

Dokumentation zur betrieblichen Projektarbeit

BerichtsheftTool

Erweiterbare PHP basierte Anwendung zum automatischen Erzeugen von Ausbildungsnachweisen aus einer bestehenden Zeiterfassungs-Software.

Pascal Wild

Fachinformatiker Fachrichtung Anwendungsentwicklung

Ausbildungsbetrieb:
EOS UPTRADE GmbH
Schanzenstraße 70
20357 Hamburg

Prüfungsbewerber:
Pascal Wild
Rittmerskamp 45
22415 Hamburg

www.eos-uptrade.de

pascal.wild@eos-uptrade.de

Inhaltsverzeichnis

Inhaltsverzeichnis.....	1
Einführung.....	3
Projektdefinition.....	3
Projektumfeld.....	3
Ist-Situation.....	3
Projektziel.....	3
Projektabgrenzungen.....	3
Schnittstellen.....	3
Projektplanung.....	4
Zeitplanung der Projektphasen.....	4
Ressourcenplanung.....	4
Kostenplanung.....	5
Einmalige Entwicklungskosten.....	5
Aufwand pro Woche und Auszubildenden.....	5
Kosten für die Erstellung der Berichtshefte im Jahr.....	5
Unterhaltungskosten.....	6
Kostenvergleich.....	6
Dauer bis zum Ausgleich der Investition.....	6
Nutzwertanalyse.....	6
Projektdurchführung.....	6
Definitionsphase.....	7
Planungsphase.....	7
Realisierungsphase.....	7
Testphase.....	8
Dokumentation.....	9
Übergabe.....	9
Projektbewertung.....	10
Zielerreichung.....	10
Soll-/Ist-Abgleich.....	10
Weiterentwicklung.....	11
Quellcodeauszüge.....	12
Glossar.....	13
Git.....	13
GitHub.....	13
Repository.....	13
Jira.....	13
JSON.....	13
OSX.....	13
PHPStorm.....	13
LAMP.....	13
Composer.....	14
Bootstrap.....	14
Rest API.....	14
JQL.....	14
Capistrano.....	14
Symfony.....	14
Bundle.....	14
Tabellenverzeichnis.....	15

Anhänge.....	16
--------------	----

Einführung

Unterstrichene Wörter werden im Glossar näher erläutert.

Das gesamte Projekt befindet sich auf der Open Source Plattform „GitHub“ unter Versionskontrolle und kann unter der URL „<https://github.com/PalliDotCom/Berichtsheft>“ aufgerufen werden.

Projektdefinition

In den folgenden Punkten wird das Projektumfeld, die Ist-Situation, das Projektziel, die Projektabgrenzungen und die Schnittstellen erläutert.

Projektumfeld

In der EOS UPTRADE GmbH wird seit dem Oktober 2013 die weit verbreitete Projektmanagement-Software „Jira“ zum Verwalten von Projekten eingesetzt. Die Mitarbeiter sind dazu verpflichtet Ihre geleistete Arbeit in dieser Projektmanagement-Software zu erfassen, des Weiteren müssen die Mitarbeiter neben der Dauer der geleisteten Arbeit auch eine Beschreibung der Arbeit verfassen.

Den Auszubildenden wird das Führen von Ausbildungsnachweisen in den nach § 14 Berufsbildungsgesetz (BBiG) erlassenen Ausbildungsordnungen vorgeschrieben (Quelle: <http://www.hk24.de>). Die Auszubildenden der EOS UPTRADE GmbH müssen Ihre Arbeit dementsprechend doppelt verwalten.

Ist-Situation

Aktuell verwalten die Auszubildenden der EOS UPTRADE GmbH Ihre geleistete Arbeit in der Projektmanagement-Software „Jira“ und zusätzlich in Ihrem Berichtsheft. Es gibt eine Vorlage im Form einer Word-Datei, welchen den Auszubildenden zur Verfügung gestellt wird.

Projektziel

Im Laufe der Projektabwicklung ist eine Webanwendung zu entwickeln, welche für die Auszubildenden der EOS UPTRADE GmbH ein Berichtsheft aus den Daten der Zeiterfassungs-Software generiert. Die Auszubildenden sollen Ihre geleistete Arbeit nur noch in der Projektmanagement-Software verwalten. Durch die zwingende Angabe einer Beschreibung der Arbeit können automatisch Nachweise für das Berichtsheft erstellt werden. Weiterhin ist eine Überprüfung des generierten Nachweises der zu erbringenden Software zwingend erforderlich um die Inhalte gegebenenfalls zu ergänzen.

Projektabgrenzungen

Es wird nur die Generierung aus der bestehenden Projektmanagement-Software der EOS UPTRADE GmbH unterstützt. Die Benutzerverwaltung wird nur rudimentär implementiert.

Außerdem sind die Routinen für die Installation dieser Anwendung auf einem Webserver nicht zu entwickeln und zu realisieren.

Schnittstellen

Die Anwendung kommuniziert mit der bestehenden Projektmanagement-Software über eine REST API im JSON Format. Aus der Schnittstelle werden die Daten zur Generierung der Nachweise entnommen.

Projektplanung

In den folgenden Abschnitten wird die Zeitplanung, die Ressourcenplanung und die Kostenplanung erläutert.

Zeitplanung der Projektphasen

Projektphase	Aufgabenbeschreibung	Soll-Stunden
Definitionsphase	Ist-Analyse	2
	Soll-Analyse	2
	Pflichtenheft erstellen	3
Planungsphase	Gestaltung der GUI	3
	Recherche Technologien zur PDF-Generierung	3
Realisierungsphase	Einrichten des Projekts	1
	Einrichten der Entwicklungsumgebung	1
	Umsetzung der GUI	3
	Kommunikation zur bestehenden Zeiterfassungs-Software	14
	Generierung eines PDF's aus den Daten der Zeiterfassungs-Software	15
Testphase	Erstellen / Einrichten von Tests	5
	Tests durchführen	2
	Auswertung der Ergebnisse	1
Dokumentation	Projektdokumentation	10
	Kurze Anwenderdokumentation	2
	Ist-Soll-Vergleich	1
Übergabe	Deployment	1
	Einweisung betroffener Mitarbeiter	1
Summe		70

Tabelle 1: Zeitplanung der Projektphasen

Ressourcenplanung

Die Entwicklung und Umsetzung der Webanwendung wird auf einem Arbeitsplatzrechner mit OSX durchgeführt. Auf dem Rechner ist die Entwicklungsumgebung „PHPStorm“ installiert, mit dem die nötigen Quellcode Dateien erstellt werden. Für die Versionsverwaltung wird die Open Source Plattform GitHub mit der dazugehörigen Versionsverwaltungs-Software „git“ verwendet. Der dazugehörige Client ist auf dem Arbeitsplatzrechner installiert. Weitere Voraussetzungen ist die neueste Version des Open Source Frameworks Symfony.

Zum Betreiben der Webanwendung wird ein LAMP – System benötigt.

Kostenplanung

Die Kostenplanung beschreibt die Wirtschaftlichkeit der entwickelten Software und führt die Kostendetails auf.

Stunden-Satz: 90€

Einmalige Entwicklungskosten

Projektphase	Zeitaufwand	Kosten
Definitionsphase	7 Stunden	630,00 €
Planungsphase	6 Stunden	540,00 €
Realisierungsphase	34 Stunden	3.060,00 €
Testphase	8 Stunden	720,00 €
Dokumentation	13 Stunden	1.170,00 €
Übergabe	2 Stunden	180,00 €
Summe	70 Stunden	6.300,00 €

Tabelle 2: Einmalige Entwicklungskosten

Aufwand pro Woche und Auszubildenden

Altes Verfahren	Neues Verfahren
1,5 Stunden	0,25 Stunden

Tabelle 3: Aufwand pro Woche und Auszubildenden

Zur Zeit werden in jeder Woche für jeden Auszubildenden der EOS UPTRADE GmbH 1,5 Stunden geplant. Bei den Testergebnissen konnte mit der neuen Software ein Berichtsheft für eine Woche mit wenigen Klicks generiert werden. Für das Überprüfen und gegebenenfalls Berichtigen des Ausdrucks werden pro Woche 0,25 Stunden veranschlagt.

Kosten für die Erstellung der Berichtshefte im Jahr

	Altes Verfahren	Neues Verfahren
Anzahl Auszubildender	8 Auszubildende	8 Auszubildende
Aufwand pro Woche	12 Stunden	2 Stunden
Aufwand pro Jahr	624 Stunden	104 Stunden
Summe	56.160,00 €	9.360,00 €

Tabelle 4: Kosten für die Erstellung der Berichtshefte im Jahr

Zur Einfachheit wurden für das Jahr 52 Wochen veranschlagt. Es gibt jedoch auch Urlaubswochen in denen kein Berichtsheft geführt werden muss.

Unterhaltungskosten

	Altes Verfahren	Neues Verfahren
Wartung der Anwendung	0,00 €	450,00 €
Server	0,00 €	60,00 €
Summe	0,00 €	510,00 €

Tabelle 5: Unterhaltungskosten

Für die Wartung der Software werden im Jahr 5 Stunden kalkuliert. Die Kosten für den Server werden auf 60,00 € im Jahr kalkuliert. Eine genaue Angabe der Werte ist nicht möglich.

Kostenvergleich

	Altes Verfahren	Neues Verfahren
Summe Kosten für die Erstellung der Berichtshefte im Jahr	56.160,00 €	9.360,00 €
Summe Unterhaltungskosten	0,00 €	510,00 €
Gesamtkosten	56.160,00 €	9.870,00 €
Ersparnis jährlich		46.290,00 €

Tabelle 6: Kostenvergleich

Dauer bis zum Ausgleich der Investition

Entwicklungskosten : Ersparnis = Dauer

6.300,00 € : 46.290,00 € \approx 0,14 Jahre \approx 52 Tage \approx 2 Monate

Die Entwicklungskosten würden bei einer Anzahl von 8 Auszubildenden nach etwa 2 Monaten gedeckt sein.

Nutzwertanalyse

Kriterium	Gewichtung	Bewertung (neu)	Bewertung (alt)
Zuverlässigkeit	5	10	5
Einheitliches Format	4	10	5
Pflegeaufwand	7	10	2
Abhängigkeit	6	2	10
Summe	22	172	119
Wirtschaftlichkeitskoeffizient		7,8	5,4

Tabelle 7: Nutzwertanalyse

Anhand der Wirtschaftlichkeitskoeffizienten ist zu sehen, dass mit der neu erstellten Software 44% wirtschaftlicher, als mit der manuellen Erstellung des Berichtsheftes gearbeitet werden kann.

Projektdurchführung

In den folgenden Abschnitten werden die Tätigkeiten bis zur fertigen Umsetzung der Anwendung beschrieben. Die Beschreibung der Tätigkeiten werden mit dem Zeitaufwand ergänzt.

Definitionsphase

Tätigkeit	Analysieren des Ist-Zustandes Definition des Soll-Zustandes Erstellung eines Pflichtenheftes
Zeitaufwand	7 Stunden

Tabelle 8: Definitionsphase

In einem Meeting mit den Auszubildenden und dem Ausbilder der EOS UPTRADE GmbH wurde die Ausgangssituation besprochen und Wünsche geäußert. Mit Hilfe der besprochenen Punkte wurde ein detailliertes Pflichtenheft erstellt. Das erstellte Pflichtenheft ist als Anhang A angefügt. Des Weiteren wurde ein Use Case Diagramm (Anhang B) erstellt, welche das Generieren eines Nachweises verdeutlichen soll.

Planungsphase

Tätigkeit	Gestaltung der GUI
Zeitaufwand	3 Stunden

Tabelle 9: Gestaltung der GUI

Aus den Anforderung und dem Use Case Diagramm wurde ein Mockup (Anhang C) erstellt, welches die Funktionen der GUI darstellen soll. Anschließend wurde anhand des Mockups ein grafisches Layout erstellt. Beim Erstellen des Layouts wurde sehr viel Wert darauf gelegt, dass sich das Layout nahtlos in das bestehende Intranet integrieren lässt.

Tätigkeit	Recherche Technologien zur PDF-Generierung
Zeitaufwand	3 Stunden

Tabelle 10: Recherche Technologien zur PDF-Generierung

Durch Erfahrungen in vorherigen Projekten wurde auf „wkhtmltopdf“ zurückgegriffen. Dies ist eine schlanke Library die auf vielen verschiedenen Plattformen installiert werden kann. Über einen schlanken Konsolen Befehl kann ein PDF aus einer URL generiert werden. <https://code.google.com/p/wkhtmltopdf/>

wkhtmltopdf www.google.de google.pdf

Codebeispiel 1: PDF-Generierung der Startseite von Google

Außerdem wurde ein Open-Source-Bundle „KnpsnappyBundle“ für das verwendete Framework Symfony 2 gefunden, welche die Funktionalitäten der Library in einem Service kapselt und das Generieren eines PDF aus der Response eines Controllers ermöglicht. Da das „KnpsnappyBundle“ auch die verwandte Library wkhtmltoimage unterstützt kann auch das Generieren eines Bildes ermöglicht werden ohne zusätzlichen Aufwand.

Realisierungsphase

Tätigkeit	Einrichten des Projekts
Zeitaufwand	1 Stunde

Tabelle 11: Einrichten des Projekts

Zum Start des Projekts wurde ein „git“ Repository auf der Open Source Plattform GitHub angelegt. Das Repository ist unter der URL <https://github.com/PalliDotCom/Berichtsheft> zu erreichen.

Das Repository wurde auf dem Entwicklungsrechner aus gecheckt und die neueste Symfony Version über Composer in die working copy kopiert. Alternativ wäre auch ein Download von Symfony von der Webseite möglich <http://www.symfony.com>.

Anschließend wurden die Symfony Core Dateien in einem initialem Commit in das Repository committed. Das Projekt befindet sich nun unter Versionskontrolle und kann global aus gecheckt und weiterentwickelt werden.

Tätigkeit	Einrichten der Entwicklungsumgebung
Zeitaufwand	1 Stunde

Tabelle 12: Einrichten der Entwicklungsumgebung

Auf dem lokalen Entwicklungsrechner läuft eine LAMP Umgebung. Es wird ein neuer vHost eingerichtet, welcher auf die working copy zeigt. Das Projekt ist nun lokal erreichbar.

Anschließend müssen die Abhängigkeiten des Projektes, mit dem Dependency Manager „Composer“, installiert werden.

Zum Schluss wird mit dem Entwicklungs-IDE „PHPStorm“ ein neues Projekt angelegt um den Code zu bearbeiten.

Tätigkeit	Umsetzung der GUI
Zeitaufwand	3 Stunden

Tabelle 13: Umsetzung der GUI

Benutzeroberfläche für die Login-Seite, das Dashboard und die Vorschau wurden mit HTML und CSS erstellt. Der HTML und CSS Code basieren auf dem Framework Bootstrap. Es wurde Wert darauf gelegt, dass sich das Layout nicht von den anderen Tools im Intranet unterscheidet.

Tätigkeit	Kommunikation zur bestehenden Zeiterfassungs-Software
Zeitaufwand	14 Stunden

Tabelle 14: Kommunikation zur bestehenden Zeiterfassungs-Software

Für die Kommunikation mit „Jira“ wurde das Open Source Bundle „MedicoreNL/JiraApiBundle“ verwendet. Dieses Bundle kapselt die Jira REST API Abrufe in Services. Der „jira_api.issue“ Service um Issues aus Jira abzurufen, der „jira_api.project“ Service um Informationen zu einem Projekt abzurufen und der „jira_api.search“ Service um Suchen mit JQL durchzuführen.

Da das Bundle nur die rudimentäre Kommunikation mit dem Hintergrundsystem ermöglicht, wurde ein Service erstellt, der die Abfrage an das Hintergrundsystem kapselt und die Daten in einer vorgegebenen Struktur zurückliefert (siehe Codebeispiel 1). Durch die Abstraktion kann die Software leicht erweitert werden um mit anderen Hintergrundsystemen zu kommunizieren. Für das Hintergrundsystem „Jira“ wurde eine konkrete Implementierung vorgenommen, da dieses Hintergrundsystem bei der EOS UPTRADE GmbH verwendet wird.

Für die Generierung eines Berichtsheftes wurde ebenfalls ein abstrakter Service erstellt, welcher einen WorklogRetriever übergeben bekommt (siehe Codebeispiel 2). Die konkrete Implementierung bekommt den WorklogRetriever für das Hintergrundsystem „Jira“ übergeben und generiert, aus den zurückgelieferten WorklogItems ein Berichtsheft.

Testphase

Tätigkeit	Erstellen / Einrichten von Tests
Zeitaufwand	5 Stunden

Tabelle 15: Erstellen / Einrichten von Tests

Es wurden Unit-Tests geschrieben die Überprüfen ob das entsprechende Hintergrundsystem erreichbar ist.

Tätigkeit	Tests durchführen
Zeitaufwand	2 Stunden

Tabelle 16: Tests durchführen

Die Applikation wurde auf einem vHost deployed und die Unit-Tests ausgeführt. Anschließend wurden GUI Tests in den Browsern Firefox, Chrome und Safari vorgenommen. Auch die Usability wurde in diesem Zuge getestet.

Tätigkeit	Auswertung der Ergebnisse
Zeitaufwand	1 Stunde

Tabelle 17: Auswertung der Ergebnisse

Im Zuge der GUI und Usability Tests wurden Änderungen an der Schriftgröße und Schriftfarbe vorgenommen . Außerdem wurden bei den Buttons Icons hinzugefügt um die Usability zu verbessern.

Dokumentation

Tätigkeit	Projektdokumentation erstellen
Zeitaufwand	10 Stunden

Tabelle 18: Projektdokumentation erstellen

Themen und Inhalte für die Dokumentationen wurden recherchiert und angefertigt. Des Weiteren wurde eine optische Darstellung an die CI der EOS UPTRADE GmbH erstellt und anschließend die Dokumentation in das PDF Format konvertiert.

Tätigkeit	Kurze Anwenderdokumentation erstellen
Zeitaufwand	2 Stunden

Tabelle 19: Kurze Anwenderdokumentation erstellen

Die Anwenderdokumentation wurde mit HTML erstellt und ist online verfügbar. Sie beinhaltet Beschreibungen für sämtliche Funktionen und Abläufe des BerichtsheftTools. Ein Screenshot der Anwenderdokumentation ist als Anhand D beigefügt.

Übergabe

Tätigkeit	Deployment
Zeitaufwand	1 Stunde

Tabelle 20: Deployment

Für das BerichtsheftTool wird ein neuer vHost angelegt und mit dem Tool Capistrano auf die entsprechende Instanz deployt.

Nach den erfolgreichen Tests wird das neue Modul über einen zusätzlichen Link im Intranet implementiert.

Tätigkeit	Einweisung betroffender Mitarbeiter
Zeitaufwand	1 Stunde

Tabelle 21: Einweisung betroffender Mitarbeiter

Nach erfolgreicher Implementierung in das Intranet der EOS UPTRADE GmbH wurde ein Meeting mit den Auszubildenden und dem Ausbilder der EOS UPTRADE GmbH angesetzt. In dem Meeting wurde das Tool

vorgestellt und Fragen geklärt. Außerdem wurde auf das Benutzerhandbuch verwiesen, welches online erreichbar ist.

Projektbewertung

Die Projektbewertung beschreibt, ob alle Zielvorgaben und Zeitplanungen eingehalten wurden und welche Erweiterungen für die Zukunft geplant sind.

Zielerreichung

Nach der erfolgreichen Einführung des neuen Tools erstellen alle Auszubildenden der EOS UPTRADE GmbH Ihre Berichtshefte mit dem BerichtsheftTool. Viele Mitarbeiter tun sich noch schwer damit, Ihre geleistete Arbeit angemessen mit einem Kommentar zu beschreiben, deswegen müssen öfters noch Inhalte bei der Generierung erweitert oder angepasst werden. Eine Bearbeitung der Einträge war in diesem Projekt jedoch nicht vorgesehen, aus diesem Grund gibt es öfters Probleme. Ich hoffe jedoch, dass sich die Mitarbeiter schnell an die neue Projektmanagement-Software „Jira“ gewöhnen und Ihre Arbeit mit einem aussagekräftigen Beschreibungstext versehen.

Im großen und ganzen sind die Auszubildenden zufrieden, da diese Ihre geleistete Arbeit nur noch an einer Stelle pflegen brauchen und das Berichtsheft in einem einheitlichen Format generiert wird. Aus diesem Grund kann ich behaupten, dass Projekt hat das Ziel erreicht.

Soll-/Ist-Abgleich

Der im Antrag vorgestellte Zeitplan von 70 Stunden wurde eingehalten. Insgesamt wurde mehr Zeit für die Recherche von Open Source Lösungen verwendet als geplant, dies machte sich aber in der Entwicklung deutlich bemerkbar, da viele Komponenten nicht mehr selbst entwickelt werden mussten.

In der nachfolgenden Tabelle werden Abweichungen im Soll-/Ist-Abgleich dargestellt.

Projektphase	Aufgabenbeschreibung	Soll-Stunden	Ist-Stunden
Definitionsphase	Ist-Analyse	2	2
	Soll-Analyse	2	2
	Pflichtenheft erstellen	3	3
Planungsphase	Gestaltung der GUI	3	3
	Recherche Technologien zur PDF-Generierung	3	6
Realisierungsphase	Einrichten des Projekts	1	1
	Einrichten der Entwicklungsumgebung	1	1
	Umsetzung der GUI	3	2
	Kommunikation zur bestehenden Zeiterfassungs-Software	14	10
	Generierung eines PDF's aus den Daten der Zeiterfassungs-Software	15	10
Testphase	Erstellen / Einrichten von Tests	5	6
	Tests durchführen	2	1
	Auswertung der Ergebnisse	1	1
Dokumentation	Projektdokumentation	10	15
	Kurze Anwenderdokumentation	2	2
	Ist-Soll-Vergleich	1	1
Übergabe	Deployment	1	1
	Einweisung betroffener Mitarbeiter	1	1
Summe		70	68

Tabelle 22: Soll-/Ist-Abgleich

Weiterentwicklung

Bei der Einführung des Projektes wurden weitere Wünsche geäußert. Es wurde gewünscht, dass man die persönlichen Daten selbstständig editieren kann und dafür nicht immer einen Administrator ansprechen muss. Des Weiteren wurde diskutiert, ob es sinnvoll ist das Editieren und speichern von Nachweisen zu ermöglichen, dies soll in einer kommenden Version umgesetzt werden.

Quellcodeauszüge

Hier werden Quellcodeauszüge dargestellt.

```
namespace Berichtsheft\BaseBundle\Worklog;

use Berichtsheft\BaseBundle\Model\AzubiInterface;

abstract class WorklogRetriever
{
    /**
     * @param AzubiInterface $user
     * @param \DateTime $from
     * @param \DateTime $to
     * @return WorklogItem[]
     */
    abstract function retrieve(AzubiInterface $user, \DateTime $from, \DateTime $to);
}
```

Codeauszug 1: Berichtsheft\BaseBundle\Worklog\WorklogRetriever

```
namespace Berichtsheft\BaseBundle\BerichtsheftBuilder;

use Berichtsheft\BaseBundle\Model\AzubiInterface;
use Berichtsheft\BaseBundle\Model\Berichtsheft;
use Berichtsheft\BaseBundle\Worklog\WorklogRetriever;

/**
 * Interface BerichtsheftBuilderInterface
 * @package Berichtsheft\BaseBundle\BerichtsheftBuilder
 */
interface BerichtsheftBuilderInterface
{
    /**
     * @param AzubiInterface $azubi
     * @param int $week
     * @param int $year
     * @param int $number
     * @return Berichtsheft
     */
    function generateBerichtsheft(AzubiInterface $azubi, $week, $year, $number = 1);

    /**
     * @return WorklogRetriever
     */
    function getWorklogRetriever();
}
```

Codeauszug 2: Berichtsheft\BaseBundle\BerichtsheftBuilder\BerichtsheftBuilderInterface

Glossar

Git

Git ist eine Software zur Versionsverwaltung ähnlich wie SVN.

GitHub

GitHub ist ein Hosting-Dienst für Projekte, welche unter einer Git Versionskontrolle stehen.

Repository

Ein Repository im Context mit „Git“ ist ein Root-Verzeichnis (ein Projekt), welches unter Versionskontrolle steht.

Jira

Eine weitverbreitete Projektmanagement-Software die zum Beispiel zur Fehlerverwaltung oder Aufgabenverwaltung eingesetzt wird. Jira wurde von dem Softwareunternehmen „Atlassian“ entwickelt.

<https://www.atlassian.com>

JSON

JSON (JavaScript Object Notation) ist ein schlankes Datenaustauschformat, das für Menschen einfach zu lesen und zu schreiben und für Maschinen einfach zu parsen und zu generieren ist. (Quelle: www.wikipedia.de)

OSX

OSX ist ein von Apple entwickeltes und auf Unix basiertes Betriebssystem, welches auf Computern von Apple zum Einsatz kommt.

PHPStorm

PHPStorm ist eine IDE für PHP basierende Projekte und wurde von JetBrains entwickelt.

<https://www.jetbrains.com>

LAMP

LAMP ist ein Akronym für den kombinierten Einsatz von Programmen auf Basis von Linux, um dynamische Webseiten zur Verfügung zu stellen. Dabei stehen die einzelnen Buchstaben des Akronyms für die verwendeten Komponenten:

- **L**inux
- **A**pache
- **M**ySQL
- **P**HP

Quelle: www.wikipedia.de

Composer

Composer ist ein Tool um Abhängigkeiten von PHP basierenden Projekten zu verwalten ähnlich wie „Maven“ in Java.

Bootstrap

Bootstrap ist eine freie Sammlung von Hilfsmitteln für die Gestaltung von Websites und Webanwendungen. Es enthält auf HTML und CSS basierende Gestaltungsvorlagen für Typografie, Formulare, Buttons, Tabellen, Grid-System, Navigations- und andere Oberflächengestaltungselemente sowie zusätzliche, optionale JavaScript-Erweiterungen. Es ist das populärste Projekt beim Open-Source-Hostingdienst GitHub und wird unter anderem von der NASA und MSNBC eingesetzt.

Quelle: www.wikipedia.de

Rest API

Representational State Transfer (mit dem Akronym REST, seltener auch ReST) bezeichnet ein Programmierparadigma für Webanwendungen. Es gibt keine explizite Norm, daher gehen die Vorstellungen, was REST ist, auseinander. Im Grunde bezeichnet REST die Idee, dass eine URL genau einen Seiteninhalt als Ergebnis einer serverseitigen Aktion (etwa das Anzeigen einer Trefferliste nach einer Suche) darstellt, wie es der Internetstandard HTTP für statische Inhalte (Permalinks) bereits vorsieht; ein Ziel, das für dynamisch erzeugte Seiten mitunter jedoch zusätzlichen Aufwand erfordert.

Quelle: www.wikipedia.de

JQL

JQL ist eine eigene Query Language von Jira mit der man in Jira erweiterte Suchen vornehmen kann.

Capistrano

Capistrano ist ein Open Source Werkzeug, um Skripte auf mehreren Servern laufen zu lassen. Sein Hauptzweck besteht in der Softwareverteilung von Webapplikationen. Es automatisiert den Prozess der Erstellung einer neuen Applikation, die auf einem oder mehreren Webservern verfügbar ist, einschließlich unterstützender Aufgaben wie Änderungen von Datenbanken.

Quelle: www.wikipedia.de

Symfony

Open Source PHP Framework welches unter anderem dem Model-View-Controller-Schema (MVC) folgt. Des Weiteren bietet Symfony einen Service Layer mit Dependency Injection.

Bundle

Für das Framework Symfony gibt es Open Source Bundles die vergleichbar sind mit Modulen oder Plugins aus anderen Frameworks.

Tabellenverzeichnis

Tabelle 1: Zeitplanung der Projektphasen.....	4
Tabelle 2: Einmalige Entwicklungskosten.....	5
Tabelle 3: Aufwand pro Woche und Auszubildenden.....	5
Tabelle 4: Kosten für die Erstellung der Berichtshefte im Jahr.....	5
Tabelle 5: Unterhaltungskosten.....	6
Tabelle 6: Kostenvergleich.....	6
Tabelle 7: Nutzwertanalyse.....	6
Tabelle 8: Definitionsphase.....	7
Tabelle 9: Gestaltung der GUI.....	7
Tabelle 10: Recherche Technologien zur PDF-Generierung.....	7
Tabelle 11: Einrichten des Projekts.....	7
Tabelle 12: Einrichten der Entwicklungsumgebung.....	8
Tabelle 13: Umsetzung der GUI.....	8
Tabelle 14: Kommunikation zur bestehenden Zeiterfassungs-Software.....	8
Tabelle 15: Erstellen / Einrichten von Tests.....	8
Tabelle 16: Tests durchführen.....	9
Tabelle 17: Auswertung der Ergebnisse.....	9
Tabelle 18: Projektdokumentation erstellen.....	9
Tabelle 19: Kurze Anwenderdokumentation erstellen.....	9
Tabelle 20: Deployment.....	9
Tabelle 21: Einweisung betroffener Mitarbeiter.....	9
Tabelle 22: Soll-/Ist-Abgleich.....	11

Anhänge

Anhang A	Pflichtenheft
Anhang B	Use Case Diagramm
Anhang C	Mockup
Anhang D	Anwenderdokumentation