Johannes Meyerhoff, Azubi-Nr.: 5002305256

Fachinformatiker fÜR Anwendungsentwicklung

GESIS Leibniz institut für sozialwissenschaften

UNTER SACHSENHAUSEN 6-8 50667 KÖLN

Erweiterung des Quartalsdaten-Erhebungs-Systems (QES) um die Funktionalität, erhobene Datensätze zu exportieren.

Inhaltsverzeichnis

1 Vorwort 1

2 Projektbeschreibung 2

2.1 Projektumfeld und technische Infrastruktur 2

2.2 Ausgangssituation 3

2.3 Zielsetzung 3

3 Projektüberblick 4

3.1 Vorgehensmodell XP 4

3.2 Zeitaufwand 6

3.3 Kosten 6

3.4 Sachmittel 6

4 Projektphasen 6

4.1 Planungsphase 6

4.1.1 Realisierungskonzept 6

4.2 vorbereitende Aufgaben 6

4.2.1 Testumgebung einrichten 6

4.3 Programmrealisierung 6

4.3.1 Erstellung Model 6

4.3.2 Erstellung Controller 6

4.3.3 Etc. 6

4.4 Qualitätssicherung 6

4.4.1 Unit-Tests 6

4.4.2 Feature-Tests 7

4.4.3 Integrationstest 7

4.4.4 Performancetest 7

4.4.5 Fehlerbeseitigung 7

4.5 Abnahme und Dokumentation 7

4.5.1 Erstellung der Anwenderdokumentation 7

4.5.2 Erstellung der Projektdokumentation 7

4.5.3 Übergabe und Abnahme 7

5 Probleme und Lösungen 7

5.1 Erstellung des Controllers 7

6 Ergebnisse und Fazit 7

7 Anhang 7

7.1 A1 Glossar 7

7.2 A2 Informationsquellenverzeichnis 7

7.3 A3 Abbildungsverzeichnis 8

7.4 A4 Tabellenverzeichnis 8

7.5 A5 Anlagenverzeichnis 8

# Vorwort

Diese Projektdokumentation beschreibt die Durchführung meines Projekts zur Erweiterung eines bestehenden Systems um die Funktionalität, Daten in ein spezifisches Excel-Format zu exportieren. Das Projekt habe ich im Rahmen der IHK Ausbildung zum Fachinformatiker für Anwendungsentwicklung bei meinem Ausbildungsbetrieb GESIS durchgeführt. Meine Haupttätigkeit als Auszubildender bei der GESIS ist die Entwicklung des QES (**Q**uartalsdaten-**E**rhebungs-**S**ystem).

GESIS - Leibniz-Institut für Sozialwissenschaften mit Sitz in Köln und Mannheim (Hauptsitz) ist die größte deutsche Infrastruktureinrichtung für die Sozialwissenschaften.

Das Institut ist in fünf wissenschaftliche Abteilungen, die mit ihrem forschungsbasierten Service- und Produktangebot den Forschungsprozess der empirischen Sozialforschung in seiner gesamten Breite abdecken, unterteilt. Die Abteilung WTS (Wissenstechnologien für Sozialwissenschaften), für die ich arbeite, unterstützt die sozialwissenschaftlichen Fachabteilungen durch die Entwicklung innovativer Wissenstechnologien zur Verbesserung der Integration und Nutzerfreundlichkeit der digitalen GESIS-Angebote.

In dieser Dokumentation werden Abkürzungen und Fachbegriffe genutzt, welche über das Grundwissen der Informatik hinausgehen. Sofern diese nicht im Kontext erläutert werden, werden sie bei der erstmaligen Verwendung durch eine orangene Farbgebung gekennzeichnet und im Glossar genauer erläutert.

# Projektbeschreibung

## Projektumfeld und technische Infrastruktur

Die Entwicklung erfolgt am Sitz der GESIS in Köln. Der virtuelle Server für die Produktiv-Version von QES ist auf das interne Firmennetz beschränkt. Es ist also außerhalb der Firma nur. Über einen VPN erreichbar. Ich als Auszubildender, mein Ausbilder und die IT-Abteilung haben administrative Rechte. Die MySQL-Datenbank für QES ist nur von dem Server, auf dem die Anwendung läuft, erreichbar. Die Anwendung QES läuft auf einem virtualisierten Ubuntu 16 System. QES ist mit dem Laravel PHP Framework nach dem Model-View-Controller Prinzip aufgebaut und verwendet Laravel in der Version 5.4 und PHP mit Version 7.2.14.

Auf meinem GESIS-Entwicklungs-Computer ist auch eine Testumgebung mit gleichen Versionen vorhanden. Diese wird jedoch nicht mit VMware, sondern mit Oracle VirtualBox virtualisiert.

Die Entwicklungsschritte werden im Firmeninternen Git versioniert. Für einige Funktionen sind bereits automatisierte Tests vorhanden. Hierzu wird die in Laravel 5.4 mitgelieferte Version 5.7.27 von PHPUnit verwendet.

Werkzeuge zur Entwicklung sind Atom Editor, iTerm2, GitLab, phpMyAdmin, Mozilla Firefox und Google Chrome.

Auftraggeber für das Projekte ist der ehemalige kommissarische Abteilungsleiter der Abteilung WTS. Ansprechpartner bei technischen oder organisatorischen Problemen ist mein Ausbilder Sascha Schüller. Fachlicher Ansprechpartner bei geschäftsprozessbezogenen Rückfragen ist Thomas Knecht.

## Ausgangssituation

Die Leitung der Abteilung WTS erfasst in einem auf Excel-Tabellen basierenden Verfahren die Auslastung der Mitarbeiter im Bezug auf Projekte. Das neue System QES kann erst mit der in diesem Projekt entwickelten Export-Funktionalität vollstängig in den Arbeitsablauf integriert werden. Erst mit dieser Funktion kann damit das alte Verfahren abgelöst werden.

## Zielsetzung

Ziel des Projektes ist es, der Abteilungsleitung und auch den Teamleitern eine tabellarische Excel-Übersicht über den tatsächlichen Ressourcenverbrauch pro Mitarbeiter und Projekt sowie Abweichungen von der Soll-Planung zu liefern. Die Übersicht ist eine entscheidende Grundlage für die weitere Ressourcenplanung und soll dynamisch nach Auswahl eines Zeitraums aus den Daten in QES generiert werden.

Im QES soll für die Teamleiter und den Abteilungsleiter ein neuer Menüpunkt ergänzt werden. Der Menüpunkt Export soll auf eine Seite führen, auf der der Export definiert und angefordert wird. Der Ablauf ist wie folgt geplant: Der Nutzer wählt einen Zeitraum und anhand dieser Auswahl wird eine Datei erstellt. Dabei erhalten Teamleiter und Abteilungsleiter gemäß ihrer Rolle unterschiedliche Ergebnisse. Die Ergebnis-Dateien stellen den Ressourcenverbrauch pro Mitarbeiter und Projekt dar. Je Mitarbeiter wird je Projekt eine eigene Zeile mit dem Soll- und dem Ist-Wert angelegt. Autorisierte Nutzer können dann auf die Export-Dateien für eine bestimmte vorab festgelegte Zeit zugreifen. Verarbeitung, Konvertierung und Export der Daten auf dem Server müssen konzipiert, implementiert und getestet werden. Leitbild für Entwurf und Umsetzung ist das Model View Controller Konzept.

# Projektüberblick

## Vorgehensmodell XP

Als Vorgehensmodell habe ich Extreme Programming nach Kent Beck (kurz XP, siehe auch [XPWIK]) gewählt, da es beispielsweise im Vergleich zu Scrum weniger Rollen und Meetings benötigt. Extreme Programming hat außerdem einen iterativen Ansatz, der sehr schnell erste Ergebnisse produziert. Extreme Programming ist ein agiles Vorgehensmodell, welches ursprünglich dazu entwickelt wurde, die Kosten von Änderungen im laufenden Entwicklungsprozess gering zu halten. Extreme Programming verwendet das Konzept des Test Driven Developments (TDD). Das bedeutet, dass zur Entwicklung von Funktionen oder Funktionalitäten Tests geschrieben werden, welche bei korrekter Programmierung der Funktion erfolgreich durchlaufen werden. Die Tests werden für Laravel meist in PHPunit geschrieben. Auf den Pair-Programming Ansatz des Extreme Programme muss ich jedoch verzichten, da ich alleine programmiere.

Die in Abbildung 1 aufgezeigten Phasen des Vorgehensmodells können je nach Bedarf und Kompetenzen angepasst werden.



Abbildung 1 Phasen des Vorgehensmodells Extreme Programming

## Zeitaufwand

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Phase | Arbeitspaket | Name | Soll in Stunden | Ist in Stunden |
| Projektinitialisierung | AP PI1 | Kick-off Meeting mit dem Kunden durchführen | 0,5 | 0,5 |
|  | AP PI2 | Projektinitiierung/ -definition abschließen | 0,5 | 0,5 |
| Projektplanung | AP P1 | Risiken einschätzen und Kosten betrachten | 1,5 | 1 |
|  | AP P2 | Benötigte Funktionalitäten für Controller ermitteln | 0,5 | 1 |
|  | AP P3 | Verfügbare Komponenten und Lösungen am Markt untersuchen | 2 | 2 |
|  | AP P4 | Arbeitspakete planen | 2 | 2 |
|  | AP P5 | Projektphasen, Tätigkeiten, Termine, Meilensteine, Zeiten,Projektablauf , Ressourcen, Kosten, Nutzen planen | 3 | 4 |
|  | AP P6 | Tests anlegen um Qualität zu planen | 2 | 2 |
| Projektdurchführung  Analyse | AP A1 | Machbarkeitsstudie durchführen | 2 | 2 |
| Entwurf | AP E1 | Aus dem Lastenheft das Pflichtenheft erstellen | 2 | 2 |
| Entwurf | AP E2 | Eigenschaften des Models festlegen | 1 | 1 |
| Entwurf | AP E3 | Formular mit Auswahlmöglichkeit zur Generierung entwerfen | 4 | 4 |
| Entwurf | AP E4 | Menüpunkt und Controller entwerfen | 4 | 4 |
| Entwurf | AP E5 | Download von Excel-Fähiger Datei konzipieren | 1 | 1 |
| Entwurf | AP E6 | Layout mit Kunde absprechen | 1 | 1 |
| Implementierung | AP I1 | Route zum Formular und zum Download anlegen | 1 | 1 |
| Implementierung | AP I2 | View-Oberfläche entwickeln und mit Kunde abstimmen | 2 | 2 |
| Implementierung | AP I3 | Mapping implementieren | 1 | 1 |
| Implementierung | AP I4 | Controller erstellen | 3 | 4 |
| Implementierung | AP I5 | View anlegen | 3 | 3 |
| Implementierung | AP I6 | Funktion implementieren, die die angefertigte Datei als Download bereitstellt | 5 | 2 |
| Implementierung | AP I7 | Qualitätssicherung durch automatisierte Tests | 4 | 4 |
| Testphase mit Nutzertests | AP T1 | Funktionalität | 2 | 2 |
|  | AP T2 | Sicherheit | 2 | 2 |
|  | AP T3 | Integrität | 2 | 2 |
| Dokumentation |  | Dokumentation erstellen | 2 | 2 |
| Projektabschluss |  | Abnahme | 2 | 2 |

Tabelle 1: Aufwand mit Soll/Ist in Stunden

## Kosten

Kosten des Projekts setzen sich aus den Personalkosten und den Kosten für Sachmittel zusammen. Der PC für die Entwicklung ist jedoch bereits in den Personalkosten enthalten. Der interne Verrechnungssatz für einen Entwickler ist bei GESIS 60€. Die Kostenaufstellung kann Tabelle 2 entnommen werden.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Name | Beschreibung | Kosten pro Stunde | Stunden | Gesamtkosten |
| Johannes Meyerhoff | Projektleiter  Entwickler | 60€ | !!!!!!!! | !!!!\*60 |
| Sascha Schüller | Ansprechpartner | 60€ | 4 | 240€ |
| **Summe** |  |  | **4+!!!!** | **240+!!!\*60** |

Tabelle 2: Kostenaufstellung

## Sachmittel

In Tabelle 3 sind die zur Umsetzung des Projektes verwendeten Software und Hardwarekomponenten dargestellt.

|  |  |  |
| --- | --- | --- |
| Name | Art | Menge |
| BenQ BL2411 | Bildschirm | 2 |
| Dell Optiplex 7040 | Computer | 1 |
| Lenovo 54Y9414 | Tastatur | 1 |
| Logitech MX Master 2 | Maus | 1 |
| Windows 10 Enterprise | Betriebssystem |  |
| MS VS Code | Entwicklungsumgebung | 1 |

Tabelle 3: Sachmittel

# Projektphasen

## Planungsphase

### Realisierungskonzept

Mithilfe der in GitHub vorhandenen Projektansicht habe ich ein Kanban Board erstellt. In meinem Kanban Board sind all meine To-Dos. Sie sind unterteilt in drei vertikal getrennte Sektionen. Links ist der „Pool“ der möglichen Aufgaben, in der Mitte sind die Issues, welche aktuell in Bearbeitung sind und auf der rechten Seite sind die fertigen Aufgaben abgelegt.

Jedes Arbeitspaket ist als Issue hinterlegt und einem Meilenstein zugeordnet. Jedes Issue hat einen Meilenstein, aber ein Meilenstein kann mehrere Issues haben. Abbildung 2 zeigt einen beispielhaften Zwischenstand.

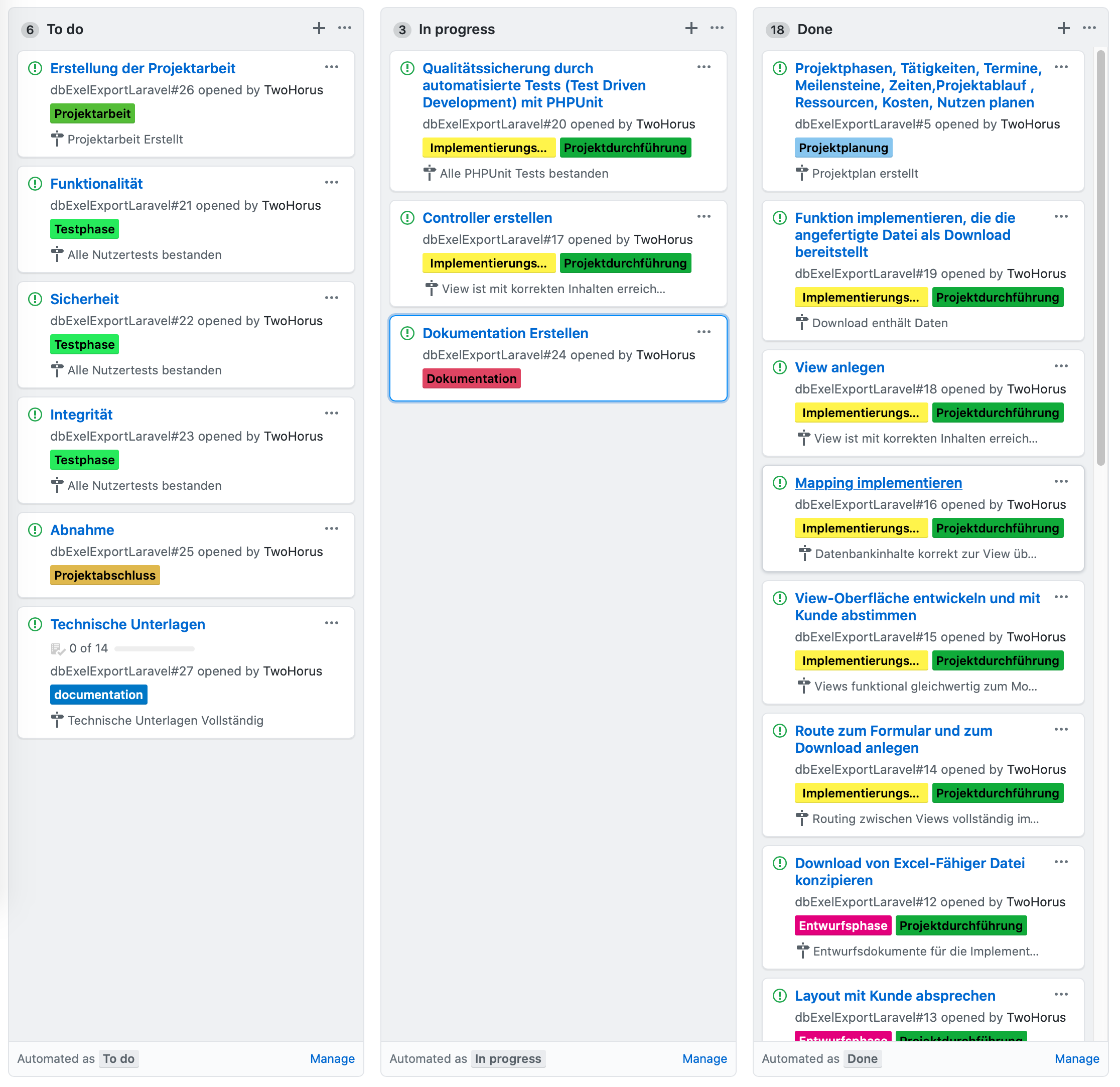


Abbildung 2: Kanban Board zur Planung

## vorbereitende Aufgaben

### Testumgebung einrichten

Ich habe mit Bash einige Shell-Scripte zum Starten meiner Tests erstellt. Wenn ich nun ein Haupt-Skript ausführe, wird die Test-Datenbank neu erstellt und ihre Integrität mithilfe der im Projekt hinterlegten Modelle geprüft. Das bedeutet, es wird getestet, ob der Mitarbeiter, der sich hinter „isworkerid“ verbirgt, auch einen Eintrag mit der ID in der Tabelle „worker“ hat. Ist dies nicht der fall, so wird auch direkt im Terminal, in dem das Skript läuft, eine Fehlermeldung angezeigt. Nach vollständiger Migration öffnet sich eine Entwickleransicht im Browser. Hier können für die Entwicklung nützliche Informationen und Formulare hinterlegt werden.

## Programmrealisierung

Das Model-View-Controller-Konzept (kurz MVC, siehe auch [MVCWIK]) wurde 1979 zunächst für Benutzeroberflächen in der Programmiersprach [Smalltalk](https://de.wikipedia.org/wiki/Smalltalk_(Programmiersprache)) durch [Trygve Reenskaug](https://de.wikipedia.org/wiki/Trygve_Reenskaug" \o "Trygve Reenskaug) beschrieben. Reenskaug arbeitete an der Weiterentwicklung von Smalltalk. MVC gilt mittlerweile als De-facto-Standard für den Grobentwurf vieler komplexer Softwaresysteme. Für mein Projekt war dieses Konzept ein wichtiger Ansatz, um schnell zu einer sinnvollen Architektur zu kommen.

### Erstellung Model

Ich habe mich für die Verwendung der Bestehenden Marktlösung von „Maatwebsite/Laravel-Excel“ zum Export nach Excel entschlossen, da dies die einzige gut dokumentierte Exportbibliothek für laravel ist. Es gibt eine weitere Bibliothek, „rap2hpoutre/fast-excel“. Dies ist eine schlankere Version von „Maatwebsite/Laravel-Excel“. In der Dokumentation von fast-excel wird zur nutzung des Larave-Excel pakets geraten, wenn man mehr als die Grundfunktionen verwenden möchte.

Eine Alternative zur Nutzung einer Bibliothek wäre es gewesen, eine eigene Bibliothek zu programmieren. Da ich dann jedoch php-excel als Basis genutzt hätte, kann ich auch die fertige Bibliothek verwenden, ohne selber den Entwicklungsaufwand erneut zu betreiben (Maatwebsite/Laravel-Excel nutzt php-excel als basis).

### Erstellung Controller

Der Controller nimmt in Laravel die Anfrage vom Routing entgegen. Hierbei kann man die Parameter einer Anfrage mit Laravel validieren und mit Parametern angeben, ob der Parameter ignoriert wird, sofern er nicht das richtige Format hat, oder man die Anfrage zurückweist. Letzteres nennt Laravel „bail“. Der Controller nimmt einen Aufruf des Nutzers als erste Anfrage entgegen, um dem Nutzer eine Liste an möglichen Optionen in einem Formular aufzuzeigen. Die Optionen sind in einer Select-Box aufgelistet. Somit ist eine fehlerhafte Eingabe seitens des Nutzers nur durch Manipulation des HTML-Formulars möglich. Nachdem der Nutzer einen Zeitraum für den Export gewählt hat, werden ihm die Informationen als Download bereitgestellt. Den Download habe ich mithilfe der „FromView“-Eigenschaft über die Bibliothek zum Export realisiert. Hierfür musste ich eine Datenbankabfrage formulieren, welche die benötigten Daten aus der Datenbank in einem Objekt ablegt, welches ich an die View weitergebe. Die View muss dann so über das Objekt iterieren, dass die Daten im richtigen Format in einer HTML-Tabelle landen, welche Vom Parser nach Excel formatiert werden kann.

### Erstellung View

Die View habe ich so erstellt, dass sie eine einzige große Tabelle anzeigt. Hierbei schaue ich in jeder Iteration wie sich das aktuelle Element verändert hat, um festzustellen, ob der Datensatz zu einem neuen Mitarbeiter gehört oder nicht. Wenn dieser zum gleichen Mitarbeiter gehört prüft die View, ob der Datensatz auch zum gleichen Projekt gehört. Je nach Ergebnis der Prüfungen wird eine neue Zeile ausgegeben und Je nach Ergebnis wird der Name des Mitarbeiters am Anfang der Zeile ausgegeben, oder eine Einrückung dort eingefügt, wo dieser sonst stehen würde. Die Ausgabe der Zeilen habe ich in eine Funktion „printCurrentRow“ ausgelagert. Die Einrückung habe ich in eine Funktion „indent“ ausgelagert. „Indent“ wird mit einem Parameter aufgerufen, welcher angibt, wie viele leere Zellen ausgegeben werden sollen. So wird bei jedem neuen Mitarbeiter sein Name, seine Abteilung und andere mitarbeiterspezifische Informationen ausgegeben. Bei jedem weiteren Projekt für denselben Mitarbeiter bleibt der Platz für diese Informationen frei und es werden nur die Soll und Ist Datensätze in den zugehörigen Spalten für die Zeiträume ausgegeben.

## Qualitätssicherung

### Unit-Tests

Einheiten

### Feature-Tests

Funktion aus Nutzer-sicht

### Integrationstest

Schnittstellen testen usw

### Performancetest

GESCHWINDIGKEIT???

Nutzersicht kontext

### Fehlerbeseitigung

TEST FIXEN

## Abnahme und Dokumentation

### Erstellung der Anwenderdokumentation

Für auftraggeber

### Erstellung der Projektdokumentation

Für ihk

### Übergabe und Abnahme

Wann wo

# Probleme und Lösungen

## Erstellung des Controllers

Problem und lösung hier

# Ergebnisse und Fazit

Was war gut schlecht änderungen

# Anhang

## A1 Glossar

## A2 Informationsquellenverzeichnis

[MVCWIK] https://en.wikipedia.org/wiki/[Model\_View\_Controller](https://de.wikipedia.org/wiki/Model_View_Controller)

[XPWIK] https://en.wikipedia.org/wiki/Extreme\_programming

## A3 Abbildungsverzeichnis

## A4 Tabellenverzeichnis

## A5 Anlagenverzeichnis