

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

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	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
2	Firmenstandards	10
2.1	Code conventions	10
2.1.1	Lizenz	10
2.2	Git conventions	11
2.3	Documentation Conventions	11
3	IPA-Schutzbedarfsanalyse	12
3.1	Datensicherheit	12
3.2	Applikationssicherheit	12
4	Organisation der IPA-Ergebnisse	14
5	Projektmethode	15
5.1	Github Board	15
5.1.1	Backlog	15
5.1.2	Refinement	15
5.1.3	Sprint Backlog	16
5.1.4	In Progress	16
5.1.5	Done	16
5.2	Sprints	16
5.3	Sprint Planning	17
5.4	Daily	17
5.5	Verwendungsgrund	17

6	Projektaufbauorganisation	18
7	Zeitplan	19
7.1	Erläuterung zum Zeitplan	19
8	Arbeitsjournale	20
8.1	Tag 1: TODO: Datum	20
8.2	Tag 2: TODO: Datum	21
8.3	Tag 4: TODO: Datum	22
8.4	Tag 1: TODO: Datum	23
8.5	Tag 5: TODO: Datum	24
8.6	Tag 6: TODO: Datum	25
8.7	Tag 7: TODO: Datum	26
8.8	Tag 8: TODO: Datum	27
8.9	Tag 9: TODO: Datum	28
8.10	Tag 10: TODO: Datum	29
9	Persönliches Fazit	30
II	 Projektdokumentation	31
10	Einführung	32
11	Analyse	33
11.1	Ist-Zustand	33
11.2	Soll-Zustand	33
11.3	Persönliche Vorgehensziele	33
11.4	Anforderungen	33
11.4.1	Nicht funktionale Anforderungen	33
11.4.2	Funktionale Anforderungen	33
11.5	Abgrenzung	33
12	Entwurf	34
12.1	Lösungsvarianten	34
12.2	Variantenentscheid	34
12.3	Ausarbeitung	34
13	Ausführung	35
13.1	Testprotokoll	35
14	Einführung	36
15	Sprintabschlüsse	37
15.1	Abschluss Sprint Initialisierung	37
15.2	Abschluss Sprint Umsetzung	37
15.3	Abschluss Sprint Finalisierung	37

III	Anhang und Verzeichnisse	38
16	Verzeichnisse	39
16.1	Tabellenverzeichnis	39
16.2	Abbildungsverzeichnis	39
16.3	Code Verzeichnis	39
	Quellenverzeichnis	40
17	Verwendete Abkürzungen	41
18	Glossar	42
19	Anhänge	43
19.1	Sitzungsprotokolle	43
19.2	Git commit convention	43
19.3	Security conventions	43
19.4	Code	44

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

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 Srintes 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 Srintes 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: 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	20
8.2	Tätigkeiten Tag 2	21
8.3	Tätigkeiten Tag 4	22
8.4	Tätigkeiten Tag 1	23
8.5	Tätigkeiten Tag 5	24
8.6	Tätigkeiten Tag 6	25
8.7	Tätigkeiten Tag 7	26
8.8	Tätigkeiten Tag 8	27
8.9	Tätigkeiten Tag 9	28
8.10	Tätigkeiten Tag 10	29
13.1	Resultat Testfall 1	35
17.1	Verwendete Abkürzungen	41
18.1	Glossar	42

16.2 Abbildungsverzeichnis

19.1	Puzzle ITC Git commit conventions	43
19.2	Puzzle ITC security conventions 1/3	43
19.3	Puzzle ITC security conventions 2/3	44
19.4	Puzzle ITC security conventions 3/3	44

16.3 Code Verzeichnis

Quellenverzeichnis

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

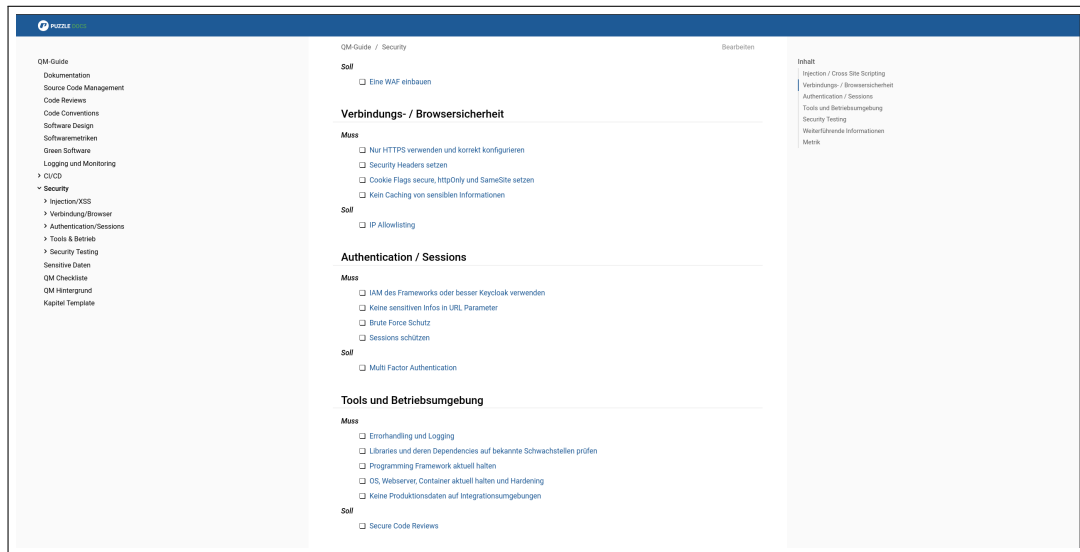


Abbildung 19.3: Puzzle ITC security conventions 2/3

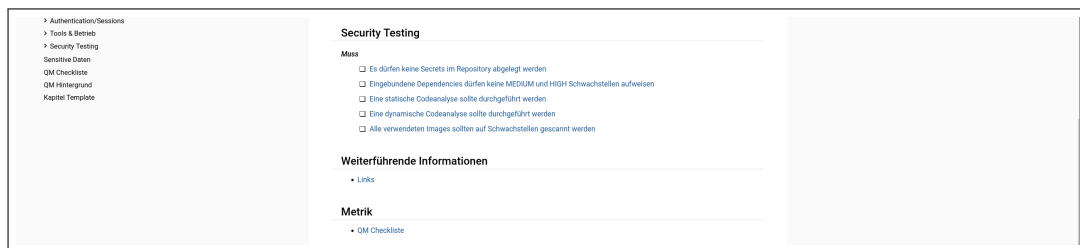


Abbildung 19.4: Puzzle ITC security conventions 3/3

19.4 Code