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

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).

2.4 Security Conventions

- Injection / Cross Size 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

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

Obwohl im Rahmen der IPA nicht mit besonders schützenswerten Daten gearbeitet muss bei der Programmierung beachtet werden, dass bei den Filterungen stets auf ein Datenset zugegriffen wird welches durch das can-can-can-Gem validiert wurde um zu Verhindern dass Personen auf die Daten anderer Zugriff haben. Dies ist wichtig, da sich bei späterer Implementierung der IPA in Hitobito besonders Schützenswerte Daten in der Datenbank befinden. Das Datenschutzkonzept dafür finden sie

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 Sprintes 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 Sprintes 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

6.1 Projektrollen in Scrum

Abbildung 6.1: Rollen in Scrum

Rollenbeschreibung	
Product Owner	Der Product Owner vertritt die Interessen des Kunden. Er priorisiert die Aufgaben im Product Backlog
Scrum Master	Der Scrum Master coached die Entwickler und beseitigt Hindernisse. Er sorgt für eine kontinuierliche Verbesserung in der Arbeit.
Entwicklerteam	Das Entwicklerteam arbeitet selbstorganisiert den Sprint Backlog ab. Durch Dailies wird ein laufender Informationsaustausch sichergestellt.

Tabelle 6.1: Rollenbeschreibung

6.2 Projektrollen IPA

Rollenbeschreibung	
Verantwortliche Fachkraft	Unterstützt den Kandidaten von seiten des Lehrbetriebes. Erste Anlaufstelle bei Problemen.
Zusätzliche verantwortliche Fachkraft	Unterstützung für die verantwortliche Fachkraft
Experten	**Validierungsexperte:** Validiert die IPA-Aufgabenstellung. **Hauptexperte:** Verantwortlich für die Bewertung der IPA. **Nebenexperte:** Unterstützung für den Hauptexperten.

Tabelle 6.2: Rollenbeschreibung

6.3 Rollenverteilung

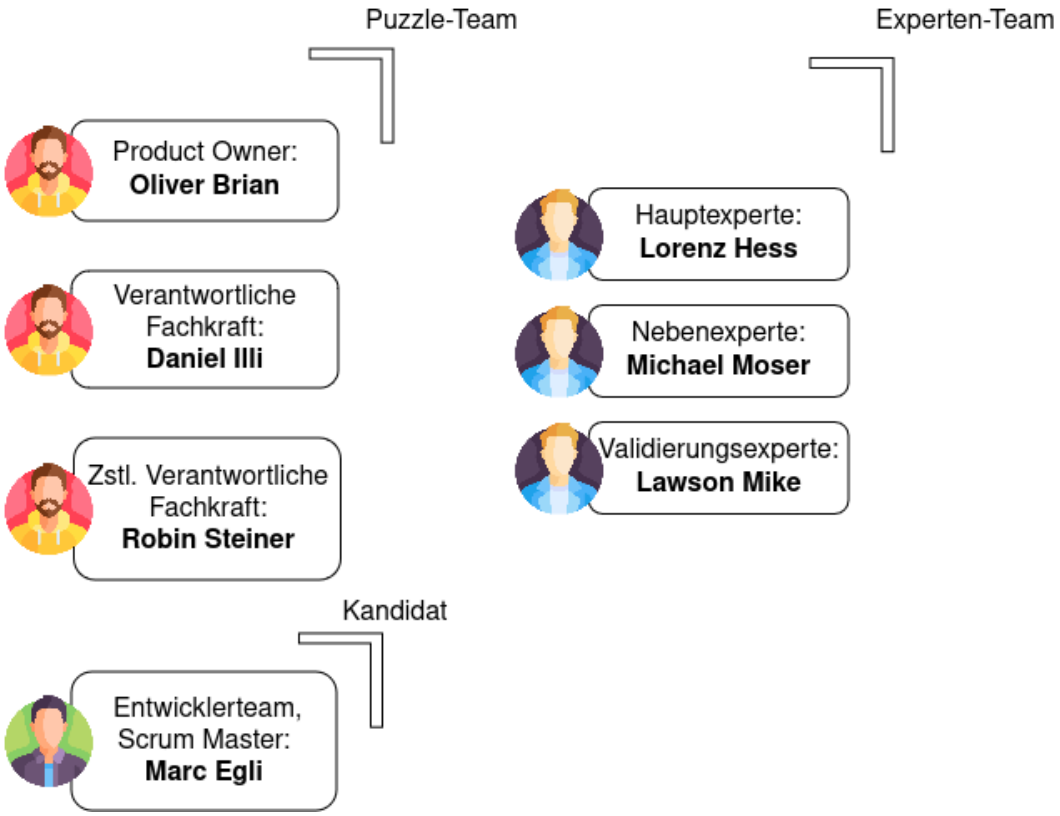

Abbildung 6.2: Rollen in Scrum

Rollenbeschreibung IPA	
Verantwortliche Fachkraft	Daniel Illi
Zusätzliche verantwortliche Fachkraft	Robin Steiner
Validierungsexperte	Lawson Mike
Hauptexperte	Lorenz Hess
Nebenexperte	Michael Moser
Scrum Master	Marc Egli
Development Team	Marc Egli
Kandidat	Marc Egli

Tabelle 6.3: Rollenbeschreibung IPA

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
Total		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 Gihub 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 dauerete 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 Abschliessen und schon in den Sprint 2, der Konzeption / Umsetzung starten.

8.2 Tag 2: 15.01.2024

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Backup Konzept	Marc Egli	0	0.5
Projektaufbauorganisation	Marc Egli	2	2.25
Standards	Marc Egli	0	0.25
Datenschutzkonzept	Marc Egli	0	0.5
Planning	Marc Egli	1	1
Daily	Marc Egli, Daniel Illi	0.25	0.5
Arbeitsjournale	Marc Egli	2	0.25
Total		5.25	5.25

Tabelle 8.2: Tätigkeiten Tag 2

Tagesablauf

Heute konnte ich dank den Erkenntnissen von gestern schnell mit der Latex Dokumentation vorankommen. Das Backup-Konzept konnte ich direkt abschliessen und habe dort sogar noch eine Viertelstunde gespart. Diese brauchte ich wiederum für die Projektaufbauorganisation. Was hier länger gedauert hat, war das erstellen der Diagramme. Ich wollte die Scrum Rollen und Rollenverteilung möglichst übersichtlich machen, was Zeit kostete. Zuletzt waren die Arbeitsjournale geplant, hier habe ich einen Fehler in meiner Planung gemerkt. Ich habe angenommen, dass ich die Arbeitsjournale noch anpassen müsste und wegen der fehlenden Latex-Erfahrung habe ich deswegen zwei Stunden eingeplant.

Allerdings hatten wir schon ein Template für das Arbeitsjournal im Projekt, deswegen hat sich diese Zeit auf 0 Stunden reduziert. Die übrige Zeit habe ich dafür aufgewendet Nachbesserungen an der Dokumentation im Bereich Datenschutzkonzept und Standards zu machen. Zudem hat das Daily auch länger gedauert, welches die nötige Zeit dann rausholen konnte.

Zum Schluss des Tages habe ich den Sprintabschluss und das Planning für die Umsetzungsphase gemacht.

Hilfestellungen

- Nils Rauch: Nachfrage Tool für Erstellung des Zeitplans
- Daniel Illi: Nachfrage Datenschutzkonzept

Reflexion

Heute konnte ich sehr erfolgreich mit der Latex Dokumentation arbeiten. Auch das Planning lief gut, ich denke ich habe nun eine saubere Planung für die Umsetzungsphase welche mir genug Spielraum lässt. Der Fehler mit dem Zeitplan und Arbeitsjournal hat mir zwar Zeit in der Dokumentation gekostet, allerdings ist es besser zu viel Zeit als zu wenig geschätzt zu haben.

Was lief gut

Die Arbeit mit Latex ging heute ohne Probleme voran und meine Effizienz war heute deutlich grösser als gestern.

Was lief weniger gut

Der Fehler im Zeitplan mit den Arbeitsjournalen hat mich in der Planung durcheinandergebracht. Ich habe die Zeiten nun korrekt im Zeitplan vermerkt, damit keine weiteren Probleme darunter entstehen.

Meine Erkenntnisse von heute

Ich sollte vor der Eintragung in den Zeitplan prüfen, ob nicht schon Dokumente existieren welche mir einen Teil der Arbeit abnehmen. Ist dies der Fall, wie bei meinen Arbeitsjournalen kann ich die Aufwandschätzung um ein Wesentliches reduzieren.

Nächste Schritte

Morgen werde ich mit der Umsetzung der IPA starten. Dabei werde ich zuerst die Einführung in das Hitobito Projekt dokumentieren und dann direkt in die Konzeption für eine Filterlösung der Personenlisten und Abos starten. Dies ist ein Schritt der mich zusätzlich motiviert, denn ich kann endlich etwas anderes machen als dokumentieren.

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
Total		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
Total		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
Total		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
Total		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
Total		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
Total		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
Total		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
Total		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

Puzzle ITC ist ein schweizer Anbieter für Softwarelösungen. Die Firma hat ihren Hauptsitz in Bern, besitzt aber weitere Standorte in Zürich, Luzern und Deutschland (Thüringen). Puzzle bietet als Unternehmen die ganze Palette an IT-Services an, von Digital Transformation bis hin zu Data Analytics. Nebst den vielen Angeboten tritt Puzzle dabei immer seine Grundwerte nach aussen, welche im Puzzlehouse abgebildet werden.

Abbildung 10.1: Rollen in Scrum

Hitobito ist eines der Angebote von Puzzle. Es ist ein Community-Management Tool und als Open-Source Projekt auf Github zu finden. Das Tool wird von zahlreichen Verbänden, Parteien und Organisationen verwendet und befindet sich darum in einer kontinuerilichen Weiterentwicklung. Mit dem Wagons-Gem ermöglicht es Hitobito zudem spezielle Kundenanpassungen in einem eigenen ”Wagonßu vollziehen, ohne die Software anderer Kunden mit-anzupassen.

Ich selbst arbeite jetzt seit einem halben Jahr im Hitobito und nahm darin vor allem Upgrades und Migrationen vor. So durfte ich bspw. das Upgrade von RoR (Ruby on Rails) von 6.1 auf 7.1 vornehmen oder die Migration von MySQL auf Postgres vollziehen.

Da Hitobito von zahlreichen Kunden verwendet wird, ist die Applikation über die Jahre gewachsen. Viele Features wurden implementiert, um sie schnell dem Kunden zur Verfügung zu stellen. Mit einem immer wachsenden Anforderungskatalog ergaben sich dadurch komplexe Arbeitsabläufe welche im Tool etabliert wurden. Einer dieser komplexen Abläufe ist die Filterung nach Personen oder Abonnemente.

Mit dieser IPA soll die Filterung zwischen diesen zwei Entitäten homogenisiert werden. Um dies zu tun, sollen zuerst zwei bis drei Konzepte ausgearbeitet und anschliessend in einem Variantenentscheid evaluiert werden. Für die Lösungsvariante wird in einem weiteren Schritt ein PoC (Proove of Concept) implementiert.

Nach der IPA soll basierend auf der neuen Filterlogik ein neues UI entworfen werden, um nebst der Ordnung im Backend eine besser User Experience für den Benutzer zu schaffen.

In einer Zeit in welcher Unternehmen mehr den je Wert auf ein sauberes Design und der User Experience von Webseiten und Applikationen geben, das auch in einer älteren Applikation zu etablieren. Gerade bei einem Community-Management Tool wie Hitobito, welches tagtäglich von Personen bedient werden, welche nicht das technische Know-How dahinter besitzen, ist es wichtig Arbeitsabläufe so einfach wie möglich zu entwerfen, um maximale Effizienz für diese Personen zu garantieren. Durch eine Vereinfachung der Hitobito-Fitler machen wir damit einen ersten Schritt in die richtige Richtung.

11 Analyse

In der Analyse der IPA wird der Rahmen geschaffen in welchem man später während des Implementierens arbeitet. Sie befasst sich mit der Aufnahme von Ist- und Zielzustand und definierte Funktionale sowie nicht funktionale Anforderungen. Es wird definiert wo sich die IPA abgrenzt.

11.1 Ist-Zustand

11.1.1 Personenlisten

Aktuell kann ein Nutzer über den Button Neuer-Filter auf die Filter Seite navigieren.

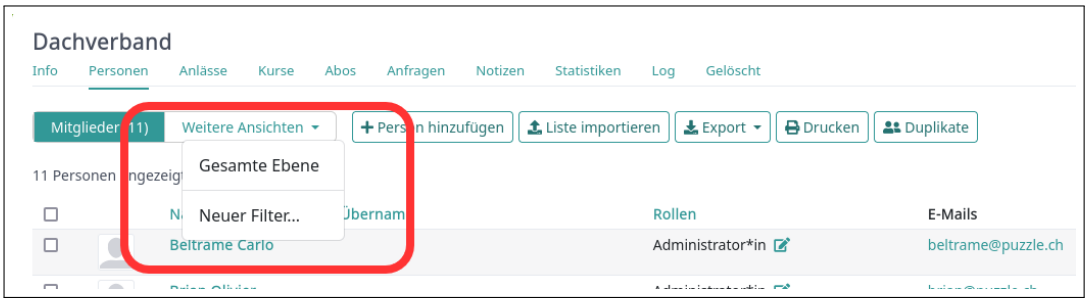

Abbildung 11.1: Hitobito Personenlisten Filtererstellung

Auf dieser definiert er die Filterungskriterien für die Attribute Rollen, Qualifikationen, Felder, Sprache und Tags.

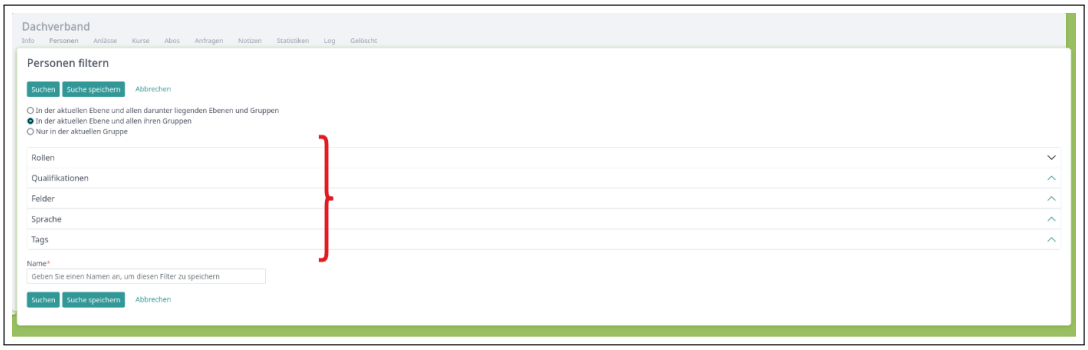

Abbildung 11.2: Hitobito Personenlisten Filterkriterien

Anschliessend ist es dem Nutzer möglich seinen Filter über einen Button für die Wiederverwendung zu speichern.

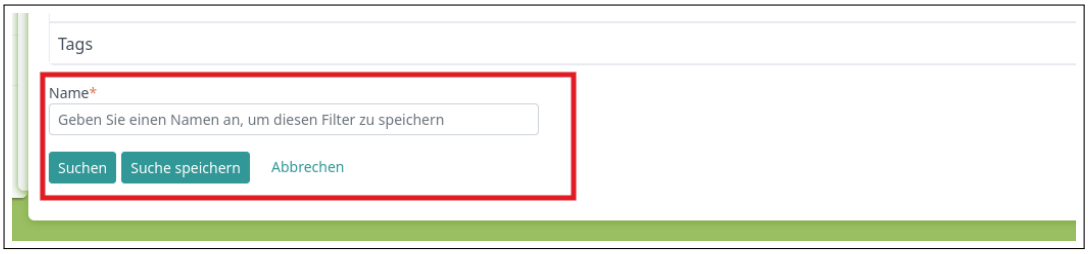

Abbildung 11.3: Hitobito Personenlistenfilter Speicherung

Technisch sind die Personenlisten-Filter nach folgendem Sequenzdiagramm aufgebaut:

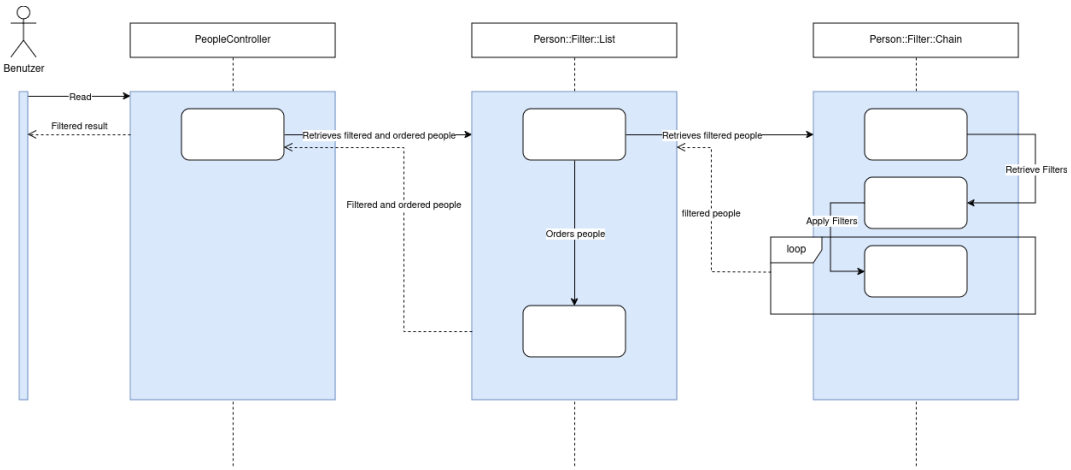

Abbildung 11.4: Sequenzdiagramm Personenlisten Filter

Beschreibung Sequenzdiagramm	
PeopleController	Der PeopleController nimmt den Request des Benutzers entgegen und erwidert die gefilterten und sortierten Personen auf welche der Benutzer zugreifen darf.
Person::Filter::List	Personen auf welche der Benutzer keinen Zugriff hat werden rausgefiltert und anschliessend sortiert.
Person::Filter::Chain	Definiert anhand der Request Parameter vom Controller die Filter und wendet diese in einem Loop via chain-pattern an.

Tabelle 11.1: Beschreibung Sequenzdiagramm

11.1.2 Abonnemente

Auf den Abonnementen kann der Nutzer Filterkriterien in den globalen Bedingungen definieren.

Abbildung 11.5: Hitobito Globale Bedingungen

Unter diesen kann der Nutzer per Dropdown entscheiden für welche Attribute er die Filterkriterien definieren möchte. Er kann auch bereits gesetzte Kriterien entfernen. Pro Filterkriterium entscheidet er im weiteren, mit welcher Genauigkeit nach diesem Filterkriterium gesucht wird. Bei Zahlen ist es möglich die Genauigkeiten ist genau, ist höher als und ist kleiner als einzustellen. Bei Textvergleichen sind es ist genau, enthält, enthält nicht.

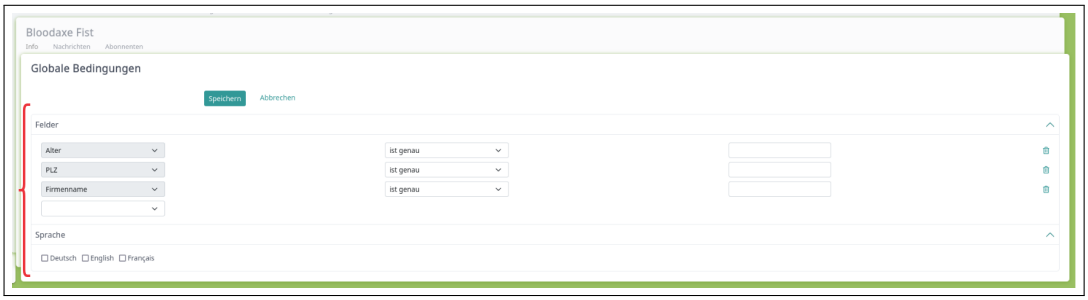

Abbildung 11.6: Hitobito Filterkriterien

Hat der Nutzer seine Filterkriterien und die dazugehörigen Genauigkeiten definiert, kann er sie über den Speicher-Button persistieren. Im Anschluss werden die ausgewählten Filterkriterien in den Globalen Bedingungen angezeigt und ein Success-Alert ausgelöst der die Aktualisierung bestätigt.

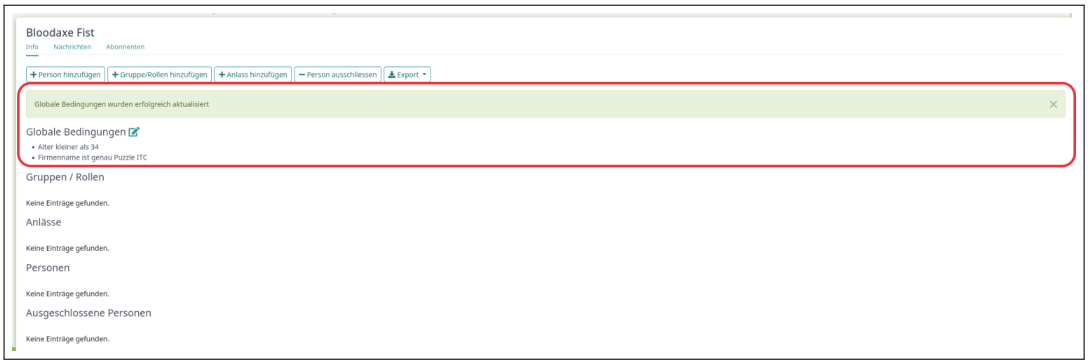

Abbildung 11.7: Hitobito Filterkriterien

Technisch haben sind die Abonnemente nach folgendem ERD aufgebaut.

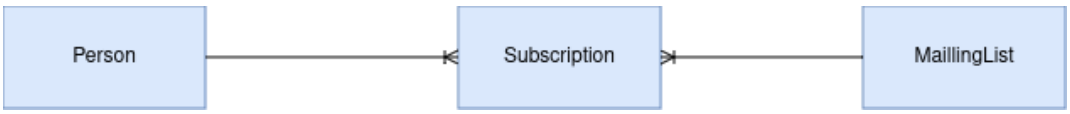

Abbildung 11.8: Hitobito Subscription ERD

Es ist zu beachten: Eine Person kann mehrere Subscriptions besitzen und jede Subscriptions ist einer Mailling list zugerodnet. Somit muss bei der Filterung nach Subscriptions zuerst die MaillingList included werden, damit man auf dieser danach die Filterung ausführen kann.

Die Globalen Filterungskriterien für die Abonnemente werden in dem Modell der MaillingList abgespeichert. Dadurch ergibt sich folgendes Sequenzdiagramm.

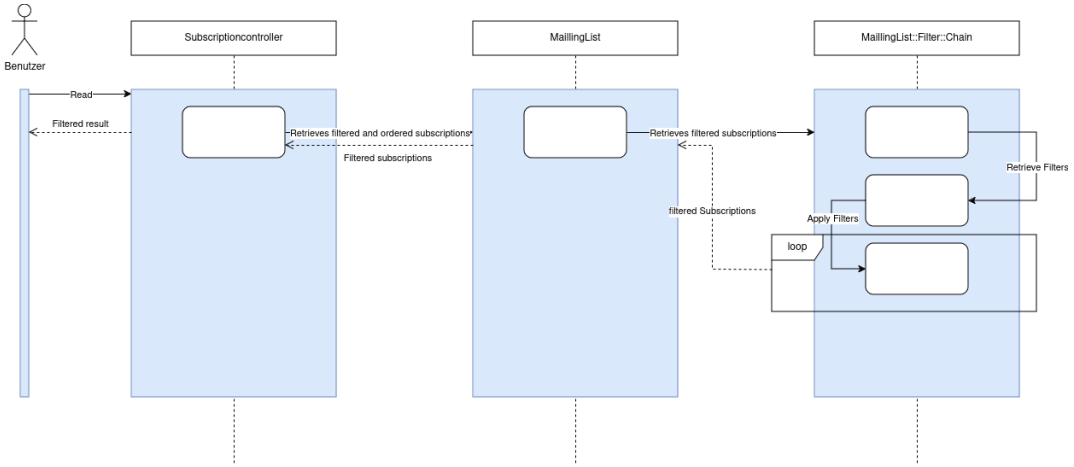

Abbildung 11.9: Hitobito Abonnementen Sequenzdiagramm

Beschreibung Sequenzdiagramm	
SubscriptionController	Der SubscriptionController nimmt den Request des Benutzers entgegen und erwidert die gefilterten und sortierten Personen auf welche der Benutzer zugreifen darf.
MaillingList	Fetcht Subscriptions aus der Datenbank und macht Aufruf zum MaillingList::Filter::Chain
MaillingList::Filter::Chain	Holt die Filterkriterien aus der Datenbank und filtered die Subscriptions danach aus.

Tabelle 11.2: Beschreibung Sequenzdiagramm

11.2 Soll-Zustand

Das neue Konzept für die Abonnementen- und Personenlistenfilterung soll durch den gleichen Prozess laufen. Die beiden Filterungsprozesse sollen dabei zu einem homogenisiert werden und trotzdem die gleichen Funktionalitäten bieten. Bestehende Datenmodelle sollen ebenfalls fusioniert werden.

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 Verzeichnise

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

16 Verzeichnise

16.1 Tabellenverzeichnis

16.2 Abbildungsverzeichnis

16.3 Code Verzeichnis

Quellenverzeichnis

[TODO: Name der Quelle] TODO:URLeinf\protect\unhbox\voidb@x\bgroup\U@D1ex{\setbox\z@\hbox{\char127}\dimen@-.45ex\advance\dimen@\ht\z@}\accent127\fontdimen5\font\U@Du\egroupgen, (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

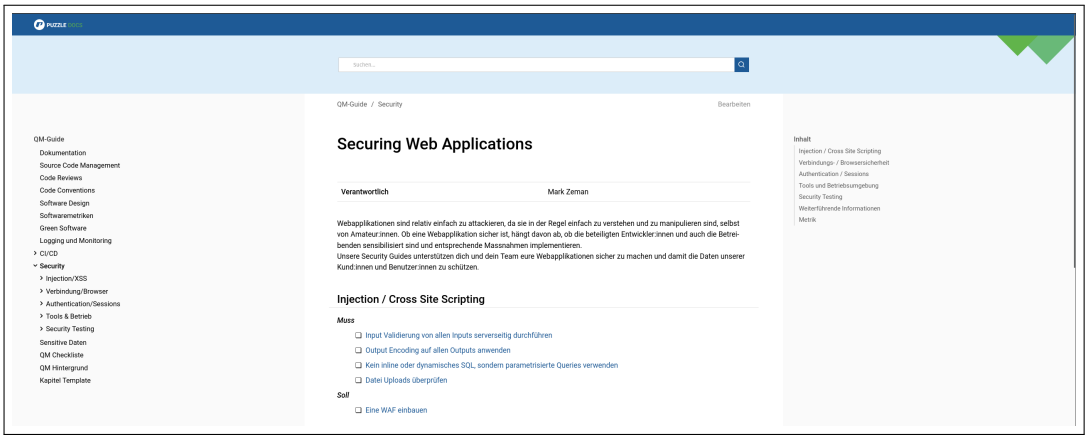

Abbildung 19.2: Puzzle ITC security conventions 1/3

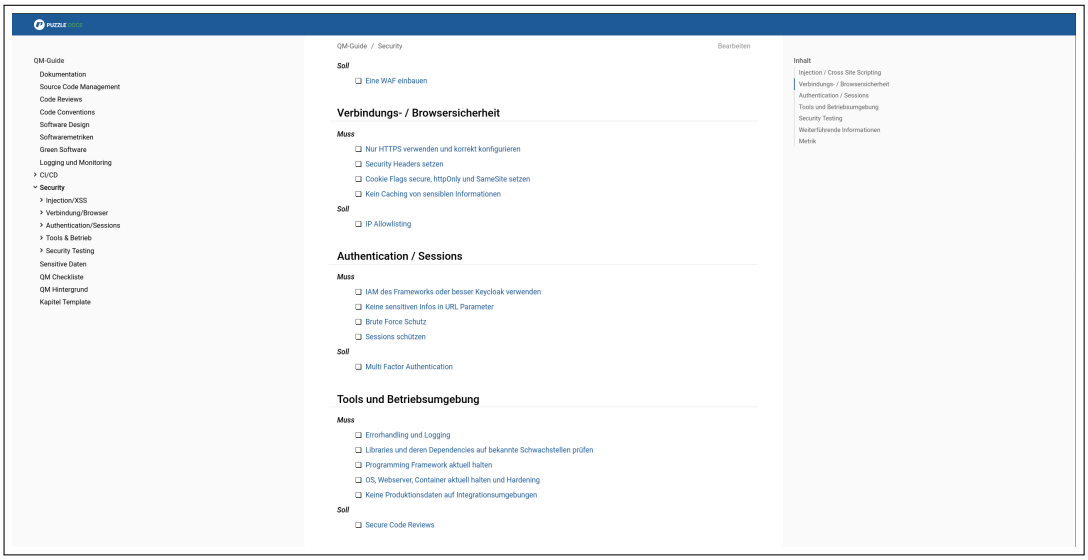

Abbildung 19.3: Puzzle ITC security conventions 2/3

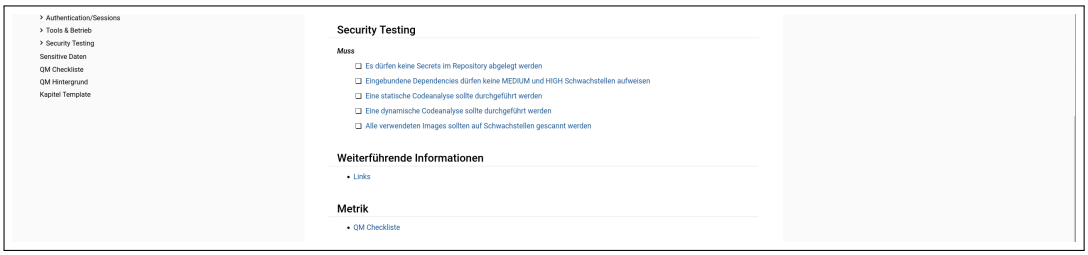

Abbildung 19.4: Puzzle ITC security conventions 3/3

19.4 Datenschutzkonzept