinde die Benutzerführung als gelungen und zeitgemäß. Die Eingaben sind sehr einfach zu machen. Einzig die aufpoppende Uhr bei der Uhrzeit finde ich störend. Es geht für mich schneller, wenn ich die Zeiten per Tastatur eintrage.“*

Zitat eines Übungsleiters der Leichtathletikabteilung

In den folgenden vier Tagen registrierten sich bereits fast 50 Nutzer, was etwas mehr als ein Viertel der bisherigen Beteiligten am Prozess entspricht. Die Übungsleiter legten innerhalb kürzester Zeit 38 Abrechnungen an, welche zum Teil auch gleich von den Abteilungsleitungen freigegeben wurden.

Drei Tage nach der Einführung wurden von einer Mitarbeiterin der Geschäftsstelle die ersten sechs Abrechnungen bestätigt, exportiert, erfolgreich in das Bankprogramm importiert und überwiesen. Anschließend wurden die Abrechnungen noch als überwiesen gekennzeichnet und damit ist der Prozess beendet.

Damit wird die Einführung durch den Entwickler als erfolgreich bewertet. Sie ist an dieser Stelle aber noch nicht als abgeschlossen anzusehen. Das Ziel ist, dass möglichst

schnell alle Beteiligten auf das Abrechnungssystem umsteigen.

Von vielen Benutzer wurden Rückmeldungen erhalten, welche durchweg positiv waren. Für alle war die Erstellung der Abrechnung schneller und einfacher und das, obwohl bei weitem noch nicht alles perfekt ist. Es wurden von fast jedem Nutzer gute Verbesserungsvorschläge geliefert, die sich zum Teil überschneiden und vorab schon bekannt waren.

Ein Beispiel ist, dass die Bildschirmbreite teilweise nicht voll genutzt wird. Das liegt an einem Kompromiss zwischen Nutzbarkeit auf dem Handy und PC. Weiterhin fehlt die Funktionalität, einmal angelegte Trainings zu löschen. Ebenso lassen sich keine Abrechnungen löschen. Für die Eingabe der Uhrzeiten wurde ein Widget eingebaut, dass für Touchgeräte optimiert ist. Am PC ist es aber eher hinderlich. Die Lösung wird sein, das Widget nur bei der Verwendung von einem Smartphone anzuzeigen.

## 6 Ausblick

Zu Beginn der Arbeit wurde von einem überschaubaren und klar abgegrenzten Projekt ausgegangen. Im Lauf der Entwicklung hat sich aber gezeigt, dass dieses Abrechnungssystem erst der Anfang von etwas viel größerem sein wird. Es sind die folgenden Ideen und Erweiterungsmöglichkeiten entstanden, die über den Rahmen der Arbeit hinaus entwickelt und hinzugefügt werden sollen.

An erster Stelle stehen natürlich kleinere Verbesserungen des System aufgrund der Rückmeldungen der Nutzer. Zu nennen sind hier die Verbesserung der Darstellung auf kleinen Geräten wie Smartphone oder sehr großen Monitoren. Ebenfalls wichtig ist die Möglichkeit, Abrechnungen und Trainings löschen zu können.

Gerade in Hinblick auf den Ausbau des Systems ist ein Wechsel der Datenbank auf PostgreSQL von hoher Priorität.

Eine wichtige und schon von den Nutzern angefragte Funktionalität ist die Versendung von automatischen E-Mailbenachrichtigungen. Es soll zum Beispiel eine E-Mail erstellt werden, wenn eine neue Abrechnung eingereicht wurde. Ebenso soll eine Erinnerungsmail erstellt werden, wenn bis zum fünften Tag des Folgemonats die Abrechnung noch nicht erstellt und eingereicht worden ist. Der Empfang von E-Mails soll durch jeden Benutzer konfigurierbar sein.

Ebenfalls eine E-Mail sollen die Beteiligten erhalten, wenn eine Abrechnung abgelehnt wird. Zu diesem Zweck soll die Oberfläche um ein Feld für den Grund der Ablehnung erweitert werden, damit der Empfänger sofort weiß, warum die Abrechnung abgelehnt wurde. Im aktuellen Stand der Software muss diese Kommunikation außerhalb des Systems erfolgen.

Auf den bisherigen Abrechnungen mussten lizenzierte Übungsleiter jedes Mal ihre Lizenznummer und Gültigkeit angeben. Dies entfällt mit dem neuen Prozess. Allerdings wird im Verein weiterhin die Lizenzinformation benötigt und deswegen soll das System um eine Verwaltung von Übungsleiterlizenzen erweitert werden. Damit sind dann auch die Beantragung von Zuschüssen sowie deren Abrechnung beim Landessportbund möglich. Dafür werden genau die Angaben Trainer, Lizenz, geleistete Stunden, der Stundensatz sowie die ausgezahlte Gesamtsumme pro Jahr benötigt.

Von einigen Übungsleitern kam auch der Vorschlag, eine Vorlage von Trainingseinheiten einer regulären Trainingswoche zu hinterlegen. Anhand dieser soll die Abrechnung bei der initialen Erstellung automatisch ausgefüllt werden. Der Übungsleiter muss dann nur noch Termine anpassen oder löschen. Das ergibt insofern Sinn, weil viele Übungsleiter jede Woche die gleichen Trainingszeiten und Gruppen haben. Diese Funktion sollte recht schnell umzusetzen sein.

Aus Sicht der Mitarbeiter, aber auch Abteilungsleitungen, sollte die Aufnahme von neuen Übungsleitern über das System möglich sein. Bisher werden die Stammdaten eines neuen Trainers über ein Formular auf der Vereinswebseite erfasst. Diese Daten werden zu einem großen Teil aber auch im Abrechnungssystem benötigt. Somit wäre es sinnvoll, dieses Formular in das Abrechnungssystem zu integrieren.

Mit dem Abrechnungssystem soll es auch möglich sein, weiterhin eingereichte Papierabrechnungen einzupflegen. Es wird einige wenige Übungsleiter geben, die weiterhin Abrechnungen per Papier einreichen werden. Diese Abrechnungen sollen nach Möglichkeit von der Abteilungsleitung eingegeben werden, damit alle Abrechnungen gesammelt an einer zentralen Stelle gespeichert werden.

Nach dem Ende eines Jahres erstellen die Mitarbeiter für jeden Übungsleiter eine Verdienstbescheinigung, in der die in einem Kalenderjahr ausgezahlten Beträge aufgeführt werden. Diese Bescheinigung ist zum Beispiel für die Steuererklärung der Übungsleiter hilfreich. Es bietet sich an, diese Bescheinigung aus dem System heraus zu generieren und jedem Übungsleiter zum Download anzubieten, da die Daten vorhanden sind. In die gleiche Richtung geht die Anforderung, immer aktuell die im Jahr bereits abgerechnete Gesamtsumme anzuzeigen. Diese wird von einigen Übungsleitern zur Überprüfung benötigt, ob sie noch unter der Freibetragsgrenze von 2.400 Euro pro Jahr liegen.

Der Vorschlag von einem Übungsleiter, das Führen der Anwesenheitslisten beim Training über dieses System zu erledigen, ist durchaus eine Überlegung wert. Allerdings kann man dann nicht mehr von einem reinen Abrechnungssystem, sondern eher einer Vereinsverwaltung reden. Damit soll die Liste der möglichen Weiterentwicklungen beendet werden, auch wenn sicherlich noch einige weitere gute Ideen für das System existieren. Der Grundstein dafür ist zumindest gelegt.

## 7 Zusammenfassung

Diese Arbeit macht deutlich, was durch eine Digitalisierung eines vormals auf Papier basierten Prozesses möglich ist. Benötigten die Übungsleiter bisher im Schnitt circa zehn Minuten für die Erstellung einer Abrechnung, sind sie jetzt um einiges schneller. Genaue Zahlen liegen dafür noch nicht vor, allerdings haben alle Übungsleiter mitgeteilt, dass sie schneller waren. Wenn man dazu noch berücksichtigt, dass sie das erste Mal mit dem neuen System gearbeitet haben und die Eingabemöglichkeit der Uhrzeit noch optimiert werden kann, sind sie in Zukunft noch mal um einiges schneller. Wenn die Funktion mit der automatischen Initialisierung anhand einer vordefinierten Woche hinzukommt, wird der Zeitaufwand für einen Übungsleiter auf ein Bruchteil sinken. Dann sollte eine Abrechnung innerhalb von zwei Minuten zu erledigen sein.

Für die Mitarbeiter ist die zeitliche Ersparnis ebenfalls erheblich. Wurden für die Bearbeitung der Abrechnung in der Vergangenheit im optimalen Fall sechs Stunden pro Monat benötigt, reduziert sich der Aufwand mindestens um den Faktor fünf, wenn nicht sogar mehr. Hinzu kommt, dass die Daten alle in einem einheitlichen Format an einem zentralen Ort vorliegen. Weitere zeitliche Einsparungen sind durch die neuen Funktionen, wie der Beantragung und Abrechnung von Zuschüssen, sowie der automatischen Ausstellung der Verdienstbescheinigungen zu erwarten.

Im Rahmen dieser Arbeit ist ebenfalls deutlich geworden, wie der konsequente Einsatz von modernen Techniken der Softwareentwicklung zu einem guten Produkt führen kann. Es ist schon sehr erstaunlich, wie reibungslos die Einführung des Abrechnungssystems funktioniert hat. Dazu beigetragen haben sicherlich nicht unerheblich die gute Testabdeckung sowie der Prozess, Änderungen erst auf dem Staging-Server zu testen, bevor sie ins Produktivsystem übertragen werden. Ebenfalls ausgezahlt hat sich das automatische Deployment, wodurch jederzeit eine neue Softwareversion innerhalb von zwei Minuten ohne händische Eingriffe veröffentlicht werden kann.

Es ist ein tolles Gefühl, dass die Einführung ohne Probleme funktioniert hat und die Lösung von den Nutzern so positiv und offen aufgenommen wurde. Bemerkenswert sind auch die sehr konstruktiven Verbesserungsvorschläge. Das ist eine gute Basis für eine Weiterentwicklung des Systems.

Es ist angedacht, den Quelltext als OpenSource zu veröffentlichen. Dafür müssen allerdings noch personenbezogene Testdaten ersetzt werden.

Der Softwarestand zum Zeitpunkt der Abgabe spiegelt sich in folgendem Git-Hash wieder: `4cf c28b752795b9af3c8ad495788cc91d3673b52`



## Literaturverzeichnis

- [1] Django Dokumentation, <https://docs.djangoproject.com/en/2.0/>, Jan. 2019
- [2] Django Releaseplan, <https://www.djangoproject.com/download/>, Jan. 2019
- [3] Angular Dokumentation | Releases, <https://angular.io/guide/releases>, Jan. 2019
- [4] Stackoverflow, Es wurde reger Gebrauch von Stackoverflow gemacht, <https://stackoverflow.com/>, Jan. 2019
- [5] Classy Class-Based Views, <https://ccbv.co.uk/>, Jan. 2019
- [6] Uberspace Wiki (Server), <https://wiki.uberspace.de/cool:django>, Jan. 2019
- [7] gitlab.com (Git, CI und CD-Server, <https://gitlab.com/>, Jan. 2019
- [8] Django Vanilla Views, <http://django-vanilla-views.org/>, Jan. 2019
- [9] Material Design Lite (CSS Framework), <https://getmdl.io/>, Jan. 2019
- [10] Materialize (CSS Framework), <https://materializecss.com/>, Jan. 2019
- [11] django-allauth, <https://github.com/pennersr/django-allauth>, Jan. 2019
- [12] django-coverage-plugin, [https://github.com/nedbat/django\\_coverage\\_plugin](https://github.com/nedbat/django_coverage_plugin), Jan. 2019
- [13] coverage.py, <https://coverage.readthedocs.io/en/v4.5.x/>, Jan. 2019

