Für internen Gebrauch

Yatzy

autor(en) : Bosshard Fabrice

version : 1.0

status : Final

quelle : Atos

dokumentendatum : 21 November 2018­

­

­

anzahl der seiten : 4

owner : Fabrice-Ronny Bosshard

**Änderungshistorie**

| Version | Datum | Beschreibung | Autor(en) |
| --- | --- | --- | --- |
| 0.1 | 21.11.2018 | Initiale Dokumentationsstruktur erstellt | Fabrice Bosshard |
| 0.2 | 22.11.2018 | Informieren abgeschlossen | Fabrice Bosshard |
| 0.3 | 22.11.2018 | Planen abgeschlossen | Fabrice Bosshard |
| 0.4 | 22.11.2018 | Entscheiden abgeschlossen | Fabrice Bosshard |
| 0.5 | 27.11.2018 | Realisieren abgeschlossen | Fabrice Bosshard |
| 0.6 | 27.11.2018 | Kontrollieren abgeschlossen | Fabrice Bosshard |
| 0.7 | 27.11.2018 | Auswerten abgeschlossen | Fabrice Bosshard |
| 0.8 | 28.11.2018 | Finale Version abgeschlossen | Fabrice Bosshard |

Tabelle 1: Änderungshistorie

© Copyright 2018, Atos AG Alle Rechte vorbehalten. Reproduktion von Teilen oder dem Gesamten ist ohne vorherige schriftliche Genehmigung des Urhebers untersagt. Für Fragen oder Anmerkungen zu diesem Dokument wenden Sie sich bitte an Atos, 078 620 00 35.

­­­Inhalt

**Teil 1: Umfang und Ablauf**

1 Aufgabenstellung 4

1.1 Projekt 4

1.2 Ausgangslage 4

1.3 Detaillierte Aufgabenstellung 4

1.4 Management Summary 4

1.5 Technologie-Stack 4

1.6 Vorkenntnisse 4

1.7 Vorarbeiten 4

2 Projektorganisation 5

2.1 Beteiligte Personen 5

2.2 Projektmanagementmethode 5

2.3 Backup-Konzept 5

3 Zeitplanung 6

3.1 Meilensteine 6

3.2 Massnahmen bei Verzug 6

4 Gantt-Diagramm 7

5 Arbeitsprotokoll 8

5.1 Mittwoch 21. November 2018 (3h) 8

5.2 Donnerstag 22. November 2018 9

5.3 Freitag 23. November 2018 10

5.4 Dienstag 27. November 2018 (3h) 11

6 Kurzfassung 12

6.1 Ausgangslage 12

6.2 Umsetzung 12

6.3 Ergebnis 12

7 Informieren 13

8 Planen 14

9 Entscheiden 15

10 Realisieren 16

11 Kontrollieren 17

12 Auswerten 18

13 Glossar 19

14 Verzeichnisse 20

14.1 Quellenverzeichnis 20

14.2 Abbildungsverzeichnis 20

14.3 Tabellenverzeichnis 21

15 Aufgabenstellung 22

16 Programmcode 23

# Aufgabenstellung

Die Aufgabenstellung wird vom Auftraggeber festgelegt und ist anbei dokumentiert.

## Projekt

Projekttitel: Yatzy

Im Auftrag von: Torben Dziuk

Auftragnehmer: Fabrice-Ronny Bosshard

Starttermin: 21. November 2018 14:00

Abgabetermin: 28. November 2018 13:00

Geplanter Projektaufwand: Drei Arbeitstage à 8 Stunden (24h+ 2h)

Erstellung einer simplen Applikation, mit Frontend, Backend und Anbindung an eine Datenbank.

## Ausgangslage

Damit die Lehrlinge in der Atos genügend auf die Individuelle Praktische Arbeit (IPA) vorbereitet sind, leitet Torben Dziuk jeden Mittwoch einen Kurs im kleinen Rahmen. Nun geht es darum eine Probe-IPA zu schreiben, damit wir aus unseren Fehlern lernen können und uns stetig verbessern.

Als Aufgabenstellung wurde die Aufgabe aus dem Informatik-Schweizermeisterschaften des Jahres 2017 genommen.

Das Projekt wird von Grund auf aufgebaut und implementiert. Der Auftragnehmer muss neben den Standard .NET-Kenntnissen auch solide Kenntnisse in Entity Framework und WPF (Windows Presentation Foundation) haben.

Die Struktur der Dokumentation soll bereits als Vorlage existieren, damit bei Projektbeginn direkt mit dem Schreiben angefangen werden kann.

## Detaillierte Aufgabenstellung

Im Anhang finden sie die detaillierte Aufgabenstellung der ICT-Skills Schweiz. Sie beschreibt die funktionalen Anforderungen an die Applikation und gibt durch Wire Frames vor, wie das Layout der Benutzeroberfläche aussehen muss.

## Management Summary

Kurze Repräsentation der Aufgabenstellung von ICT-Skills Schweiz.

Yatzy ist ein Würfelspiel für Klein bis Gross. Das Ziel dieses Spieles ist, die meisten Punkte mit den verschiedenen vorgegebenen Kombinationen zu sammeln. Sobald alle Kombinationen abgedeckt sind, gewinnt der Spieler mit der höchsten Punktzahl.

## ****Technologie-Stack****

IDE: Visual Studio 2015 (mit ReSharper)

Database: MSSQL 2014 mit Management Studio

Front- / Backend: C# 5.0 mit Windows Presentation Foundation (WPF)

Versionsverwaltung: Git mit TortoiseGit

Programmarchitektur wird nach MVVM (Model View ViewModel) umgesetzt.

## Vorkenntnisse

In diesem Abschnitt werden die bereits erlangten Kenntnisse aufgelistet.

Projektkenntnisse:

* C# (2 Jahre)
* WPF (2 Jahre)
* Visual Studio (2 Jahre)
* Angular *TypeScript* (1 Jahr)
* Git (2 Jahre)
* CI (1 Jahr)
* DI (1 Jahr)

## Vorarbeiten

In der IPA muss eine vorgegebene Arbeit umgesetzt und dokumentiert werden. Oftmals sprengen alle Tätigkeiten in dieser Arbeit den Zeitrahmen und deshalb werden gewisse Arbeiten bereits zuvor erledigt. In diesem Kapitel werden alle Vorbereitungen für die Probe-IPA deklariert.

* Initiale Dokumentationsstruktur aufbauen nach IPERKA.
  + Struktur
  + Layout
  + Verzeichnisse

# Projektorganisation

Dieses Kapitel gibt einen Überblick über den Aufbau der Organisation, informiert über die beteiligten Personen und zeigt die Vorgehensweise im Allgemeinen.

## Beteiligte Personen

|  |  |  |  |
| --- | --- | --- | --- |
| **Person** | **Rolle** | **Verantwortung** | **Kontakt** |
| Torben Dziuk | Auftraggeber & Verantwortliche Fachkraft | Aufgabe definieren & Unterstützung Kandidat | torben.dziuk@atos.net |
| Fabrice Bosshard | Kandidat | Umsetzung der Arbeit | fabrice.bosshard.external@atos.net  fabrice.bosshard@siemens.com |

Tabelle 2: Beteiligte Personen

## Projektmanagementmethode

Im Betrieb arbeiten wir mit der Projektmanagementmethode Scrum, welche sich sehr gut für die agile Prozessentwicklung eignet. Bei Scrum werden die verschiedenen Arbeitsschritte in Zyklen (Sprints) abgehandelt, was das Anpassen von Vorgaben zwischen den Zyklen und die transparente Kommunikation innerhalb des Entwicklerteams ermöglicht.

Scrum ist jedoch für eine IPA nicht gut geeignet, da der Mehraufwand für die Formalitäten (Daily, Retro etc.) zu hoch wäre.

Ich habe mich deshalb für die Projektmanagementmethode IPERKA (**I**nformieren, **P**lanen, **E**ntscheiden, **R**ealisieren, **K**ontrollieren, **A**uswerten) entschieden, da ich damit schon viel Erfahrungen in der Schule sammeln konnte. Diese klassische Projektmanagementmethode eignet sich für die IPA perfekt, da der Ablauf respektive die Struktur mit den Anforderungen der Arbeit übereinstimmt.

## Backup-Konzept

Bei der Umsetzung der Arbeit ist es essentiell, dass man stets auf eine valide Version der Arbeit zurückgreifen kann. Wenn man zum Beispiel einen groben Fehler macht und die ganze Applikation nicht mehr zum Laufen kriegt, ist es wichtig, dass man nicht von Neuem beginnen muss, sondern auf den Stand vor dem Fehler zurückgreifen kann.

Die Dokumentation und der Quellcode werden in ein Git-Repository geladen und sind somit zentral in einer Cloud gespeichert. Dazu wird das externe Tool GitHub verwendet. GitHub macht bei sich selber immer wieder Backups und so muss ich mich nicht um ein Backup-System kümmern. Das Repository muss stets aktualisiert werden, damit möglichst immer der neuste Stand verfügbar ist. Um dies zu ermöglichen, sollte für jede neue Funktionalität ein Commit erstellt werden, welcher den neuen Stand darstellt (Daumenregel: 2-4 Mal am Tag).

Als weitere Absicherung wird ein USB-Stick genutzt, auf welchem mindestens ein Mal am Tag ein komplettes Backup erstellt wird. Im Ernstfall würde man als erstes versuchen auf das Git-Repository zurückzugreifen, doch wenn dies ebenfalls fehlschlagen sollte, gibt es noch eine Sicherung auf dem Stick.

# Zeitplanung

In diesem Kapitel werden die verschiedenen Aufwände geschätzt und aufgelistet. Die Zeitplanung dient als Leitfaden für den Projektablauf und hilft bei der Selbstkontrolle.

## Meilensteine

Meilensteine sind wichtige Teilpunkte im Projektverlauf. Sie werden als Prüfpunkte verwendet und wirken sich positiv auf die Qualitätssicherung aus. Man kann somit die erledigten Ergebnisse mit den erwarteten Vorgaben abgleichen.

|  |  |  |
| --- | --- | --- |
| Meilenstein | Erfüllungskriterien | Datum |
| Projektbeginn | Kick-off der Probe IPA | 21.11.2018 14:00 Uhr |
| Informieren | Auftragsanalyse durchgeführt und dokumentiert | 22.11.2018 10:00 Uhr |
| Planen | Anwendungsfälle, Architektur, Testfälle planen | 22.11.2018 15:00 Uhr |
| Entscheidung | Tools / Framework und Architektur-Entscheidungen definieren | 22.11.2018 17:30 Uhr |
| Realisieren | Implementation der Arbeit & Arbeit dokumentiert | 27.11.2018 15:00 Uhr |
| Kontrollieren | Die zuvor definierten Kontrollkriterien überprüft und Fazit daraus gezogen | 27.11.2018 17:30 Uhr |
| Auswerten | Reflexion der Arbeit und des Arbeitsverhalten erstellt & Kurzfassung geschrieben | 28.11.2018 11:00 Uhr |
| Projektabschluss | Projektarbeit abgeschlossen | 28.11.2018 13:00 Uhr |

Tabelle 3: Meilensteine

## Massnahmen um Verzug zu verhindern

Ein Zeitplan oder Ähnliches ist ein Leitfaden den es unbedingt braucht. Jedoch wird es in praktisch keinem Projekt gelingen, sich stets an den Plan zu halten. Man muss vorbereitet sein auf gewisse Abweichung, denn diese passieren überall. Deshalb wird in dieser Arbeit für jeden Tag die letzte Stunde als „offen“ geplant. Das heisst ich habe immer eine Stunde pro Tag, welche für nichts eingeplant wurde und welche ich nutzen kann, falls ich mit einer geplanten Arbeit etwas länger brauche als geplant.

## Massnahmen bei Verzug

Sobald ein Meilenstein oder eine Tätigkeit nicht zur definierten Zeit erledigt werden konnte, muss eine Massnahme ergriffen werden, damit weitere wichtige Tätigkeiten nicht in Gefahr geraten. Dazu würden die kleineren Teil-Tätigkeiten, welche nicht grösseren Einfluss auf den Fertigstellungsgrad haben, ignoriert und als optional für nach der Fertigstellung markiert werden.

Es werden jedoch nur Teil-Tätigkeiten aus der Realisierungs-Phase weggenommen, damit der allgemeine Projektablauf nicht manipuliert wird. Das heisst, wenn eine Massnahme ergriffen werden müsste, würde als Erstes die letzte Funktionalität laut Aufgabenstellung (siehe Anhang) rausfallen.

# Gantt-Diagramm



Abbildung 1: Gantt-Diagramm

# Arbeitsprotokoll

Das Arbeitsprotokoll dient zur Repräsentation des Tagesablaufs und der Fortschrittsbelegung. An jedem Arbeitstag wird das Arbeitsprotokoll erweitert und gepflegt. Es wird für jeden Tag mindestens eine Stunde für das Protokoll eingerechnet, damit die Qualität des Produkts sichergestellt werden kann. Es zeigt die geplanten Aufgaben pro Tag und stellt dar, ob es von der Tagesplanung Abweichungen gegeben hat. Zudem enthält das Protokoll jeweils eine Reflexion für jeden Tag, bei der man auf Erfolge, Misserfolge und den daraus zu schliessenden Massnahmen achtet.

## Mittwoch 21. November 2018 (3h)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (21.11.2018) | | | | |
| Phase: Informieren | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
| Dokumentationsstruktur | 0.5 | 0.5 |  |
| Projektorganisation definieren | 0.5 | 0.5 |  |
| Meilensteine definieren + Massnahmen | 0.5 | 0.5 |  |
| Gantt-Diagramm | 1 | 1 |  |
| Arbeitsjournal | 0.5 | 0.5 |  |
| Total | 3 Stunden | 3 Stunden |  |

Tabelle 4: Arbeitsprotokoll 21.11.2018

Heute habe ich mit dem Projekt gestartet. Als Erstes hatte ich die Aufgabe mit Torben Dziuk besprochen und wir haben kleinere Verständnisfragen zum Inhalt geklärt. Danach machte ich mich gleich daran die Dokumentationsstruktur aufzubauen, welche ich teilweise aus der bereits erstellen Probe-IPA aus dem Modul 226 übernommen habe. Den Rest der Zeit habe ich noch kleinere Tätigkeiten erledigt, welche bereits zur Phase „Informieren“ gehören.

### Erfolge

Ich konnte die Struktur des Dokuments fast 1 zu 1 von der Probe-IPA übernehmen. Ich merkte heute den enormen Mehrwert, einer guten Vorbereitung. Aus diesem Grund geling es mir recht schnell die geplanten Tätigkeiten abzuhandeln, obwohl ich relativ müde und deshalb leistungsineffizient war.

### Misserfolge

Ich finde es etwas Schade, dass wir Mitten im Tag mit dem Projekt gestartet haben. Die Anfangsphase (Informieren) ist sehr wichtig und da sollte man mit dem Kopf 100% dabei sein. Gegen Ende des Tages ist man jedoch meistens sehr angeschlagen und es fiel mir deshalb ein bisschen schwer die Zeitplanung zu erstellen. Ausserdem musste ich, sobald ich mal in der Arbeit drin war, direkt aufhören und die Arbeit am nächsten Tag weiterführen. Dies stört den Arbeitsfluss und verschlechtert meine Effizienz.

### Fazit

Ich konnte alle geplanten Tätigkeiten erledigen und bin vollständig bereit für den nächsten Tag. Durch die bereits durchgelaufene Probe-IPA habe ich viele Kenntnisse erworben, welche mir helfen effizienter und qualitativ besser zu arbeiten. Diese konnte ich schon heute einsetzen.

## Donnerstag 22. November 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (22.11.2018) | | | | |
| Phase: Informieren & Planen | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total |  |  |  |

Tabelle 5: Arbeitsprotokoll 22.11.2018

### Erfolge

### Misserfolge

### Massnahmen

### Fazit

## Freitag 23. November 2018

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (23.11.2018) | | | | |
| Phase: Informieren & Planen | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total |  |  |  |

Tabelle 6: Arbeitsprotokoll 23.11.2018

### Erfolge

### Misserfolge

### Massnahmen

### Fazit

## Dienstag 27. November 2018 (3h)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (27.11.2018) | | | | |
| Phase: Informieren & Planen | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total |  |  |  |

Tabelle 7: Arbeitsprotokoll 27.11.2018

### Erfolge

### Misserfolge

### Massnahmen

### Fazit

## Mittwoch 28. November 2018 (3h)

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Tag 1 (28.11.2018) | | | | |
| Phase: Informieren & Planen | | | | |
| Tätigkeit | **Soll-Zeit (Stunden)** | **Ist-Zeit (Stunden)** | **Bemerkung** |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Total |  |  |  |

Tabelle 7: Arbeitsprotokoll 28.11.2018

### Erfolge

### Misserfolge

### Massnahmen

### Fazit

# Kurzfassung

In der Kurzfassung wird kurz und bündig das Kapitel «Teil 2: Projekt» zusammengefasst. Es zeigt die wichtigsten Stichpunkte aus diesem Teil und gibt dem Leser einen raschen Überblick der Arbeit.

## Ausgangslage

## Umsetzung

## Ergebnis

# Informieren

Das Informieren ist die erste Stufe in der gewählten Projektmethode «IPERKA». In diesem Schritt geht es vor allem um die Auftragsanalyse. Hier werden benötigte Information herausgesucht und zusammengestellt. Dabei werden Kriterien wie Umgebung, Bedingungen, Vorkenntnisse, Ressourcen und Ziel ganz genau hinterfragt. Man versucht in diesem Schritt den Auftrag ganz genau zu untersuchen, damit man bereits offene Fragen abdecken und mögliches Fehlerpotenzial aufweisen kann, um einen groben Überblick zu erhalten. Dieser Schritt dient als Grundlage für das nächste Kapitel «Planung».

# Planen

Im zweiten Schritt der IPERKA-Methode werden nun die möglichen Lösungsvarianten und das Vorgehen ausgearbeitet. Es werden einzelne Arbeitsschritte geplant, um einen gesamten Arbeitsablauf zu erstellen. Im Arbeitsablauf werden Hilfsmittel und Werkzeuge ermittelt, sowie der Zeitbedarf für die Arbeitsschritte geschätzt. Zudem werden Qualitäts-Kriterien für den Schritt «Kontrollieren» gesammelt.

# Entscheiden

Damit man mit dem dritten Schritt von IPERKA beginnen kann, müssen zuerst die Anforderungen aus dem zweiten Schritt «Planen» erfüllt sein. Nach der Planung müssen die verschiedenen Lösungsvarianten gegenübergestellt werden und eine davon ausgewählt werden. Hierbei werden verschiedene Faktoren miteinander abgeglichen und ausgewertet.

# Realisieren

Die Realisierung und somit der vierte Schritt der IPERKA-Methode, ist die zeitintensivste Phase in den meisten Projekten. Hier werden die zuvor erfüllten Schritte von IPERKA (Planen & Entscheiden) aus der Theorie in die Praxis umgewandelt. Dabei ist es wichtig den Arbeitsablauf einzuhalten und nicht ohne zwingende Gründe zu ändern. In diesem Schritt ist es äusserst wichtig genügend Zeit für eine gute Reflexion einzuplanen, sodass man schnell auf Misserfolge reagieren kann und die Massnahmen daraus planen und umsetzen kann.

# Kontrollieren

Der wichtigste Schritt in der IPERKA-Methode ist das Kontrollieren. Man kann seine Arbeit nicht abgeben, ohne dass man alles nochmals kontrolliert hat. Hierbei werden vor allem die Vorgaben, welche sich aus dem Schritt «Planen» ergeben haben, mit dem tatsächlichen Ergebnis verglichen und die Resultate festgehalten. Neben der Kontrolle des eigentlichen Produktes, ist es auch wichtig das Arbeitsverhalten zu überprüfen und allenfalls Abweichungen aus den Plänen aufzuzeigen.

# Auswerten

Im letzten Schritt von IPERKA reflektiert man nochmals, vom Informieren bis zum Kontrollieren die gesamte Bearbeitung des Auftrags. Beim Auswerten geht es um eine Selbsteinschätzung der erledigten Tätigkeiten. Man überlegt sich darüber hinaus was man aus der Arbeit gelernt hat, was man verbessern kann und wo man das Gelernte in der Zukunft anwenden kann.

# Glossar

Im Glossar werden die wichtigsten Fremdwörter und Technische Ausdrücke erklärt.

|  |  |
| --- | --- |
| **Begriff** | **Erklärung** |
| Vulnerability | Vulnerability, also eine Vulnerabilität, ist eine Schwachstelle in der Applikation. |
| Continious Integration (CI) | Kontinuierliche Integration beschreibt den Prozess, bei dem man fortlaufend Komponenten aus der Software zusammenfügt. CI wird vor allem für die stetige Steigerung der Softwarequalität genutzt. |
| Model View ViewModel (MVVM) | MVVM ist eine Beschreibung einer Architektur in der Software-Programmierung |
| Daily | Ein Daily ist eines der Scrum Artefakte. Ein Daily ist ein tägliches Meeting innerhalb des Entwicklerteams, wo über abgeschlossene und bevorstehende Arbeiten geredet wird. |
| ORM | Object-Relational Mapping, ist ein Konzept zum Abbildung von Objekten in einer relationalen Datenbank. |
| Wire-Frames | Wire-Frames werden dazu benutzt um eine konzeptionelle Abbildung einer Benutzeroberfläche darzustellen. Dabei wird wenig auf Design und Funktionalität geachtet, sondern man konzentriert sich auf die Anordnung der Elemente |
| UI-Controls | UI (User Interface) ist die Bezeichnung für die Benutzeroberfläche. UI-Controls sind die angewendeten Bausteine, für die Benutzeroberflächen-Programmierung. |
| Inversion of Control (IOC) | Dieses Programmierparadigma beschreibt die Arbeitsweise einer Einheit, eines Programms, oder eines Frameworks. In meinem Beispiel: Statt, dass die einzelnen Klassen sich um ihre Abhängigkeiten kümmern, wird die Kontrolle einer anderen Einheit übergeben. Sie regelt den Kontrollfluss und löst die Abhängigkeiten auf. |
| Command | Ein Command ist ein Befehl. Wenn ein Benutzer etwas klickt, führt er einen Befehl aus, welcher in der Applikation etwas auslöst. |
| Kohäsion | Kohäsion beschreibt, wie gut eine Einheit seine Logik abbildet. In der Programmierung versucht man eine möglichst hohe Kohäsion zu erreichen. Das heisst, dass jede Einheit(Klasse) nur für eine Aufgabe zuständig ist. |
| Mocks | Mocks, meistens Mockups, sind Vorführmodelle, welche in der Programmierung zur Modellierung von gewissen Einheiten genutzt werden. |

Tabelle 8: Glossar

# Verzeichnisse

Anbei werden alle verwendeten Ressourcen aufgelistet.

## Quellenverzeichnis

Developer, C. C. (2015 ). Clean Code Prinzipien. Abgerufen am 09. November 2018 von https://clean-code-developer.de/

mackenir. (2009). Databinding TabControl. Abgerufen am 07. November 2018 von https://stackoverflow.com/questions/686074/wpf-tabcontrol-databinding

S.A, S. (2008). Leading Product for CI. Abgerufen am 30. Oktober 2018 von https://www.sonarqube.org/

VSSM, V. s. (2014). Die 6-Schritte-Methode. Abgerufen am 24. Oktober 2018 von https://www.vssm.ch/sites/default/files/vssm/BB/docs/AusbildungEFZ/IPERKA\_1\_4\_D2017.pdf

Wikipedia. (16. September 2018). MVVM. Abgerufen am 01. November 2018 von https://de.wikipedia.org/wiki/Model\_View\_ViewModel

## Abbildungsverzeichnis

[Abbildung 1: Gantt-Diagramm 8](#_Toc530582091)

## Tabellenverzeichnis

[Tabelle 2: Beteiligte Personen 5](#_Toc530573644)

[Tabelle 3: Meilensteine 6](#_Toc530573645)

[Tabelle 5: Arbeitsprotokoll 31.10.2018 8](#_Toc530573646)

[Tabelle 5: Arbeitsprotokoll 31.10.2018 9](#_Toc530573647)

[Tabelle 5: Arbeitsprotokoll 31.10.2018 10](#_Toc530573648)

[Tabelle 5: Arbeitsprotokoll 31.10.2018 11](#_Toc530573649)

[Tabelle 31: Glossar 19](#_Toc530573650)

# Aufgabenstellung

# Programmcode