

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

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

Inhaltsverzeichnis

I	Ablauf, Organisation und Umfeld	1
1	Aufgabenstellung	5
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	7
1.4.2	Vorkenntnisse	8
1.4.3	Vorarbeiten	8
1.4.4	Neue Lerninhalte	8
1.4.5	Arbeiten in den letzten 6 Monaten	8
2	Firmenstandards	9
2.1	Code conventions	9
2.1.1	Lizenz	9
2.2	Git conventions	10
2.3	Documentation Conventions	10
3	IPA-Schutzbedarfsanalyse	11
4	Organisation der IPA-Ergebnisse	12
5	Projektmethode	13
6	Projektaufbauorganisation	14
7	Zeitplan	15
7.1	Erläuterung zum Zeitplan	15
8	Arbeitsjournale	16
8.1	Tag 1: TODO: Datum	16
8.2	Tag 2: TODO: Datum	17
8.3	Tag 4: TODO: Datum	18
8.4	Tag 1: TODO: Datum	19
8.5	Tag 5: TODO: Datum	20
8.6	Tag 6: TODO: Datum	21
8.7	Tag 7: TODO: Datum	22

8.8	Tag 8: TODO: Datum	23
8.9	Tag 9: TODO: Datum	24
8.10	Tag 10: TODO: Datum	25
9	Persönliches Fazit	26
II	Projektdokumentation	27
10	Einführung	28
11	Analyse	29
11.1	Ist-Zustand	29
11.2	Soll-Zustand	29
11.3	Persönliche Vorgehensziele	29
11.4	Anforderungen	29
11.4.1	Nicht funktionale Anforderungen	29
11.4.2	Funktionale Anforderungen	29
11.5	Abgrenzung	29
12	Entwurf	30
12.1	Lösungsvarianten	30
12.2	Variantenentscheid	30
12.3	Ausarbeitung	30
13	Ausführung	31
13.1	Testprotokoll	31
14	Einführung	32
15	Sprintabschlüsse	33
15.1	Abschluss Sprint Initialisierung	33
15.2	Abschluss Sprint Umsetzung	33
15.3	Abschluss Sprint Finalisierung	33
III	Anhang und Verzeichnisse	34
16	Verzeichnisse	35
16.1	Tabellenverzeichnis	35
16.2	Abbildungsverzeichnis	35
16.3	Code Verzeichnis	35
	Quellenverzeichnis	36

17	Verwendete Abkürzungen	37
18	Glossar	38
19	Anhänge	39
19.1	Sitzungsprotokolle	39
19.2	Git commit convention	39
19.3	Code	40

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 (<https://github.com/rubocop-hq/ruby-style-guide>) und der Rails Style Guide (<https://github.com/rubocop-hq/rails-style-guide>) gemäss Rubocop Konfiguration des Projekts (<https://github.com/puzzle/cryptopus/blob/master/.rubocop.yml>)

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

4 Organisation der IPA-Ergebnisse

5 Projektmethode

6 Projektaufbauorganisation

7 Zeitplan

7.1 Erläuterung zum Zeitplan

8 Arbeitsjournale

8.1 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 8.1: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- TODO: Hilfestellungen auflisten

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

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
Total		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
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

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
8.1	Tätigkeiten Tag 1	16
8.2	Tätigkeiten Tag 2	17
8.3	Tätigkeiten Tag 4	18
8.4	Tätigkeiten Tag 1	19
8.5	Tätigkeiten Tag 5	20
8.6	Tätigkeiten Tag 6	21
8.7	Tätigkeiten Tag 7	22
8.8	Tätigkeiten Tag 8	23
8.9	Tätigkeiten Tag 9	24
8.10	Tätigkeiten Tag 10	25
13.1	Resultat Testfall 1	31
17.1	Verwendete Abkürzungen	37
18.1	Glossar	38

16.2 Abbildungsverzeichnis

19.1	Puzzle ITC Git commit conventions	39
------	---	----

16.3 Code Verzeichnis

Quellenverzeichnis

[TODO: Name der Quelle] [TODO:URL](#)`inf\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 Code