



# IPA

NAME DER IPA

IPA von Niculin Steiner

Ergon Informatik AG  
7. November 2024

# Inhalt

<b>I</b>	<b>Umfeld und Ablauf</b>	<b>3</b>
<b>1</b>	<b>Aufgabenstellung</b>	<b>4</b>
1.1	Ausgangslage . . . . .	4
1.2	Detaillierte Aufgabenstellung . . . . .	5
1.3	Mittel und Methoden . . . . .	6
1.4	Vorkenntnisse . . . . .	6
1.5	Vorarbeiten . . . . .	6
1.6	Neue Lerninhalte . . . . .	7
1.7	Arbeiten in den letzten 6 Monaten . . . . .	7
<b>2</b>	<b>Projektaufbauorganisation</b>	<b>8</b>
<b>3</b>	<b>Benützte Firmenstandards</b>	<b>9</b>
<b>4</b>	<b>Arbeitsumgebung</b>	<b>10</b>
4.1	Arbeitsplatz . . . . .	10
4.2	Verwendete Tools . . . . .	11
<b>5</b>	<b>Versionierung und Sicherung der Arbeitsergebnisse</b>	<b>12</b>
5.1	Git als Versionierungstool . . . . .	12
5.2	Git im Zusammenspiel mit Gerrit . . . . .	12
<b>6</b>	<b>Projektmanagementmethode</b>	<b>13</b>
6.1	IPERKA . . . . .	13
6.1.1	Informieren . . . . .	13
6.1.2	Planen . . . . .	13
6.1.3	Entscheiden . . . . .	13
6.1.4	Realisieren . . . . .	13
6.1.5	Kontrollieren . . . . .	13
6.1.6	Auswerten . . . . .	14
6.2	Alternative Methode - Scrum . . . . .	14
<b>7</b>	<b>Arbeitsprotokoll</b>	<b>15</b>
<b>8</b>	<b>Zeitplan</b>	<b>18</b>

<b>II Projekt</b>	<b>21</b>
<b>9 Informieren</b>	<b>22</b>
9.1 Analyse . . . . .	22
9.2 Technische Referenzen . . . . .	22
9.3 Anforderungen . . . . .	23
9.3.1 Rest / Backend . . . . .	23
9.3.2 SPA . . . . .	23
9.3.3 Kundendokumentation . . . . .	24
<b>10 Planen</b>	<b>25</b>
10.1 Arbeitspakete . . . . .	25
10.1.1 Informieren . . . . .	25
10.1.2 Planen . . . . .	26
10.1.3 Entscheiden . . . . .	27
10.1.4 Realisieren . . . . .	28
10.1.5 Kontrollieren . . . . .	29
10.1.6 Auswerten . . . . .	30
10.1.7 Rahmenaufgaben . . . . .	30
<b>11 Entscheiden</b>	<b>32</b>
<b>12 Realisieren</b>	<b>33</b>
<b>13 Kontrollieren</b>	<b>34</b>
<b>14 Auswerten</b>	<b>35</b>
<b>Abbildungsverzeichnis</b>	<b>35</b>

# Teil I

## Umfeld und Ablauf

# 1 Aufgabenstellung

In diesem Kapitel sind die Aufgabenstellung und die Rahmenbedingungen aufgeführt. Der grösste Teil des Inhalts stammt aus der originalen Aufgabenstellung.

## 1.1 Ausgangslage

Airlock Identity and Access Management (IAM) ist ein bestehendes, in unserer Abteilung entwickeltes Produkt, das unter anderem Logins (Authentisierungen) ermöglicht. Eine weitere Funktionalität eines IAMs ist der Admin-Bereich (adminapp). Airlock IAM unterstützt unterschiedliche Stufen von Administratoren, um beispielsweise Mitarbeitenden im Support oder an einem Kundenshalter spezifisch eingeschränkten Zugriff für die Verwaltung von Usern zu erlauben.

Airlock 2FA erlaubt es, nebst beispielsweise Usernamen und Passwort, einen weiteren Authentisierungsfaktor zu verwenden. Üblicherweise wird dazu die Airlock 2FA App auf dem Smartphone installiert und aktiviert.

Es gibt mehrere Möglichkeiten, wie Kund:innen für den eigenen Login die Airlock 2FA aktivieren können. Ein Weg ist beispielsweise über einen Brief mit einem QR Code, welchen Kund:innen dann mit der Airlock 2FA App scannen können. Die Aktivierung ist auch über einen 16-stelligen Aktivierungscode möglich.

Immer wieder kommt es vor, dass Kund:innen Unterstützung bei der Aktivierung von Airlock 2FA benötigen und sich telefonisch beim Firmen-Helpdesk oder am physischen Schalter melden. Damit das Support- oder Schalterpersonal der Kundschaft helfen kann, die Airlock 2FA zu aktivieren, braucht es eine Möglichkeit, den 16-stelligen Aktivierungscode für den spezifischen User anzuzeigen.

Bisher gibt es in Airlock IAM noch kein Feature, damit der Administrator-Bereich solche 16-stelligen Aktivierungscodes pro User anzeigen kann.

## 1.2 Detaillierte Aufgabenstellung

### Ziele

- UC1: Helpdesk kann Kunden am Telefon helfen, ein Gerät zu aktivieren.
- UC2: Schaltermitarbeiter kann Kunde am Schalter helfen, ein Gerät zu aktivieren
- UC3: Es soll möglich sein, den Zugriff auf die userspezifischen 16-stelligen Aktivierungscodes nur für bestimmte Administratoren-Rollen (bspw. Rolle Helpdesk) freizugeben, damit nicht alle Administratoren sich den 16-stelligen Aktivierungscode anzeigen lassen können.
- UC4: Im User Activities Logfile des spezifischen Users soll geloggt werden, welcher Administrator-Account zu welchem Zeitpunkt den 16-stelligen Aktivierungscode angezeigt hat, damit im Nachhinein nachvollziehbar ist, welche Administratoren je Zugriff auf den Aktivierungscode hatten.

### Weitere Anforderungen

- Der Code soll auf Knopfdruck in der Adminapp angezeigt werden. Dabei sind UI-Komponenten zu verwenden, die an anderen Stellen in der Adminapp auch schon verwendet werden. Eine mögliche Lösung ist ein SPA Popup (kein Browser Popup) mit einem 'Schliessen' Knopf.
- Neue Plugins oder Plugin Properties sollen einen klaren und vollständigen Hilfetext haben.

### Erwartete Artefakte

Nebst der IPA Dokumentation werden diese technischen Artefakte erwartet:

- Sinnvolles Slicing und Anzahl von Gerrit Changes mit der implementierten Lösung und Git Kommentaren, die unseren internen Konventionen entsprechen. Der Kandidat entscheidet selbst, wie viele Gerrit Changes sinnvoll sind. Er hat dabei zu beachten, dass die Changes aufeinander aufbauen sollten und «verdaubare» Review-Größen haben.
- Beschreibung wie das neue Feature konfiguriert werden kann in der Airlock IAM Kundendokumentation. Dazu soll das Kapitel 18.5 Airlock 2FA configuration sinnvoll erweitert werden. Die angepasste Kundendokumentation soll auf Englisch und in den restlichen PDF-Unterlagen enthalten sein (es ist nicht nötig, mit unserem Kundendokumentation-Tool SMC zu arbeiten).

## **Abgrenzung**

- Administratoren können pro User bereits Aktivierungsbriefe erstellen oder anfordern. An dieser Logik soll im Rahmen dieses Issues nichts verändert oder erweitert werden.

## **1.3 Mittel und Methoden**

Es wird auf dem aktuellen Stand der Entwicklung von Airlock IAM 8.4 aufgebaut.

### **REST Technologien**

- Java(Guice als Dependency Injection Framework), JSON, JUnit
- Jackson, Jersey, Guice
- REST Integration Tests

### **SPA Technologien**

- Angular (Typescript/RXJS)
- Bootstrap (HTML/CSS/SASS)
- Selenium UI Testing

### **Wichtigste Tools**

- IntelliJ(IDE)
- Gerrit + Git (SCM)

## **1.4 Vorkenntnisse**

Der Kandidat war involviert in die Implementation von SPA und REST Features im Bereich IAM Protected Self-Service.

Das Grundgerüst der SPA und REST Endpunkte ist bekannt.

## **1.5 Vorarbeiten**

Der Kandidat hat für die Probe-IPA keine vorbereiteten Tätigkeiten erarbeitet, hat sich aber in das Thema Airlock 2FA eingelesen.

## 1.6 Neue Lerninhalte

Erfahrung bei der selbständigen Entwicklung einer produktrelevanten Erweiterung unter realistischen Bedingungen.

- Futurae API: <https://www.futurae.com/docs/api/auth/>
- IAM Kunden Dokumentation: <https://docs.airlock.com/iam/8.3/>

## 1.7 Arbeiten in den letzten 6 Monaten

In den letzten sechs Monaten hat der Kandidat Erfahrungen in folgenden Bereichen gesammelt:

- OAuth 2.0 / OpenID Connect Consent Management Self-Service, SPA und REST
- Have I Been Pwnd Scriptable Step, 3rd Party REST API, Lua
- HTTP Cache Control Konfiguration im Zusammenhang mit JWKS REST End-point



## 2 Projektaufbauorganisation

Die folgenden Personen sind in dieses Projekt involviert:

Person	Rolle	Aufgabe/Verantwortung
Niculin Steiner	Kandidat (K)	Umsetzen der Facharbeit
Pascal Knecht	Verantwortliche Fachkraft (VF)	Facharbeit begleiten, technische Fragen beantworten, Bewertung der Facharbeit
Bernd Lienberger	Hauptexperte (HEX)	IPA bezogene Fragen beantworten, Entscheiden bei auftretenden Problemen, Besuchstermine festlegen, Fachgespräch leiten, Bewertung der Facharbeit

### 3 Benützte Firmenstandards

## 4 Arbeitsumgebung

In diesem Abschnitt wird der Arbeitsplatz und die Umgebung, während der Probe-IPA, des Kandidaten beschrieben.

### 4.1 Arbeitsplatz

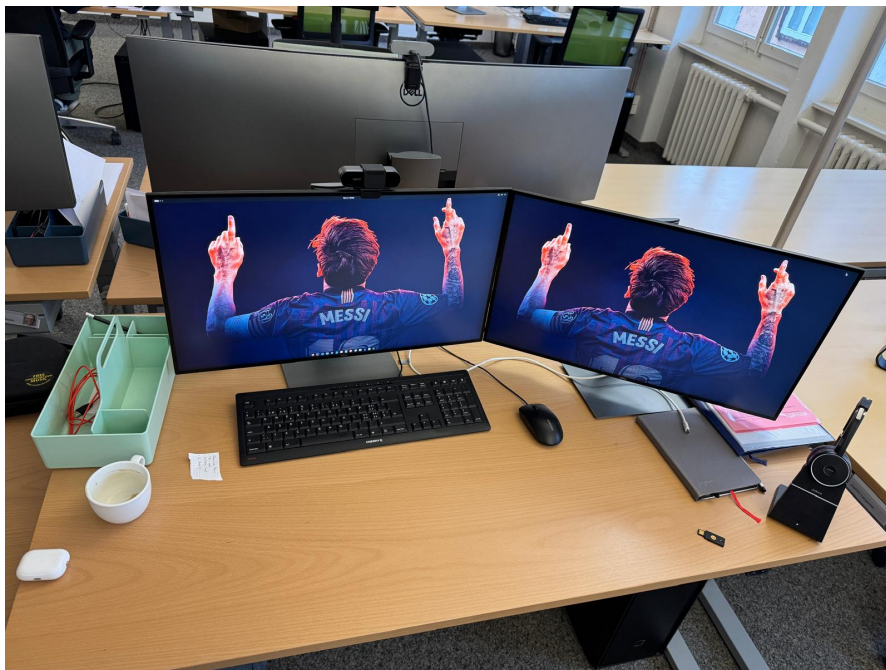


Abbildung 4.1: Arbeitsplatz während der Probe-IPA

Da seit der Mitarbeit im IAM nie im Homeoffice gearbeitet wurde, findet auch die Probe-IPA wie gewohnt vor Ort statt. Der Desktop PC mit dem Betriebssystem Linux (Distribution Ubuntu) ist für die maximale Effizienz mit 2 Bildschirmen verbunden. Um im Grossraumbüro möglichst ungestört zu arbeiten, liegen dem Kandidaten ein Paar AirPods Pro mit Noise Cancelling vor.

## 4.2 Verwendete Tools

Die folgende Tabelle zeigt einen Überblick der wichtigsten Tools, welche für die Umsetzung der Probe-IPA verwendet wurden:

Tool	Einsatzzweck	Link
Intellij Ultimate	Entwicklungsumgebung, zur Entwicklung des Features	<a href="https://www.jetbrains.com/de-de/idea/">https://www.jetbrains.com/de-de/idea/</a>
TeXstudio	Entwicklungsumgebung für Latex, mit welchem die Dokumentation geschrieben wurde	<a href="https://www.texstudio.org/">https://www.texstudio.org/</a>
Gerrit	Quellcode Verwaltung	<a href="https://www.gerritcodereview.com/">https://www.gerritcodereview.com/</a>
Git	Versionskontrollsystem	<a href="https://git-scm.com/">https://git-scm.com/</a>
Jenkins	Automatisierte Testruns	<a href="https://www.jenkins.io/">https://www.jenkins.io/</a>
Outlook	Termin für den Expertenbesuch aufsetzen	<a href="https://www.microsoft.com/de-ch/microsoft-365/outlook">https://www.microsoft.com/de-ch/microsoft-365/outlook</a>

## 5 Versionierung und Sicherung der Arbeitsergebnisse

Die Arbeitsergebnisse sollten gesichert werden. Damit, im Falle eines unerwarteten Ausfalls während der Probe-IPA, z.B des Rechners, von einem anderen Gerät wieder auf den Stand zugegriffen werden kann. Zu dem sollte es generell möglich sein jeder Zeit auf einen älteren Stand zurück zukommen.

### 5.1 Git als Versionierungstool

Für die Versionierung der Arbeitsergebnisse wurde Git verwendet. Git ist weit verbreitet und ist auch aus der Schule und diversen anderen Projekten bekannt. Es wird verwendet um Änderungen am Code zu verfolgen und erstellt dabei eine Versionshistorie.

### 5.2 Git im Zusammenspiel mit Gerrit

Der Quellcode liegt in einem Git-Repository auf Gerrit. Gerrit dient dazu als Review und Code Management Tool. Im Vergleich zur «gewöhnlichen» Entwicklung mit Git, bei der man für neue Features Branches und Commits erstellt arbeitet man bei Gerrit sozusagen auf Commitbasis. Pusht man einen neuen Commit auf Gerrit, erstellt dieser ein neues «Changeset» mit einem Patchset. Gibt es nun weitere Änderungen werden diese einfach Amandet, dies erstellt dann ein weilers Patchset in diesem Changeset. Für grössere und komplexere Änderungen können auch aufeinander aufbauende Changesets erstellt werden.

## 6 Projektmanagementmethode

In diesem Kapitel wird die Projektmanagementmethode IPERKA beschrieben. Es wird dargelegt wieso diese Methode gewählt wurde und was die Vor und/oder Nachteile daran sind.

### 6.1 IPERKA

Für die Probe-VA wurde IPERKA als Projektmanagementmethode gewählt. Sie eignet sich gut für kleine Projekte. Sie lassen sich damit einfach und strukturiert planen sowie umsetzen. Die IPERKA Methode setzt sich aus folgenden 6 Schritten zusammen:

#### 6.1.1 Informieren

Der erste Punkt bei IPERKA ist das Informieren. Dabei wird sich ein Überblick über das Projekt / den Projektauftrag verschaffen. Es gilt zu klären was genau der Auftrag ist, und ob alle Informationen vorhanden sind.

#### 6.1.2 Planen

Als zweiten Schritt kommt das Planen. Hier wird das Projekt konkreter und es wird ein Zeitplan erstellt. Und je nach Team größe, werden bestimmte Aufgaben zugeteilt. Im Probe-IPA Fall fällt dies natürlich weg.

#### 6.1.3 Entscheiden

Beim Entscheiden, wird entschieden welchen Lösungsweg gegangen werden soll. Es wird z.B definiert mit welchen Tools / Technologien gearbeitet wird. Wichtig ist auch, dass die Kriterien, welche zu dieser Entscheidung geführt haben, definiert werden.

#### 6.1.4 Realisieren

In diesem Teil geht es an die Umsetzung. Das Projekt wird nach dem definierten Plan sowie Zeitplan versucht umzusetzen.

#### 6.1.5 Kontrollieren

Der fünfte Schritt erfolgt teilweise parallel zum Vierten. In diesem Schritt wird von oben auf das laufende Projekt geblickt und geschaut, ob alles nach Plan läuft. Gibt es Abweichungen und falls ja, können diese begründet werden?

### **6.1.6 Auswerten**

Der letzte Schritt dient dazu, nochmals auf das Projekt zurückzublicken und es zu Reflektieren.

## **6.2 Alternative Methode - Scrum**

Nebst IPERKA gibt es auch noch andere Alternativen. Eine davon ist Scrum. Scrum eignet sich allerdings nicht besonders für die Umsetzung eines Projekts wie die Probe-IPA. Sie ist eine Agile Projektmanagementmethode, welche sich für Projekte eignet, die sehr dynamisch und doch komplex sind. Meistens sind die konkreten Anforderungen zu Beginn sogar noch unklar. Zudem kann Scrum nur teilweise alleine durchgeführt werden. Dies ist bei der Probe-IPA nicht der Fall. Deshalb wurde sich für IPERKA entschieden.

## 7 Arbeitsprotokoll

<b>Datum</b>	06.11.2024
<b>Bearbeitete Arbeitspakete</b>	1.1, 1.2, 2.1, 2.2, 7.1, 7.2, 7.3
<b>Arbeitszeit</b>	8h
<b>Überzeit</b>	0
<b>Vergleich mit dem Zeitplan</b>	Da ich den Zeitplan noch nicht fertig erstellt habe, kann ich für heute keinen Vergleich ziehen.
<b>Erfolge und Probleme</b>	Zu Beginn wusste ich nicht genau wie ich am besten vorgehe resp. was ich zuerst angehe, da hat mir das vorhandene Template einen sehr guten Leitfaden gegeben. Und so habe ich begonnen alles der Reihe nach auszufüllen/ zu dokumentieren. Und bin am Schluss weiter gekommen als gedacht.
<b>Tagesreflexion</b>	Heute bin ich sehr gut voran gekommen. Ich konnte bereits den Teil 1 der Dokumentation abschliessen und mit den Arbeitspaketen beginnen.
<b>In Anspruch genommene Hilfe</b>	Fragen an Pascal bezüglich der Aufgabenstellung. War mir unsicher, wo genau die Kundendoku hin muss. Jetzt weiss ich, dass es reicht, wenn ich sie im Anhang anhänge.



<b>Datum</b>	...
<b>Bearbeitete Arbeitspakete</b>	...
<b>Arbeitszeit</b>	...
<b>Überzeit</b>	...
<b>Vergleich mit dem Zeitplan</b>	...
<b>Erfolge und Probleme</b>	...
<b>Tagesreflexion</b>	...
<b>In Anspruch genommene Hilfe</b>	...

<b>Datum</b>	...
<b>Bearbeitete Arbeitspakete</b>	...
<b>Arbeitszeit</b>	...
<b>Überzeit</b>	...
<b>Vergleich mit dem Zeitplan</b>	...
<b>Erfolge und Probleme</b>	...
<b>Tagesreflexion</b>	...
<b>In Anspruch genommene Hilfe</b>	...

## 8 Zeitplan

Die folgenden 2 Seiten beinhalten den Zeitplan. Er soll für die 2 Wochen einen groben leitfaden sein. Der Zeitplan ist dargestellt in einem GANT-Diagramm. In diesem werden 2h Blöcke verwendet.

[illegible]

[illegible]

# Teil II

## Projekt

## 9 Informieren

In diesem Kapitel geht es um die erste von 6 Phasen der IPERKA-Methode, dem Informieren. Es bietet Platz um aufzuzeigen, was während dieser Phase unternommen wurde.

### 9.1 Analyse

Als erster Schritt wurde die Aufgabe analysiert und einen Überblick verschaffen.

#### Auftrag

Umzusetzen ist eine Funktion in der Admin App der Airlock IAM Applikation, welche es den Admin nutzern ermöglicht die 16 stelligen Airlock 2FA Aktivierungscode der Nutzer anzeigen zu lassen. Dies hilft Ihnen, die Endnutzer bei der Aktivierung der Airlock 2FA zu unterstützen. Der Aktivierungscode sollte per Konpfdruck angezeigt werden können, zum Beispiel als Popup. Dies ist jedoch noch zu evaluieren, vielleicht bieten sich auch noch andere Optionen an. Sicher ist, dass der Code in im Airlock 2FA Management angezeigt werden soll und nur, falls durch den Admin gewollt.

Weiter soll es möglich sein, dass nicht alle Admins den Code anzeigen können, sondern nur die mit der entsprechenden Rolle.

Das ganze muss mit UI, Unit und Integration Tests getestet werden, und in der Kundendokumentation erwähnt werden.

#### Abgrenzung

Es gibt bereits die Funktionalität, dass Admins pro Nutzer Aktivierungsbriefe generieren können. Diese Briefe enthalten Bereits den Aktivierungscode. Im Rahmen dieses Auftrags, soll dieser Bereich nicht erweitert oder verändert werden.

### 9.2 Technische Referenzen

Für die Techinschen Infos sind folgende zwei Links sehr hilfreich:

- Futurae API: <https://www.futurae.com/docs/api/auth/>  
Da der Aktivierungs Code von Futurea kommt, ist dessen API Dokumentation eine wichtige Quelle.
- IAM Kunden Dokumentation: <https://docs.airlock.com/iam/8.3/> ' Notwendig, um allgemeine Informationen bezüglich Airlock 2FA nachzulesen

## 9.3 Anforderungen

Nach der Analyse und nachdem der Auftrag verstanden wurde konnten die Anforderungen definiert werden. Diese sind immer in funktionale und nicht-funktionale aufgeteilt.

Folgende Abkürzungen werden verwendet:

- FA <Zahl> ... bedeutet Funktionale Anforderung, mit numerisch aufsteigendem Index.
- NFA <Zahl> ... bedeutet nicht-funktionale Anforderung, mit numerisch aufsteigendem Index.

### 9.3.1 Rest / Backend

Folgend, sind die Anforderungen für das Backend resp. den Rest teil definiert.

#### Funktionale Anforderungen

- FA 1: Das Backend soll der SPA den Activation Code anbieten.
- FA 2: Der Activation Code darf nur angeboten werden, wenn der Admin auch die notwendige Rolle hat.
- FA 3: Es soll geloggt werden, welcher Administrator zu welchem Zeitpunkt den 16-stelligen Aktivierungscode angezeigt hat.
- FA 4: Neue Plugins oder Properties sollen einen klaren und vollständigen Hilfetext haben.

#### Nicht-funktionale Anforderungen

- NFA 1: Sämtliche Fehlerfälle werden korrekt behandelt.
- NFA 2: Der Code entspricht dem bestehenden Codeschema.
- NFA 3: Alle neuen Funktionalitäten werden durch Tests abgedeckt.
- NFA 4: Veränderte / neue Restendpoints werden um die notwendige Doku erweitert.

### 9.3.2 SPA

Folgend, sind die Anforderungen für die SPA definiert.

#### Funktionale Anforderungen

- FA 5: Die SPA muss in der Lage sein den 16-stelligen QR-Code auf Knopfdruck anzuzeigen.
- FA 6: Das neue UI verhält sich gleich wie das bisherige.
- FA 7: Das neue UI hat den gleichen Style wie das bisherige.



## **Nicht-funktionale Anforderungen**

- NFA 5: Es werden nur in der Adminapp existierende UI Komponenten verwendet.
- NFA 6: Das UI lädt in jedem Fall ohne Probleme.
- NFA 7: Alle neuen Funktionalitäten werden durch Selenium Integration Tests abgedeckt.

### **9.3.3 Kundendokumentation**

Folgend, sind die Anforderungen für die Kundendokumentation definiert.

## **Funktionale Anforderungen**

- FA 8: Die Kunden Doku wird sinnvoll um das neue Feature erweitert.
- FA 9: Die Kundendoku ist auf Englisch geschrieben.

## **Nicht-funktionale Anforderungen**

- NFA 8: Die Kundendoku hat keine Schreibfehler.
- NFA 9: Die Kundendoku passt in das bestehende Produkt.

## 10 Planen

In diesem Abschnitt, wird die Planung beschrieben. In dieser Phase werden basierend auf den Anforderungen Arbeitspakete erstellt, und in einem GANT-Diagramm auf die 10 Tage eingeteilt.

### 10.1 Arbeitspakete

Um den ganzen Auftrag in kleine übersichtliche Teile aufzuteilen, wird er in verschiedene kleine Arbeitspakete unterteilt. Die Arbeitspakete sind jeweils nummeriert, haben einen Namen, einen geschätzten Aufwand in h und eine «Definition of Done»/ ein erwartetes Ergebnis. Die Aufwände sind oft mit einem gewissen Puffer geschätzt.

Die Pakete sind nach den 6 Phasen der IPERKA Methode aufgelistet. Arbeiten welche IPA-spezifisch sind, sind unter Rahmenaufgaben aufgeführt.

#### 10.1.1 Informieren

Hier, sind die Arbeitspakete, welche während der IPERKA-Phase «Informieren» bearbeitet wurden, aufgelistet.

<b>Nummer</b>	1.1
<b>Name</b>	Projektumfeld analysieren und beschreiben
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Das Ziel der Arbeit ist klar, ein grober Überblick besteht.

<b>Nummer</b>	1.2
<b>Name</b>	Anforderungen definieren
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Die funktionalen und nicht-funktionalen Anforderungen sind definiert und beschrieben

### 10.1.2 Planen

Hier, sind die Arbeitspakete, welche während der IPERKA-Phase «Planen» bearbeitet werden, aufgelistet.

<b>Nummer</b>	2.1	
<b>Name</b>	Arbeitspakete definieren	
<b>Geschätzter Aufwand</b>	3h	
<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Die ganze Arbeit ist in kleine logische Arbeitspakete unterteilt. Alle Arbeitspakete sind klar definiert.
<b>Nummer</b>	2.2	
<b>Name</b>	Zeitplan erstellen	
<b>Geschätzter Aufwand</b>	1h	
<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Der GANT-Zeitplan ist anhand der Arbeitspakete erstellt. Es sind alle Arbeitspakete vorhanden.
<b>Nummer</b>	2.3	
<b>Name</b>	Lösungskonzept für das Backend erarbeiten	
<b>Geschätzter Aufwand</b>	4h	
<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Es ist mindestens ein Lösungsvorschlag definiert und so weit wie Sinnvoll beschrieben und durchgedacht. Der relevante Backendcode ist verstanden.

<b>Nummer</b>	2.4
<b>Name</b>	Lösungskonzept für die SPA erarbeiten
<b>Geschätzter Aufwand</b>	4h
<b>Erwartetes Ergebnis</b>	Es ist mindestens ein Lösungsvorschlag definiert und so weit wie Sinnvoll beschrieben und durchgedacht. Es sind verschiedene Mockups vorhanden, und der relevante SPA Code ist verstanden.

<b>Nummer</b>	2.5
<b>Name</b>	Test- und Qualitätssicherungskonzept erstellen
<b>Geschätzter Aufwand</b>	4h
<b>Erwartetes Ergebnis</b>	Das Testkonzept ist erstellt und dokumentiert. Das Qualitätssicherungskonzept ist erstellt und dokumentiert.

### 10.1.3 Entscheiden

Hier, sind die Arbeitspakete, welche während der IPERKA-Phase «Entscheiden» bearbeitet werden, aufgelistet.

<b>Nummer</b>	3.1
<b>Name</b>	Lösungsvarianten evaluieren
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Aus den verschiedenen Lösungsvarianten der SPA und des Backends wurde sich für eine entschieden, und dies Dokumentiert.

#### 10.1.4 Realisieren

Hier, sind die Arbeitspakete, welche während der IPERKA-Phase «Realisieren» bearbeitet werden, aufgelistet.

<b>Nummer</b>	4.1
<b>Name</b>	Das Backend erweitern
<b>Geschätzter Aufwand</b>	14h
<b>Erwartetes Ergebnis</b>	Alle Anforderungen für das Backend sind nach dem definierten Lösungsansatz umgesetzt. Zugleich ist die Lösung dokumentiert

<b>Nummer</b>	4.2
<b>Name</b>	Unit- und Integrationtests schreiben
<b>Geschätzter Aufwand</b>	6h
<b>Erwartetes Ergebnis</b>	Alle neuen Funktionalitäten sind mit Unit- und/oder Integrationtests getestet.

<b>Nummer</b>	4.3
<b>Name</b>	Die SPA erweitern
<b>Geschätzter Aufwand</b>	6h
<b>Erwartetes Ergebnis</b>	Alle Anforderungen für die SPA sind nach dem definierten Lösungsansatz umgesetzt. Zugleich ist die Lösung dokumentiert.

<b>Nummer</b>	4.4
<b>Name</b>	Selenium Integrationtests implementieren
<b>Geschätzter Aufwand</b>	5h
<b>Erwartetes Ergebnis</b>	Alle neuen Funktionalitäten sind mit Selenium Integrationtests getestet.

<b>Nummer</b>	4.5
<b>Name</b>	Kundendokumentation schreiben
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Die neue Funktionalität ist in der Kundendokumentation dokumentiert, und alle Anforderungen sind erfüllt.

### 10.1.5 Kontrollieren

Hier, sind die Arbeitspakete, welche während der IPERKA-Phase «Kontrollieren» bearbeitet werden, aufgelistet.

<b>Nummer</b>	5.1
<b>Name</b>	Tests durchführen, und Fehler beheben
<b>Geschätzter Aufwand</b>	4h h
<b>Erwartetes Ergebnis</b>	Tests sind gemäss Testkonzept durchgeführt, und mögliche Fehler sind behoben.

<b>Nummer</b>	5.2
<b>Name</b>	Codequalität prüfen, und Refactorn
<b>Geschätzter Aufwand</b>	1h
<b>Erwartetes Ergebnis</b>	Code ist nochmals durchgeschaut, und Unschönheiten sind bereinigt.

<b>Nummer</b>	5.3
<b>Name</b>	Dokumentation finalisieren
<b>Geschätzter Aufwand</b>	8h
<b>Erwartetes Ergebnis</b>	Die Dokumentation ist soweit wie möglich finalisiert und entspricht den Vorgaben.

### 10.1.6 Auswerten

Hier, sind die Arbeitspakete, welche während der letzten IPERKA-Phase «Auswerten» bearbeitet werden, aufgelistet.

<b>Nummer</b>	6.2
<b>Name</b>	Reflexion schreiben
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Reflexion zu den relevanten Abschnitten ist geschrieben.

### 10.1.7 Rahmenaufgaben

Hier, sind die Arbeitspakete, welche IPA-spezifische Arbeit erfordern, aufgelistet.

<b>Nummer</b>	7.1
<b>Name</b>	Projektstruktur aufsetzen
<b>Geschätzter Aufwand</b>	2h
<b>Erwartetes Ergebnis</b>	Das Grundgerüst für den Bericht steht. Der Latex-Build ist lauffähig und generiert ein anschauliches PDF.

<b>Nummer</b>	7.2
<b>Name</b>	Aufgabenstellung und Rahmenbedingungen beschreiben
<b>Geschätzter Aufwand</b>	1h
<b>Erwartetes Ergebnis</b>	Die Aufgabenstellung ist in den Bericht übernommen. Benützte Firmenstandards sowie die Projektaufbauorganisation sind definiert und beschrieben.

<b>Nummer</b>	7.3
<b>Name</b>	Projektmanagementmethode definieren
<b>Geschätzter Aufwand</b>	1h

<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Es steht fest mit welcher Projektmanagementmethode die Probe-IPA umgesetzt werden soll. Der Bericht wurde so gegliedert.
<b>Nummer</b>		7.4
<b>Name</b>		Expertenbesuche
<b>Geschätzter Aufwand</b>		4h
<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Infos aus dem Gespräch sind am richtigen Ort festgehalten.
<b>Nummer</b>		7.5
<b>Name</b>		Anhang erstellen
<b>Geschätzter Aufwand</b>		2h
<b>Erwartetes Ergebnis</b>	<b>Er-</b>	Der Anhang ist erstellt und beinhaltet alle nötigen und verlangten Inhalte.



## 11 Entscheiden

## 12 Realisieren

## 13 Kontrollieren

## 14 Auswerten

# Abbildungsverzeichnis

1	Logo der Ergon Informatik AG . . . . .	1
4.1	Arbeitsplatz . . . . .	10