

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	

Tabelle 1: IPA Daten

Teil I

Ablauf, Organisation und Umfeld

Inhaltsverzeichnis

I	Ablauf, Organisation und Umfeld	1
1	Aufgabenstellung	6
1.1	Titel der Arbeit	6
1.2	Ausgangslage	6
1.3	Detaillierte Aufgabenstellung	7
1.3.1	Mittel und Methoden	9
1.3.2	Vorkenntnisse	10
1.3.3	Vorarbeiten	10
1.3.4	Neue Lerninhalte	10
1.3.5	Arbeiten in den letzten 6 Monaten	10
2	Firmenstandards	11
2.1	Code conventions	11
2.1.1	Mehrsprachigkeit	11
2.1.2	Lizenz	11
2.2	Git Commit Message Conventions	12
3	IPA-Schutzbedarfsanalyse	13
3.1	Datensicherheit	13
3.2	Applikationssicherheit	13
4	Organisation der IPA-Ergebnisse	14
4.1	Datensicherung	14
4.1.1	Dokumentation	14
4.1.2	Code	14
4.1.3	Wiederherstellung des Codes	14
5	Projektmethode	15
5.1	Organisation - Jira	15
5.2	Sprints	15
5.3	Sprint Planning	15
5.4	Daily	15
5.5	Definition of Done	15
5.6	Verwendungsgrund	15
6	Projektaufbauorganisation	16
6.1	Projektrollen in Scrum	16

6.2	Projektrollen IPA	16
6.3	Anpassungen der Projektmethode	16
6.4	Rollenverteilung	16
7	Zeitplan	17
7.1	Erläuterung zum Zeitplan	17
7.2	Sprints	17
8	Arbeitsjournale	18
8.1	Tag 1: Datum	18
8.2	Tag 2: Datum	19
8.3	Tag 3: Datum	20
8.4	Tag 4: Datum	21
8.5	Tag 5: Datum	22
8.6	Tag 6: Datum	23
8.7	Tag 7: Datum	24
8.8	Tag 8: Datum	25
8.9	Tag 9: Datum	26
8.10	Tag 10: Datum	27
8.11	Tag 11: Datum	28
8.12	Tag 12: Datum	29
9	Persönliches Fazit	30
9.1	Was lief weniger gut	30
9.2	Was lief gut	30
9.3	Schlussreflexion	30
II	 Projektdokumentation	31
10	Einführung	32
11	Analyse	33
11.1	Ist-Zustand	34
11.1.1	Personenlisten	34
11.1.2	Abonnemente	34
11.2	Soll-Zustand	34
11.3	Bedürfniserhebung	34
11.4	Risikoanalyse	34
11.5	Anforderungen	34
11.5.1	Nicht funktionale Anforderungen	34
11.5.2	Funktionale Anforderungen	34
11.6	Abgrenzung	34
11.7	Benötigter Rahmen	34
11.7.1	Fehlende Informationen	34
11.8	Persönliche Vorgehensziele	34

12 Entwurf	35
12.1 Anwendungskonzept	36
12.1.1 Anwendungsdiagramm	36
12.1.2 Anwendungsfälle	36
12.2 Systemkonzept	36
12.2.1 Betroffene Services	36
12.2.2 Status quo	36
12.2.3 Lösungsvarianten	36
12.2.4 Variantenentscheid	36
12.3 Sicherheitskonzept	36
12.3.1 SQL-Injection	36
12.3.2 Cross-Site Scripting	36
12.3.3 URL Interpretation	36
12.3.4 Kommunikation HTTP/S	36
12.4 Fehlerbehandlungskonzept	36
12.4.1 Nutzereingabe	36
12.4.2 Laufzeitfehler	36
12.5 Testsetup	36
12.6 Testkonzept	36
12.6.1 Testinfrastruktur	36
12.6.2 Fehlerklassen	36
12.6.3 Manuelle Tests	36
13 Ausführung	37
13.1 Einsatz von KI-Modellen	37
13.2 Gems	37
13.2.1 can-can-can	37
13.2.2 dry-crud	37
14 Einführung	38
14.1 Instruktion	38
14.2 Unvorhergesehene Änderungen	38
14.2.1 application.rb	38
14.2.2 _list.html.haml	38
15 Sprintabschlüsse	39
15.1 Abschluss Sprint Initialisierung	39
15.1.1 Backlog	39
15.2 Abschluss Sprint Umsetzung	39
15.2.1 Backlog	39
15.3 Abschluss Sprint Finalisierung	39
15.3.1 Backlog	39

III	Anhang und Verzeichnisse	40
16	Verzeichnisse	41
16.1	Code	41
16.2	Tabellenverzeichnis	41
16.3	Abbildungsverzeichnis	41
	Quellenverzeichnis	42
17	Verwendete Abkürzungen	43
18	Glossar	44
19	Anhänge	45
19.1	Git Commit Message Convention	45
19.2	Sitzungsprotokolle	45
19.3	Git commit convention	45
19.4	Security conventions	45

1 Aufgabenstellung

1.1 Titel der Arbeit

Hitobito: Neue Generation von Personen-Filtern

1.2 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: [Hitobito](#)

Eine Kernfunktionalität von Hitobito ist das Filtern von Personenlisten und von Mailinglistenempfängern mit konfigurierbaren Filtern. Diese werden über das Webinterface konfiguriert. Das Webinterface wurde mit statischen Webtechnologien entwickelt und ist inzwischen ziemlich in die Jahre gekommen.

Eine Erneuerung dieser Komponente ist ein Wunsch vieler Kunden.

1.3 Detaillierte Aufgabenstellung

Mit dieser IPA soll ein neues UI mit Hotwire für die Persistierung von Filter-Parametern im Hitobito Generic-Wagon erstellt werden (rein Frontend).

- Die Ansichten zur Konfiguration für Filter der Personenlisten und Abonnemente werden mit dem neuen UI ersetzt.
- Die neuen Ansichten werden nach einem gegebenen Mockup umgesetzt. Dieses Mockup wurde vom Kandidaten in Zusammenarbeit mit einem UX Experten erarbeitet und muss als Grundlage für die Ausarbeitung des Interfaces verwendet werden. Des weiteren muss das Interface auf das visuelle Design der existierenden Applikation abgestimmt sein.
- Das Backend darf nicht angepasst werden, das heisst das neue Interface verwendet die bestehenden Endpunkte und schickt die Daten im selben Format wie das alte Interface. Dies muss mit automatisierten Tests sichergestellt werden.
- Formular zur Konfiguration von Personen-Listen Filter: Das bestehende Formular muss ersetzt werden durch eine neue Implementation mit den in Mittel und Methoden definierten Web Technologien. Diese neue Umsetzung muss es erlauben, dynamisch weitere Filterkriterien hinzuzufügen im Gegensatz zur alten Implementation welche mit einem statischen Formular implementiert ist.
- Formular zur Konfiguration von Abo-Empfänger Filter: Das bestehende Formular besteht aus mehreren Teilen, wovon im Rahmen der IPA nur der Teil für die Globalen Filterbedingungen angepasst werden muss. Wie bei den Personen-Listen Filter muss das Formular nun dynamisch implementiert werden. Die Formulare für die weiteren Filterbedingungen werden im Rahmen der IPA nicht angepasst.
- Code der während dieser IPA entsteht soll auf ein privates Github Repo gepushed werden. Die VFs haben dabei stets Lese-Rechte.
- Die Konventionen des Ruby Style Guide, des Rails Style Guide und für Git Commit Messages müssen eingehalten werden (siehe Mittel und Methoden).

Out of Scope - wird erst nach der IPA umgesetzt:

- Filterung für Rollen, Gruppen, Events, People bei Abonnements.
- Anpassungen der Ansicht in den anderen Wagons.
- Anpassungen der bisher bestehenden Tests in Hitobito welche die zu erweiternden Ansichten betreffen.

Weitere Anforderungen zu spezifischen Bewertungskriterien:

- G1: Dokumentation fachlicher und technischer Anforderungen: Die fachlichen und technischen Anforderungen müssen dokumentiert werden.
- G10: Konforme Implementierung und Versionierung: Applikationen und Schnittstellen müssen konform implementiert und versioniert werden.
- A13: Erhebung und Dokumentation der Bedürfnisse und Umfeld: Die Bedürfnisse und das Umfeld werden adäquat erhoben und dokumentiert.
- A15: Instruktion: Es wird für den Projektowner eine Instruktion durchgeführt. Diese muss dem Projektowner die relevanten Änderungen aufzeigen.
- C11: Einsatz von KI-Modellen: Wir setzen bei Puzzle KI in Form von Kopiloten und Chatbots als Hilfsmittel ein. Die Lernenden werden im sinnvollen Einsatz von solcher KI geschult. Dies umfasst z.B. den Umgang in Bezug auf Output Validierung, Transparenz und Sicherheit. Die IPA soll möglichst repräsentativ für unseren Alltag als Entwickler sein, dementsprechen darf KI ein Teil davon sein.
- G5: Risikoanalyse und Sicherheitsmassnahme: Sicherheitsrisiken von Applikationen und Schnittstellen müssen identifiziert und adressiert werden.
- G6: Entwicklung und Anpassung des Anforderungskatalogs: Ein Anforderungskatalog für Sicherheitsmassnahmen von Applikationen und/oder Schnittstellen muss erstellt oder angepasst werden.
- User Experience und visuelles Design: Das Feature muss visuell gut gestaltet sein um die Usability und Nutzerfreundlichkeit des Features sicherzustellen.

- Versionsverwaltung mit Git (Source Code): Die Versionsverwaltung mit Git muss gemäss den Best Practices erfolgen. Es müssen sprechende und einheitliche Commit-Messages geschrieben werden und commit-spezifische Inhalte müssen passend zur Message sein und unter der Einhaltung der Firmenguidelines erfolgen.
- Bewertung von Aussagen: Aussagen in der Arbeit müssen klar zwischen persönlichen Meinungen und auf Quellen basierenden Informationen differenziert werden.

1.3.1 Mittel und Methoden

Technologie und Plattform:

- Ruby, Ruby on Rails, Active Record
- HTML, CSS, Javascript, Hotwire
- PostgreSQL
- Git

Entwicklungsumgebungen:

- IntelliJ
- Visual Studio Code
- Github
- Rake
- Rubocop

Textverarbeitung und Diagramme

- Latex
- draw.io

1.3.2 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.

1.3.3 Vorarbeiten

- Vorbereitung Dokumentvorlage
- Probe-IPA: Vereinheitlichung der Personenlisten- und Abonnementsfilterlogik im Backend
- Entwurf eines Mockups

1.3.4 Neue Lerninhalte

- Eigenständiges Umsetzen eines Designs nach gegebenem Mockup
- Eigenständiges Projektmanagement während der IPA

1.3.5 Arbeiten in den letzten 6 Monaten

- Umsetzung diverser Features und Bugfixes für Hitobito (Ruby on Rails)
- Probe-IPA: Vereinheitlichung der Personenlisten- und Abonnementsfilterlogik
- PostgreSQL Migration Hitobito
- Ruby on Rails Major Upgrade 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 Mehrsprachigkeit

Hitobito ist eine mehrsprachige Applikation. Alle Erweiterungen oder Anpassungen müssen in Deutsch übersetzt werden. Übersetzungen werden in einer Übersetzungsdatei gespeichert oder können vom Kunden in einem Tool namens Transifex verwaltet werden.

2.1.2 Lizenz

Hitobito ist ein Open Source Projekt. In jedem File in Hitobito wird das Copyright für den jeweiligen Kunden in Kommentarform beschrieben. Diese Lizenz- und Kundeninformationen können über folgenden Befehl eingefügt werden:

```
rake license:insert
```

Die daraus entstehende Lizenz sieht wie folgt aus:

```
1  # Copyright (c) 2012 -2021 , hitobito AG . This file is part of
2  # hitobito and licensed under the Affero General Public License version 3
3  # or later . See the COPYING file at the top - level directory or at
4  # https :// github . com / hitobito / hitobito .
```

Alternativ dazu können diese Informationen mit

```
rake license:remove
```

entfernt oder mit

```
rake license:update
```

aktualisiert werden.

2.2 Git Commit Message Conventions

Die Git Commit Messages werden nach den Regeln von Puzzle ITC formuliert. Im Anhang unter Git Commit Message Conention finden sie eine Kopie der Firmenkonventionen. Diese wurden basierend auf folgendem Tutorial definiert: [Tutorial](#)

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

3 IPA-Schutzbedarfanalyse

3.1 Datensicherheit

3.2 Applikationssicherheit

4 Organisation der IPA-Ergebnisse

4.1 Datensicherung

4.1.1 Dokumentation

4.1.2 Code

4.1.3 Wiederherstellung des Codes

5 Projektmethode

5.1 Organisation - Jira

5.2 Sprints

5.3 Sprint Planning

5.4 Daily

5.5 Definition of Done

5.6 Verwendungsgrund

6 Projektaufbauorganisation

6.1 Projektrollen in Scrum

6.2 Projektrollen IPA

6.3 Anpassungen der Projektmethode

6.4 Rollenverteilung

7 Zeitplan

7.1 Erläuterung zum Zeitplan

7.2 Sprints

8 Arbeitsjournale

8.1 Tag 1: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.1: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.2 Tag 2: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.2: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.3 Tag 3: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.3: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.4 Tag 4: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.4: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.5 Tag 5: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.5: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.6 Tag 6: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.6: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.7 Tag 7: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.7: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.8 Tag 8: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.8: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.9 Tag 9: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.9: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.10 Tag 10: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamtstunden soll	Gesamtstunden ist

Tabelle 8.10: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.11 Tag 11: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.11: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

8.12 Tag 12: Datum

Tätigkeiten	Beteiligte Personen	Aufwand Geplant (std)	Aufwand Effektiv (std)
Tätigkeit	Personen	Stunden soll	Stunden ist
Total		Gesamt-stunden soll	Gesamt-stunden ist

Tabelle 8.12: Tätigkeiten Tag 1

Tagesablauf

Hilfestellungen

- Person: Hilfestellung

Reflexion

Was lief gut

Was lief weniger gut

Meine Erkenntnisse von heute

Nächste Schritte

9 Persönliches Fazit

9.1 Was lief weniger gut

9.2 Was lief gut

9.3 Schlussreflexion

Teil II

Projektdokumentation

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

10 Einführung

11 Analyse

11.1 Ist-Zustand

11.1.1 Personenlisten

11.1.2 Abonnemente

11.2 Soll-Zustand

11.3 Bedürfniserhebung

11.4 Risikoanalyse

11.5 Anforderungen

11.5.1 Nicht funktionale Anforderungen

11.5.2 Funktionale Anforderungen

11.6 Abgrenzung

11.7 Benötigter Rahmen

11.7.1 Fehlende Informationen

11.8 Persönliche Vorgehensziele

12 Entwurf

12.1 Anwendungskonzept

12.1.1 Anwendungsdiagramm

12.1.2 Anwendungsfälle

12.2 Systemkonzept

12.2.1 Betroffene Services

12.2.2 Status quo

12.2.3 Lösungsvarianten

12.2.4 Variantenentscheid

12.3 Sicherheitskonzept

12.3.1 SQL-Injection

12.3.2 Cross-Site Scripting

12.3.3 URL Interpretation

12.3.4 Kommunikation HTTP/S

12.4 Fehlerbehandlungskonzept

Version 1.0

4. März 2025

Seite 36 von 45

12.4.1 Nutzereingabe

12.4.2 Laufzeitfehler

13 Ausführung

13.1 Einsatz von KI-Modellen

13.2 Gems

13.2.1 can-can-can

13.2.2 dry-crud

14 Einführung

14.1 Instruktion

14.2 Unvorhergesehene Änderungen

14.2.1 application.rb

14.2.2 _list.html.haml

15 Sprintabschlüsse

15.1 Abschluss Sprint Initialisierung

15.1.1 Backlog

15.2 Abschluss Sprint Umsetzung

15.2.1 Backlog

15.3 Abschluss Sprint Finalisierung

15.3.1 Backlog

Teil III

Anhänge und Verzeichnisse

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

16 Verzeichnisse

16.1 Code

16.2 Tabellenverzeichnis

1	IPA Daten	1
8.1	Tätigkeiten Tag 1	18
8.2	Tätigkeiten Tag 1	19
8.3	Tätigkeiten Tag 1	20
8.4	Tätigkeiten Tag 1	21
8.5	Tätigkeiten Tag 1	22
8.6	Tätigkeiten Tag 1	23
8.7	Tätigkeiten Tag 1	24
8.8	Tätigkeiten Tag 1	25
8.9	Tätigkeiten Tag 1	26
8.10	Tätigkeiten Tag 1	27
8.11	Tätigkeiten Tag 1	28
8.12	Tätigkeiten Tag 1	29
17.1	Verwendete Abkürzungen	43
18.1	Glossar	44

16.3 Abbildungsverzeichnis

19.1	Puzzle ITC Git commit conventions	45
------	---	----

Quellenverzeichnis

[Github Docs - Understanding connections between repositories]

[https://docs.github.com/en/repositories/
viewing-activity-and-data-for-your-repository/
understanding-connections-between-repositories](https://docs.github.com/en/repositories/viewing-activity-and-data-for-your-repository/understanding-connections-between-repositories), (04.03.2025)

[Github Docs - Configuring issue templates] [https://docs.github.com/en/
communities/using-templates-to-encourage-useful-issues-and-pull-requests/
configuring-issue-templates-for-your-repository](https://docs.github.com/en/communities/using-templates-to-encourage-useful-issues-and-pull-requests/configuring-issue-templates-for-your-repository), (04.03.2025)

[Leo - Translating] <https://dict.leo.org/german-english>, (04.03.2025)

17 Verwendete Abkürzungen

Abkürzung	Bedeutung
UML	Unified Modeling Language

Tabelle 17.1: Verwendete Abkürzungen

18 Glossar

Bezeichnung	Bedeutung
Hitobito	Community Management Tool

Tabelle 18.1: Glossar

19 Anhänge

19.1 Git Commit Message 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

19.2 Sitzungsprotokolle

19.3 Git commit convention

19.4 Security conventions