

# IPA Marc Egli - Puzzle ITC

IPA-Daten und beteiligte Personen	
Firma, Abteilung	Puzzle ITC, /dev/ruby
Berufsschule	GIBB
Valid Experte	Lawson Mike
Hauptexpertin	Müller Lorenz
Nebenexperte	Moser Michael
Verantwortliche Fachkraft	Illi Daniel
Zusätzliche Verantwortliche Fachkraft	Steiner Robin
Berufsbildner	Steiner Robin
Fachrichtung	Applikationsentwicklung
Projektvorgehensmodell	SCRUM
Jahrgang der IPA-Durchführung und Kanton	IPA 2025, Kanton Bern
Abgabedatum	22.01.2025

Tabelle 1: IPA Daten

# Teil I

## Ablauf, Organisation und Umfeld

# Inhaltsverzeichnis

<b>I</b>	<b>Ablauf, Organisation und Umfeld</b>	<b>1</b>
<b>1</b>	<b>Aufgabenstellung</b>	<b>5</b>
1.1	Titel der Arbeit . . . . .	5
1.2	Thematik . . . . .	5
1.3	Ausgangslage . . . . .	5
1.4	Detaillierte Aufgabenstellung . . . . .	6
1.4.1	Mittel und Methoden . . . . .	8
1.4.2	Vorkenntnisse . . . . .	8
1.4.3	Vorarbeiten . . . . .	9
1.4.4	Neue Lerninhalte . . . . .	9
1.4.5	Arbeiten in den letzten 6 Monaten . . . . .	9
<b>2</b>	<b>Firmenstandards</b>	<b>10</b>
2.1	Code conventions . . . . .	10
2.1.1	Lizenz . . . . .	10
2.2	Git conventions . . . . .	11
2.3	Documentation Conventions . . . . .	11
<b>3</b>	<b>IPA-Schutzbedarfsanalyse</b>	<b>12</b>
3.1	Datensicherheit . . . . .	12
3.2	Applikationssicherheit . . . . .	12
<b>4</b>	<b>Organisation der IPA-Ergebnisse</b>	<b>14</b>
4.1	Datensicherung . . . . .	14
4.1.1	Dokumentation . . . . .	14
4.1.2	Code . . . . .	15
4.1.3	Wiederherstellung des Codes . . . . .	15
<b>5</b>	<b>Projektmethode</b>	<b>16</b>
5.1	Github Board . . . . .	16
5.1.1	Backlog . . . . .	16
5.1.2	Refinement . . . . .	16
5.1.3	Sprint Backlog . . . . .	17
5.1.4	In Progress . . . . .	17
5.1.5	Done . . . . .	17
5.2	Sprints . . . . .	17
5.3	Sprint Planning . . . . .	18

5.4	Daily . . . . .	18
5.5	Verwendungsgrund . . . . .	18
<b>6</b>	<b>Projektaufbauorganisation</b>	<b>19</b>
<b>7</b>	<b>Zeitplan</b>	<b>20</b>
7.1	Erläuterung zum Zeitplan . . . . .	20
<b>8</b>	<b>Arbeitsjournale</b>	<b>21</b>
8.1	Tag 1: 14.01.2025 . . . . .	21
8.2	Tag 2: TODO: Datum . . . . .	24
8.3	Tag 4: TODO: Datum . . . . .	25
8.4	Tag 1: TODO: Datum . . . . .	26
8.5	Tag 5: TODO: Datum . . . . .	27
8.6	Tag 6: TODO: Datum . . . . .	28
8.7	Tag 7: TODO: Datum . . . . .	29
8.8	Tag 8: TODO: Datum . . . . .	30
8.9	Tag 9: TODO: Datum . . . . .	31
8.10	Tag 10: TODO: Datum . . . . .	32
<b>9</b>	<b>Persönliches Fazit</b>	<b>33</b>
<b>II</b>	<b>Projektdokumentation</b>	<b>34</b>
<b>10</b>	<b>Einführung</b>	<b>35</b>
<b>11</b>	<b>Analyse</b>	<b>36</b>
11.1	Ist-Zustand . . . . .	36
11.2	Soll-Zustand . . . . .	36
11.3	Persönliche Vorgehensziele . . . . .	36
11.4	Anforderungen . . . . .	36
11.4.1	Nicht funktionale Anforderungen . . . . .	36
11.4.2	Funktionale Anforderungen . . . . .	36
11.5	Abgrenzung . . . . .	36
<b>12</b>	<b>Entwurf</b>	<b>37</b>
12.1	Lösungsvarianten . . . . .	37
12.2	Variantenentscheid . . . . .	37
12.3	Ausarbeitung . . . . .	37
<b>13</b>	<b>Ausführung</b>	<b>38</b>
13.1	Testprotokoll . . . . .	38
<b>14</b>	<b>Einführung</b>	<b>39</b>
<b>15</b>	<b>Sprintabschlüsse</b>	<b>40</b>
15.1	Abschluss Sprint Initialisierung . . . . .	40

15.2	Abschluss Sprint Umsetzung . . . . .	40
15.3	Abschluss Sprint Finalisierung . . . . .	40
<b>III</b>	<b>Anhang und Verzeichnisse</b>	<b>41</b>
<b>16</b>	<b>Verzeichnisse</b>	<b>42</b>
16.1	Tabellenverzeichnis . . . . .	42
16.2	Abbildungsverzeichnis . . . . .	42
16.3	Code Verzeichnis . . . . .	42
	Quellenverzeichnis . . . . .	43
<b>17</b>	<b>Verwendete Abkürzungen</b>	<b>44</b>
<b>18</b>	<b>Glossar</b>	<b>45</b>
<b>19</b>	<b>Anhänge</b>	<b>46</b>
19.1	Sitzungsprotokolle . . . . .	46
19.2	Git commit convention . . . . .	46
19.3	Security conventions . . . . .	46
19.4	Code . . . . .	47

# 1 Aufgabenstellung

## 1.1 Titel der Arbeit

Hitobito: Neue Generation von Personen-Filtern

## 1.2 Thematik

Eines der Kernfunktionalitäten von Hitobito ist das Filtern via vom Benutzer definierten Kriterien von Personen auf Personenlisten und Abos. Diese Funktionalität ist in den über 10 Jahren seit es Hitobito gibt oft erweitert worden. Durch die vielen neuen Filtermöglichkeiten wurde speziell das UI immer komplexer und unübersichtlicher. Die Personen-Filteroptionen für Personenlisten und die der Abos sehen ähnlich aus, weisen aber diverse nicht offensichtliche Unterschiede auf. Mit dieser Probe-IPA soll für den Backendteil der Abos (MailingLists) eine neue Generation von Personen-Filtern für Hitobito entwickelt werden.

## 1.3 Ausgangslage

Hitobito ist eine Open Source Webapplikation zum Verwalten von Mitgliedern, Events und vielem mehr. Die Ruby on Rails Applikation wurde 2012 von Puzzle ITC initiiert und wird stets weiterentwickelt.

Die Basis für die Software bildet das Webframework Ruby on Rails. Für das User Interface wird neben statischer Technologie wie HTML und CSS auch JavaScript oder Hotwire verwendet. Der komplette Source-Code steht auf Github zur Verfügung: <https://github.com/hitobito>

## 1.4 Detaillierte Aufgabenstellung

Mit dieser Probe-IPA soll ein neues Konzept und Datenmodell für die Persistierung von Filter-Parametern erstellt werden (rein Backend). Anschliessend soll dieses Konzept in einem Proof of Concept (PoC) bei einem Teil der Mailinglisten (Abos) umgesetzt werden.

- Die Klassen Subscription, RelatedRoleType, PeopleFilter, usw. werden im neuen Konzept komplett ersetzt oder ggf. ergänzt
- Eine Möglichkeit ist das PeopleFilter die Basis für das neue Konzept bilden
- Es sollen 2-3 Grobkonzepte gegenüber gestellt werden und das ausgewählte Konzept detaillierter ausgearbeitet werden

### PoC

- Folgende Komponenten der MailingLists Filter sollen mit dem neuen Konzept im PoC umgesetzt werden:
  - Globale Bedingungen & Sprache
  - Personen
  - Ausgeschlossene Personen
  - Optional: Gruppen / Rollen
- Persistierte Subscriptions/Filter müssen für den PoC vorerst nicht migriert werden
- Die nicht erwähnten Komponenten müssen nicht mehr funktionieren
- Die erwähnten Komponenten (ohne Optionale) funktionieren im UI und haben eine minimale, funktionierende Testabdeckung (happy path)

Out of Scope - wird nicht oder erst nach der Probe IPA umgesetzt

- Konzept und Anpassungen Frontend/UI
- PoC Umbau/Migration People Filter Personenlisten
- JSON API Filter (Grafiti)



### 1.4.1 Mittel und Methoden

Technologie und Plattform:

- Ruby, Ruby on Rails, Active Record

Entwicklungsumgebung:

- IntelliJ
- Git, Github
- Rake
- Rubocop

Textverarbeitung und Diagramme:

- Latex
- draw.io
- Google Sheets

Projektmethode:

- Scrum IPA

Konventionen:

- Es gilt der [Ruby Style Guide](#) und der [Rails Style Guide](#) gemäss Rubocop [Konfiguration des Projekts](#)

### 1.4.2 Vorkenntnisse

Marc arbeitet bereits seit einigen Monaten an Features von Hitobito. Ausserdem hat er bereits seit dem 2. Lehrjahr Erfahrung auch in anderen Ruby on Rails Projekten gesammelt.

### 1.4.3 Vorarbeiten

- Vorbereitung Dokumentvorlage
- Ist-Analyse Personen-Filter Personen-Listen/Abos
- Dokumentation in der Developer-Dokumentation der bestehenden Implementation von MailingLists, FilteredList, Personen-Filter

### 1.4.4 Neue Lerninhalte

- Eigenständiges Entwerfen der Datenstruktur/Klassen

### 1.4.5 Arbeiten in den letzten 6 Monaten

- Umsetzung diverser Features für Hitobito (Ruby on Rails)
- Postgresql Migration Hitobito

## 2 Firmenstandards

### 2.1 Code conventions

Als Code convention werden die Ruby [Style Guides](#) verwendet. Die Überprüfung dieser Style Guidelines wird mit Rubocop (Formatter) sichergestellt. Die Konfiguration dieses Formatters ist unter [rubocop.yml](#) ersichtlich.

#### 2.1.1 Lizenz

In jedem File in Hitobito wird das Copyright für den jeweiligen Kunden und die Lizenz dazu in Kommentarform beschrieben. Diese Lizenz- sowie Kundeninformationen können über folgenden Befehl eingefügt werden.

```
rake license:insert
```

Alternativ dazu können diese Informationen mit

```
rake license:remove
```

entfernt oder mit

```
rake license:update
```

aktualisiert werden.

## 2.2 Git conventions

Für das cloudbasierte Hosting unseres Git-Repositories wird Github verwendet. Die Git Commitnachrichten werden nach den Regeln von Puzzle ITC formuliert. Im Anhang unter Git Conventions finden sie eine Kopie unserer Firmenkonventionen

- Sprache: Englisch
- Kurze und prägnante Message, idealerweise unter 50 Zeichen [Details](#)
- Mit Grossbuchstaben beginnen [Details](#)
- Kein Punkt am Schluss [Details](#)
- Den *imperative mood* (Befehlsform) verwenden, also «Fix bug with X» statt «Fixed bug with X» oder «More fixes for broken stuff» [Details](#)
- Wenn vorhanden Ticket referenzieren:
  - Bei Open Project Work Packages: «Add X, refs #12345»
  - Bei Gitlab/Github Issues: «Add X #12345»

## 2.3 Documentation Conventions

Als Documentation covention wird arc42 verwendet (Siehe [arc 42 documentation](#)).

## 3 IPA-Schutzbedarfanalyse

### 3.1 Datensicherheit

Die notwendigen Daten welche im Rahmen der IPA zu Test- und Vorführungszwecken verwendet werden, sind werden durch das [Faker-Gem](#) generiert und sind somit NICHT besonders schützenswert. Dazu gehören unter anderem Adressen, Familiendaten, Finanzdaten.

### 3.2 Applikationssicherheit

Da keine besonders schützenswerte Daten verwendet werden, werden die Standards von Puzzle ITC nach Firmen-Sicherheitsbuch verwendet. Sie können dieses unter Security Conventions einsehen. Diese Security Conventions umfassen:

- Injection / Cross Site Scripting
  - Input Validierung von allen Inputs serverseitig durchführen
  - Output Encoding auf allen Outputs anwenden
  - Kein inline oder dynamisches SQL, sondern parametrisierte Queries verwenden
  - Datei Uploads überprüfen
- Verbindungs- / Browsersicherheit
  - Nur HTTPS verwenden und korrekt konfigurieren
  - Security Headers setzen
  - Cookie Flags secure, httpOnly und SameSite setzen
  - Kein Caching von sensiblen Informationen

- Authentication / Sessions
  - IAM des Frameworks oder besser Keycloak verwenden
  - Keine sensitiven Infos in URL Parameter
  - Brute Force Schutz
  - Sessions schützen
- Tools und Betriebsumgebung
  - Errorhandling und Logging
  - Libraries und deren Dependencies auf bekannte Schwachstellen prüfen
  - OS, Webserver, Container aktuell halten und Hardening
  - Keine Produktionsdaten auf Integrationsumgebungen
- Security Testing
  - Es dürfen keine Secrets im Repository abgelegt werden
  - Eingebundene Dependencies dürfen keine MEDIUM und HIGH Schwachstellen aufweisen
  - Eine statische Codeanalyse sollte durchgeführt werden
  - Eine dynamische Codeanalyse sollte durchgeführt werden
  - Alle verwendeten Images sollten auf Schwachstellen gescannt werden

## 4 Organisation der IPA-Ergebnisse

### 4.1 Datensicherung

In dieser IPA unterteilen wir die Datensicherung in:

- Dokumentation
- Code

#### 4.1.1 Dokumentation

Dokumentation	
Tools	Git und USB
Versioniert	Ja
Interval	Mind. 2x täglich
Beschreibung	Die Dokumentation ist im ipa-puzzle-template Repository unter dem Branch probe-ipa angelegt. Sobald ein Dokumentationsticket abgeschlossen wurde, werden die Änderungen auf den Github Server in das private Repository gepushed. Dies geschieht mind. 2x täglich. Zusätzlich, wird pro Tag ein Ordner auf einem USB-Stick erstellt. Am Ende des Tages wird eine Kopie der Dokumentation in diesen Ordner geladen.

Tabelle 4.1: Sicherung Dokumentation

### 4.1.2 Code

Code	
Tools	Git und USB
Versioniert	Ja
Interval	Mind. 2x täglich
Beschreibung	Für die Entwicklung habe wurden die Repositories hitobito und hitobitogeneric geforked. Auf diesen Repositories wird an Tagen an welchen Entwickelt wird, mind. 2x täglich committed. An diesen Tagen wird zur doppelten Sicherung zusätzlich eine Kopie des Projektes auf den USB Stick gespeichert, unter dem Ordner des jeweiligen Tages.

Tabelle 4.2: Sicherung Code

### 4.1.3 Wiederherstellung des Codes

Gehen die Daten lokal verloren, können diese entweder über das Github Repository oder den USB-Stich wiederhergestellt werden. Bei der Wiederherstellung mit Git, wird der SSH-Key des Repositories benötigt, damit dieses von Github geklont werden kann. Ist dieser SSH-Key nicht verfügbar, wird die Wiederherstellung über den USB-Stick vorgenommen und das Projekte des letzten Speicherstandes kopiert. Im Falle des USB-Sticks sind mit mehr Datenverlusten zu rechnen, falls der Datenverlust gegen Mittag oder Nachmittag auftritt, da die Speicherung erst am Ende des Tages erfolgt. Aus diesem Grund ist die Datenwiederherstellung mit Git zu bevorzugen.

Die Nachweise für die jeweiligen Datensicherungen finden sie im Anhang unter: TODO(Screenshots in Anhang einfügen)

- USB-Sicherung
- Git-Sicherung



## 5 Projektmethode

Die verwendete Projektmethode dieser IPA ist SCRUM. Abweichungen und Werkzeuge welcher der Umsetzung dieser IPA nach SCRUM verwendet werden, sind im folgenden Abschnitt beschrieben.

### 5.1 Github Board

Um die Userstories, Aufwandschätzungen und den Projektstatus zu verfolgen verwende ich Github Projects.

#### 5.1.1 Backlog

Zu Beginn der IPA wurde ein Backlog erstellt indem alle User Stories aufgeführt werden. Die Stories im Backlog müssen noch nicht detailliert spezifiziert sein, sie dienen dazu eine Übersicht über noch offene Aufgaben während der IPA zu erhalten.

#### 5.1.2 Refinement

In der Refinement Spalte werden die Userstories vor dem Sprint Planning detaillierter beschrieben und mit Akzeptanzkriterien versehen. Der Detailbeschrieb dient dazu die Story später im Sprint Planning besser schätzen zu können. Falls eine Userstory zu gross wird, wird sie in dieser Spalte auf zwei oder mehrere Stories unterteilt. Ausserdem werden pro Userstory Akzeptanzkriterien definiert welche erfüllt werden müssen um diese während des Sprints abzuschliessen.

### 5.1.3 Sprint Backlog

Anfangs Sprint wird immer ein Sprint Planning durchgeführt. Dabei werden die Userstories geschätzt und in den Sprint Backlog gezogen. Am Ende des Sprints sollte der Sprint Backlog leer sein. Ist dies nicht der Fall muss die Story zurück ins Refinement, neu beschrieben werden (falls Änderungen aufgetaucht sind) und muss dann in den nächsten Sprint weitergezogen werden.

### 5.1.4 In Progress

Während des Sprints werden Ticket in die In Progress-Spalte geschoben sobald die Arbeit daran beginnt.

### 5.1.5 Done

Eine Userstory kann in die Done-Spalte gezogen werden, wenn alle Akzeptanzkriterien erfüllt wurden. Die Story gilt danach als abgeschlossen.

## 5.2 Sprints

Die gesamte IPA wird in drei Sprints unterteilt, diese umfassen je eine der folgenden Phasen:

- Initialisierung
- Umsetzung
- Finalisierung

Jedes Ticket wurde mit einem der Phasen gelabeled. So kann abgeschätzt werden, welche Tickets in welchem Sprint erledigt werden müssen.

## 5.3 Sprint Planning

Das Planning findet immer zu Beginn des nächsten Sprints statt. Während des Sprint Plannings werden die zu erledigenden Stories vom Refinement in den Sprint Backlog geschoben und geschätzt. Um die Planung im Zeitplan besser darzustellen, wird definiert dass die Stories in Stunden anstatt Story Points geschätzt werden. Die niedrigste Schätzung entspricht dabei einem Betrag von 0.5 Stunden.

## 5.4 Daily

Jeden Morgen findet ein Daily mit der verantwortlichen Fachkraft und der zusätzlichen verantwortlichen Fachkraft statt welche den Stand des Sprints prüfen und offene Fragen von mir beantworten. Ausserdem präsentiere ich im Daily den Stand der Dokumentation welche meine zuständigen Fachkräfte prüfen und mir Tipps zur Verbesserung geben.

## 5.5 Verwendungsgrund

Die Projektvorgehensmethod wurde so gewählt, da sie für die IPA mehrere Vorteile bringt:

- **Sprint Ende:** SCRUM zwingt den Entwickler dazu am Ende des Sprints ein brauchbares Produkt zu haben
- **Agilität:** Wenn eine Story nicht erreicht wurde, kann sie in den nächsten Sprint gezogen werden
- **Daily:** Durch die Dailies wird ein täglicher Austausch zwischen Fachkraft und Kandidat sichergestellt
- **Akzeptanzkriterien:** Mit den Kriterien verhindern wir das abschliessen von halbfertigen Features oder fehlerhafter Software
- **Board:** Durch das Github Projects Board ermöglichen wir eine schnelle Übersicht über den Stand der IPA

## 6 Projektaufbauorganisation

# 7 Zeitplan

## 7.1 Erläuterung zum Zeitplan

## 8 Arbeitsjournale

### 8.1 Tag 1: 14.01.2025

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Planning	Marc Egli	1	1
Zeitplan	Marc Egli	2	2
Aufgabenstellung übernehmen	Marc Egli	1	0.5
Standards aus Github übernehmen	Marc Egli, Nils Rauch	1	1.5
IPA Schutzbedarfanalyse	Marc Egli, Nils Rauch, Olliver Brian, Olliver Dietschi, Thomas Ellenberg	1	0.75
Scrum Beschrieb	Marc Egli	1	1.5
Arbeitsjournal	Marc Egli	0.25	0.5
Backupkonzept	Marc Egli	1	0.25
<b>Total</b>		8.25	8.25

Tabelle 8.1: Tätigkeiten Tag 1

### Tagesablauf

Heute bin ich motiviert in die IPA gestartet. Als erstes habe ich am morgen nochmals die Spezifikationen für die Dokumentation, durchgelesen und das Template für die IPA angepasst. Nachdem ich eine passende Struktur hatte, startete ich auch schon direkt mit dem ersten Sprint Planning dieser IPA. Dabei habe ich alle Tasks für den Sprint 1 im Backlog erfasst, diese dann im Refinement detaillierter Beschrieben und am Schluss in den Sprint Backlog geschoben. Die ganze Planung habe ich mit Github Projects gemacht, leider kam ich da bezüglich Issues an die Grenzen denn leider kann mann diese nur definieren wenn die Issues einem Projekt, welches NICHT geforked ist,

zugewiesen werden können. Dieses Problem werde ich am Daily morgen mit meiner Fachkraft besprechen, evtl. weis er mehr dazu.

Nach dem Planning begann ich mit dem Bereitstellen des Zeitplans. Ich übernahm das Tempalte welches ich ausgewählt hatte und passte es auf meine drei Sprints in den kommenden zwei Wochen an. Zuerst dachte ich, dass ich den Zeitplan schneller fertigstellen könnte jedoch hatte ich Probleme mit Google Sheets und das anlegen von gemergeden Spalten dauerte lange. Trotzdem ist die Planung aufgegangen und nach 2 Stunden hatte ich einen geeigneten Zeitplan.

Am Nachmittag Startete ich direkt mit dem Dokumentieren, angefangen bei den Standards unserer Firma. Es dauerte länger als gedacht, alle Standards zu sammeln und in die Struktur der Dokumentation zu bringen, weswegen ich dort etwas Zeit verlor. Ein Teil davon konnte ich dann bei der Schutzbedarfsanalyse wieder reinholen. Hier suchte ich den Kontakt mit anderen Mitarbeitern, um herauszufinden wo das Datenschutzkonzept für Hitobito hinterlegt ist. Anscheinend wusste das Niemand aussert Oliver Brian, welcher mir dieses für die Ablage im Anhang zur Verfügung stellte.

Gegen den Ende des Tages habe ich die Projektmethode Scrum Beschrieben und dokumentiert wie ich mich während der IPA organisieren werde. Bezüglich der Aufteilung der Spalten der User Stories bin ich hier noch unsicher, ich werde dies sicher morgen am Daily auch mit Daniel Illi abklären.

## Hilfestellungen

- Oliver Brian: Nachfrage Datenschutzkonzept
- Nils Rauch: Nachfrage Sicherheitskonzept / Sicherheitsconventions  
Puzzle ITC

## Reflexion

Ich konnte heute schon einiges dokumentieren und habe nun eine Vorlage von der aus ich einfach weiterarbeiten kann. Zusätzlich habe ich mit Github Project einen Ort an dem ich meinen Fortschritt verwalte und mich selbst organisiere. Probleme gab es nur bei der Beschaffung des Datenschutzkonzeptes und der Arbeit mit Google Sheets.

## Was lief gut

Grundsätzlich lief das Dokumentieren selbst sehr gut. Ich konnte alle restlichen Informationen für die Standars oder die Projektmethode schnell Beschaffen und mich dann dem Dokumentieren widmen.

## Was lief weniger gut

Weniger gut lief die Arbeit mit Google Sheets und die Arbeit mit der Latex Vorlage. Zum Teil hatte ich recht lange bis ich herausfand wie ich eine Liste anlege oder ein Bild einfügen kann. Ausserdem habe ich mich im Zeitplan verschätzt und heute 9.25 anstatt 8.25 Stunden geschätzt, da ich im Google Sheets einen Fehler gemacht habe. Diesen konnte ich aber schnell korrigieren, so dass ich heute auf geplante 8.25 Stunden komme, welche ich nun auch erreiche.

## Meine Erkenntnisse von heute

Mit erweitertem Latex know-how und dem Datenschutzkonzept in den Händen kann ich nun weiter dokumentieren. Ich denke ich werde somit auch weniger Probleme mit Google Sheets und Latex haben, da ich heute schon viele meiner Probleme lösen konnte.

## Nächste Schritte

Als nächstes werde ich morgen das Backupkonzept fertig machen und dann direkt zur Projektaufbauorganisation gehen. Nach Abschluss dieser Story kann ich den Sprint 1 Abschiessen und schon in den Sprint 2, der Konzeption / Umsetzung starten.



## 8.2 Tag 2: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.2: Tätigkeiten Tag 2

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.3 Tag 4: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.3: Tätigkeiten Tag 4

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.4 Tag 1: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.4: Tätigkeiten Tag 1

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.5 Tag 5: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.5: Tätigkeiten Tag 5

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.6 Tag 6: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.6: Tätigkeiten Tag 6

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.7 Tag 7: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.7: Tätigkeiten Tag 7

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.8 Tag 8: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.8: Tätigkeiten Tag 8

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 8.9 Tag 9: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.9: Tätigkeiten Tag 9

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte



## 8.10 Tag 10: TODO: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
TODO: Tätigkeit	TODO: Beteiligte Personen	TODO: Stunden Soll	TODO: Stunden Ist
<b>Total</b>		TODO: Stunden Soll Total	TODO: Stunden Ist Total

Tabelle 8.10: Tätigkeiten Tag 10

### Tagesablauf

### Hilfestellungen

- TODO: Hilfestellungen auflisten

### Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

### Nächste Schritte

## 9 Persönliches Fazit

## Teil II

# Projektdokumentation

**Hitobito: Neue Generation von Personen-Filtern**  
**Autor: Marc Egli**

# 10 Einführung

# 11 Analyse

## 11.1 Ist-Zustand

## 11.2 Soll-Zustand

## 11.3 Persönliche Vorgehensziele

## 11.4 Anforderungen

### 11.4.1 Nicht funktionale Anforderungen

### 11.4.2 Funktionale Anforderungen

## 11.5 Abgrenzung

# 12 Entwurf

## 12.1 Lösungsvarianten

## 12.2 Variantenentscheid

## 12.3 Ausarbeitung

# 13 Ausführung

## 13.1 Testprotokoll

Resultat Testfall Nr. 1	
Testname	
Testkontext	
Testperson	
Ausführungs Datum	
Testergebnis	
Beschreibung	
Fehlerklasse	

Tabelle 13.1: Resultat Testfall 1

# 14 Einführung



# 15 Sprintabschlüsse

## 15.1 Abschluss Sprint Initialisierung

## 15.2 Abschluss Sprint Umsetzung

## 15.3 Abschluss Sprint Finalisierung

## Teil III

# Anhänge und Verzeichnisse

**Hitobito: Neue Generation von Personen-Filtern**  
**Autor: Marc Egli**

# 16 Verzeichnisse

## 16.1 Tabellenverzeichnis

1	IPA Daten . . . . .	1
4.1	Sicherung Dokumentation . . . . .	14
4.2	Sicherung Code . . . . .	15
8.1	Tätigkeiten Tag 1 . . . . .	21
8.2	Tätigkeiten Tag 2 . . . . .	24
8.3	Tätigkeiten Tag 4 . . . . .	25
8.4	Tätigkeiten Tag 1 . . . . .	26
8.5	Tätigkeiten Tag 5 . . . . .	27
8.6	Tätigkeiten Tag 6 . . . . .	28
8.7	Tätigkeiten Tag 7 . . . . .	29
8.8	Tätigkeiten Tag 8 . . . . .	30
8.9	Tätigkeiten Tag 9 . . . . .	31
8.10	Tätigkeiten Tag 10 . . . . .	32
13.1	Resultat Testfall 1 . . . . .	38
17.1	Verwendete Abkürzungen . . . . .	44
18.1	Glossar . . . . .	45

## 16.2 Abbildungsverzeichnis

19.1	Puzzle ITC Git commit conventions . . . . .	46
19.2	Puzzle ITC security conventions 1/3 . . . . .	46
19.3	Puzzle ITC security conventions 2/3 . . . . .	47
19.4	Puzzle ITC security conventions 3/3 . . . . .	47

## 16.3 Code Verzeichnis

# Quellenverzeichnis

[TODO: Name der Quelle] [TODO:URL](#)`in\protect\unhbox\voidb@x\bgroup\U@D1ex{\setbox\z@\hbox{\char127}\dimen@-.45ex\advance\dimen@\ht\z@}\accent127\fontdimen5\font\U@Du\egroup`gen, (TODO: Datum von Tag wo Quelle verwendet wurde)

# 17 Verwendete Abkürzungen

Abkürzung	Bedeutung
TODO: Abkürzung	TODO: Beschreibung

Tabelle 17.1: Verwendete Abkürzungen

## 18 Glossar

Bezeichnung	Bedeutung
TODO: Wort	TODO: Beschreibung

Tabelle 18.1: Glossar

# 19 Anhänge

## 19.1 Sitzungsprotokolle

## 19.2 Git commit convention

### Konvention Commit Message

Falls keine besonderen Vorgaben durch den Kunden vorhanden, empfehlen wir – angelehnt an den Artikel [How to Write a Git Commit Message](#) – folgende Konvention zu verwenden:

- Sprache: Englisch
- Kurze und prägnante Message, idealerweise unter 50 Zeichen ([Details](#))
- Mit Grossbuchstaben beginnen ([Details](#))
- Kein Punkt am Schluss ([Details](#))
- Den *imperative mood* (Befehlsform) verwenden, also «Fix bug with X» statt «Fixed bug with X» oder «More fixes for broken stuff» ([Details](#))
- Wenn vorhanden das Ticket referenzieren:
  - Bei Open Project Work Packages: «Add X, refs #12345»
  - Bei Gitlab/Github Issues: «Add X #12345»

Dies entspricht grundsätzlich auch dem Stil wie ihn viele Open Source Projekte wie z.B. der [Linux Kernel](#), [Spring Boot](#), [Rails](#) oder auch [Git](#) selber anwenden.

Für grössere Projekte, bei welchen auch das Changelog automatisiert generiert wird, kann die [Conventional Commits](#) Spezifikation sinnvoll sein.

Abbildung 19.1: Puzzle ITC Git commit conventions

## 19.3 Security conventions

PUZZLE ITC

Suchen...

QM-Guide / Security

Bearbeiten

### Securing Web Applications

Verantwortlich

Mark Zeman

Webanwendungen sind relativ einfach zu attackieren, da sie in der Regel einfach zu verstehen und zu manipulieren sind, selbst von Amateur innen. Ob eine Webanwendung sicher ist, hängt davon ab, ob die beteiligten Entwickler:innen und auch die Betreiber:innen sensibilisiert sind und entsprechende Massnahmen implementieren. Unsere Security Guides unterstützen dich und dein Team eure Webanwendungen sicher zu machen und damit die Daten unserer Kund:innen und Benutzer:innen zu schützen.

#### Injection / Cross Site Scripting

Mass

☐ Input Validierung von allen Inputs serverseitig durchführen

☐ Output Encoding auf allen Outputs anwenden

☐ Kein inline oder dynamisches SQL, sondern parametrisierte Queries verwenden

☐ Datei Uploads überprüfen

Soll

☐ Eine WAF einbauen

Inhalt

Injection / Cross Site Scripting

Verbindungs- / Browsericherheit

Authentification / Sessions

Tools und Betriebsumgebung

Security Testing

Weiterführende Informationen

Metrik

Abbildung 19.2: Puzzle ITC security conventions 1/3

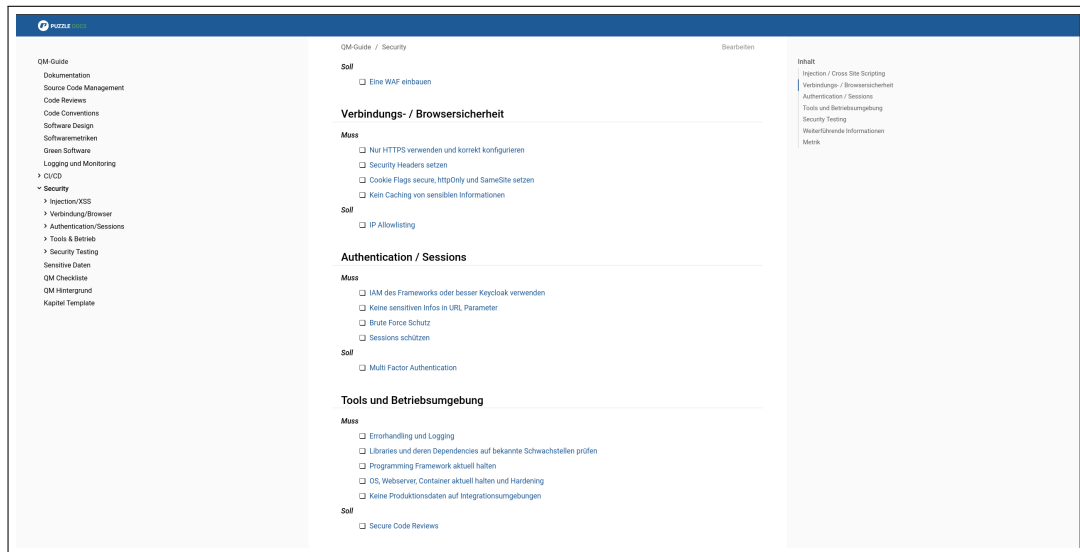


Abbildung 19.3: Puzzle ITC security conventions 2/3

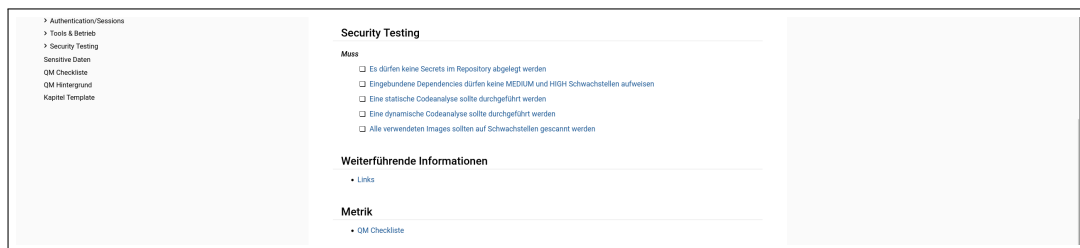


Abbildung 19.4: Puzzle ITC security conventions 3/3

## 19.4 Code