

Projektnummer:	<b>3R IT 22 07</b>	Wien, im September 2021	
Antrag um Genehmigung einer Aufgabenstellung für die			
<b>DIPLOMARBEIT</b>			
Schuljahr:	<b>2021/22</b>	Anzahl Blätter:	<b>28</b>
Thema:	<b>Fabalous (Fabasoft Scrum-Tool)</b>		
<p><b>Aufgabenstellung:</b>          Das Projektteam soll im Rahmen der Diplomarbeit ein sogenanntes Scrum-Tool für die Fabasoft AG entwickeln. Dieses soll als Webpage erreichbar sein und das Bearbeiten von User-Story-Changelogs, sowie das Generieren von Scrum-Reports ermöglichen. Im Backend muss dafür eine entsprechende Anbindung zu der Fabasoft-Cloud, sowie ein Parser für die firmeninterne Changelog-Syntax geschrieben werden.</p>			
Kandidatinnen/Kandidaten:	Klasse	Individ. Betreuung	Unterschrift Kandidatinnen
Projektleiter	5BN	DAZ	
<b>Benjamin Komar</b>			
Stellv. Projektleiter	5BN	DAZ	
Nick Müllner			
Yannik Wailzer	5AI	DAZ	
Nick Fischer	5BN	KLE	
Betreuerinnen/Betreuer:			Unterschrift
Individuelle Betreuung (Hauptbetreuung):			
<b>Robert Dazing</b> (ohne Titel)			
Individuelle Betreuung (Hauptbetreuung Stv.):			
Christian Klein (ohne Titel)			

Als Diplomarbeit zugelassen

Wien, am .....

.....

AV Gabriela Herrele

## Executive Summary

### Objectives

The goal of the diploma is to create a so-called *ScrumTool* for the company Fabasoft. This tool should be able to:

- Create reports of different types with various settings
- Edit changelogs of User-Stories that are stored in the Fabasoft Business Cloud

The end-user should be able to access the *ScrumTool* via a Web-Interface where the backend handles the following scenarios:

- Upload and download User Stories as well as images that may be inside of changelogs
- Parse the internal syntax of the User-Story changelogs
- Generating reports based on the chosen User-Stories, which type of report should be generated and type-specific settings

Finally, an administrator should also be able to:

- Monitor / Log the behavior of the different services included in the *ScrumTool*
- Define which users are able to use the application (Authentication and Authorization)
- Build, test and deploy the application from a centralized point

### Risks

- **Licensing:**

As Fabasoft owns the source code of the Scrum tool, it must remain closed-source and must not use any libraries/tools licenced under violating licences (e.g. copyleft). Such licences may not be noticed by team members which can cause problems later on.

The licence of each and every library/tool must be checked (automatically or manually).

- **Interface to Fabasoft:**

The project has a dependency on Fabasoft: the communication between team and Fabasoft, but also their adherence to deadlines. If communication or deadlines fail, the project may get stuck and problems could appear.

To avoid misunderstandings Nick Müllner (currently working at Fabasoft) will act as the main communicator. The team may also compare goals and deadlines when presenting product increments.

## Milestones

Date	Milestone
24.09.2021	Project start
03.10.2021	A prototype has been created
31.10.2021	Editor, parsing and authentication have been finished
28.11.2021	Report- / Appsettings, Report-Generation, Cloudadapter and Logging have been completed
26.12.2021	Completion of optional goals
End of December	Project intern intermediate presentation has been held
23.01.2022	End-To-End Testing
February 2022	First version of diploma has been sent
20.03.2022	Diploma has been completed
March 2022	Final presentation has been held
April 2022	Final version of the diploma has been sent
May 2022	Thesis defence has been held

## Budget and Resources

Pos.	Description of the effort	Costs
1	Server costs for 1 year	EUR 314,40
2	Domain costs for 1 year	EUR 15,49
	Total costs	EUR 329,89

Pos.	Description of the cost coverage
1	Costs covered by the cooperation partner (Fabasoft)
2	Costs covered by the project team

Project budget	EUR 329,89
Costs for school	EUR 0
Total man hours	760 h

## Inhaltsverzeichnis

<b>1</b>	<b>PROJEKTIDEE .....</b>	<b>5</b>
1.1	AUSGANGSSITUATION .....	5
1.2	BESCHREIBUNG DER IDEE .....	5
<b>2</b>	<b>PROJEKTZIELE .....</b>	<b>6</b>
2.1	HAUPTZIELE .....	6
2.2	OPTIONALE ZIELE .....	9
2.3	NICHT ZIELE .....	11
2.4	INDIVIDUELLE AUFGABENSTELLUNGEN DER TEAMMITGLIEDER IM GESAMTPROJEKT....	12
<b>3</b>	<b>PROJEKTORGANISATION.....</b>	<b>14</b>
3.1	GRAFISCHE DARSTELLUNG.....	14
3.2	PROJEKTTEAM.....	14
<b>4</b>	<b>PROJEKTUMFELDDANALYSE .....</b>	<b>15</b>
4.1	GRAFISCHE DARSTELLUNG.....	15
4.2	BESCHREIBUNG DER WICHTIGSTEN UMFELDER.....	16
<b>5</b>	<b>RISIKOANALYSE .....</b>	<b>19</b>
5.1	BESCHREIBUNG DER WICHTIGSTEN RISIKEN .....	19
5.2	RISIKOPORTFOLIO.....	20
5.3	RISIKO GEGENMAßNAHMEN .....	21
<b>6</b>	<b>OBJEKTSTRUKTURPLAN .....</b>	<b>22</b>
<b>7</b>	<b>PROJEKTSTRUKTURPLAN .....</b>	<b>23</b>
<b>8</b>	<b>MEILENSTEINLISTE .....</b>	<b>24</b>
<b>9</b>	<b>PROJEKTRESSOURCEN .....</b>	<b>25</b>
9.1	PROJEKTRESSOURCEN: SOLL – IST VERGLEICH.....	25
9.2	PERSONELLE RESSOURCEN .....	25
9.3	BUDGET .....	26
<b>10</b>	<b>GEPLANTE EXTERNE KOOPERATIONSPARTNER .....</b>	<b>27</b>
<b>11</b>	<b>GEPLANTE VERWERTUNG DER ERGEBNISSE.....</b>	<b>28</b>

# 1 Projektidee

## 1.1 Ausgangssituation

Das Projekt wird in Kooperation mit der Firma Fabasoft durchgeführt, die Projektidee stammt somit auch von ihnen. Das Team hatte sich bereits Anfang der vierten Klasse gefunden und bestand auf der Bedingung, dass das Projekt großteils aus Softwareentwicklung besteht, aber auch durch die Unterstützung eines Teammitgliedes aus der Medientechnik sollen Elemente der Medientechnik wie Web-Frontend-Entwicklung und UI-Design eingebaut werden.

Wie bereits erwähnt werden die Ziele von der Firma vorgegeben, diese können in Absprache mit Fabasoft priorisiert werden und somit mehr oder weniger fokussiert werden. Deshalb hat sich das Team entschieden, das Projekt mit der Methode Scrum durchzuführen, diese bietet in der Softwareentwicklung eine deutlich höhere Flexibilität und gestattet dem Kunden (Fabasoft) auf unsere Entwicklung zeitnah Feedback zu geben und beim Abdriften der Ziele während dem Entwickeln einzuschreiten.

Die Kommunikation und Vermittlung zur Firma wird von Nick Müllner übernommen, welcher auch den Vorschlag hatte, die Diplomarbeit in Verbindung mit Fabasoft durchzuführen.

## 1.2 Beschreibung der Idee

Ziel ist es, das momentane Scrum-Tool der Abteilung *Fabasoft eGov Editions*, momentan genannt *ScrumTool3*, komplett zu erneuern. Dieses Tool funktioniert zum jetzigen Stand nur als Fat-Client mit Anbindungen an die Fabasoft Cloud, welches jedoch am Ende der Diplomarbeit als State-of-the-Art Server-Client Applikation bestehen soll. Dabei wird die gesamte Anwendung hinsichtlich Architektur und UI neu designed und entwickelt und nicht umgeschrieben.

Das Scrum-Tool besitzt drei Hauptfunktionen:

1. Generierung von Reports von mehreren Stories mit verschiedenen Ausgabeformaten
2. Erstellen und Bearbeiten von Change Logs innerhalb Stories (besitzt eigene Markup-Syntax)
3. Einbindung in die bestehende Fabasoft Business Process Cloud und Active Directory Infrastruktur

Das Tool besitzt innerhalb der zwei Funktionen sehr umfangreiche Logik, wie bestimmte Generierungsbedingungen und -optionen oder einen Lexer, Parser und Compiler mit entsprechender Markup-Syntax. Diese Funktionen sollen Schritt für Schritt implementiert werden und werden bei der Größe des jetzigen Tools wahrscheinlich innerhalb der gegebenen Zeit nicht vollständig als Web-Anwendung abbildbar sein. Deshalb werden die potenziellen Ziele als Story vom PO und Fabasoft priorisiert und gereiht. Außerdem besitzt das Tool eine Anbindung zur Fabasoft Business Process Cloud über das Scrum Service mittels SOAP bzw. einer JSON-basierten REST API.

## 2 Projektziele

### 2.1 Hauptziele

#### Ziel-H 1 Reports generieren: Positivliste an Stories angeben

Das Tool nimmt eine Liste an Scrum-Stories aus der Cloud entgegen in Form von

- Fabasoft Cloud Ordner
- einzelne Backlog Elemente als COO-Adresse
- mx-URL
- Friendly-URL
- Story-ID
- Objektliste

und gibt je nach ausgewähltem Ausgabeformat einen Report aus.

#### Ziel-H 2 Reports generieren: Negativliste an Stories angeben

Das Tool nimmt eine Liste an Scrum-Stories aus der Cloud entgegen in Form von

- Fabasoft Cloud Ordner
- einzelne Backlog Elemente als COO-Adresse
- mx-URL
- Friendly-URL
- Story-ID
- Objektliste

und subtrahiert die erhaltene Liste an Stories von der Positivliste bei der Generierung eines Reports.

#### Ziel-H 3 Reports generieren: Report Settings

Neben der Positiv- und Negativliste sollen auch allgemeine Settings vor dem Generieren eines Reports angegeben werden können:

- Ausgabeformat
- Angabe der Kategorien, die in den Changelogs angezeigt werden sollen
- Export-Formate
- Story-Tests anzeigen
- Story-COO-Adressen vor dem Generieren anzeigen
- Für Hotfix
  - o Angabe der Zielversion
  - o Check-In Daten anzeigen
  - o Nur Stories mit passender Version anzeigen
  - o Story-COO-Adressen im Report anzeigen
  - o Nach Buildnummern und/oder Datum und Uhrzeit filtern

#### Ziel-H 4 Reports generieren: Ausgabeformat „What's New“

Rendering eines „What's New“ Reports: Auflistung der Stories sortiert nach Kategorien. Angezeigt wird der Change Log und die Story-ID der Story. Zeigt niemals Change Logs mit der Kategorie *internal* an.

**Ziel-H 5**      Reports generieren: Ausgabeformat „Hotfix“

Rendering eines „Hotfix“ Reports: Filtert Stories die als Hotfix markiert sind und listet diese mit Entwicklerinformationen (siehe Change Log) auf. Es muss auch eine Zielversion angegeben werden, es werden nur Stories die diese Version beeinflussen angezeigt.

**Ziel-H 6**      Reports generieren: Ausgabeformat „Changelog“

Rendering eines „Change Log“ Reports: Auflistung der Stories mit Entwicklerinformationen wie Entwicklernamen, Status, Change Log, veränderte Dateien in den Commits, ... (alle Informationen sind im momentanen Scrum-Tool sichtbar).

**Ziel-H 7**      Globale Programm-Einstellungen

Das Tool bezieht unternehmensspezifische Informationen und Konfigurationen aus einer Konfigurations-Datei. Diese Datei kann durch den zu wartenden Mitarbeiter mittels eines externen Editors editiert werden. Diese Datei beinhaltet, unter anderem, folgende Einstellungen:

- Daten des Servicebenutzers für Zugang in die Fabasoft Cloud
- Zugangsdaten zur Fabasoft Cloud (https URL)
- Optional: Zugangsdaten zu Jenkins (für die Abfrage der Builds; URL, Login und Passwort; nicht zwingend notwendig)
- Unterstützte und bekannte Versionen, wobei pro Version mehrere Branches mit Lebensdauer existieren
- Unterstützte Editionen
- Unterstützte Softwarekomponenten mit SCM Branch-Pfaden, Kürzel, Ziel-COO-Datei am ISO und Ziel-JAR-Datei am ISO, etc.
- Zu exkludierende SCM Pfade

**Ziel-H 8**      Markup-Editor

Die Webansicht bietet einen Editor mit dem mittels der firmeninternen Markup-Syntax Change-Logs eingetragen und bearbeitet werden können.

Die Syntax ist an inline HTML/XML-Tags angelehnt und Tags stellen Objekte der Fabasoft-Cloud dar. Dafür werden die angegebenen Daten auf andere Weise formatiert, um sie von normalem Text abzuheben.

Die wichtigsten Tags sind:

- `<obj name="" />`
- `<obj ref="" />`
- `<obj name="" ref="" />`
- `{{ imgpath }}`
- `*`, `**` (ungeordnete Aufzählungen)
- `#`, `##` (geordnete Aufzählungen)

Die Attribute der obj-Tags name und ref stehen für den Objektnamen und die zugehörige COO-Adresse (z.B. `<obj name="Objekt A" ref="COO@21.1:object" />`) und werden kursiv bzw. monospace (bzw. monospace in Klammern) formatiert.

Bilder werden über deren absoluten Pfad in der Fabasoft-Cloud (URL) eingefügt und werden als Inline-Bilder angezeigt.

Geordnete/ungeordnete Aufzählungen können mit \* und # erstellt und maximal einmal eingerückt werden.

Alle anderen Tags der Logentries werden über das Tool außerhalb des Editors generiert und zu einem gesamten Scrumdoc-Eintrag zusammengefügt.

Der gesamte Changelog kann schlussendlich zugeordnet zu einer Userstory (zu deren COO-Adresse bzw. deren Feld für Changelogs) in die Fabasoft-Cloud hochgeladen werden.

Bilder können zusätzlich entweder per Dateiauswahl (Button) oder direkt per Copy/Paste aus der Zwischenablage (z.B. Screenshots) in den Editor eingefügt und hochgeladen werden, wodurch die angegebene Image-URL durch die des hochgeladenen Bildes ersetzt wird.

#### Ziel-H 9 Markup-Editor: Syntax-Highlighting

Der Editor unterstützt Syntax-Highlighting für Tags, Attributen innerhalb Tags und Attributen-Werte. Z.B.

```
<obj name="Dynamischer Filter für Aktivitäten"  
ref="FSCWFEXT@15.1001:usedynamicactivityfilter">
```

#### Ziel-H 10 Markup-Editor: Live-Preview

Neben dem Markup-Editor wird in Echtzeit eine geparte Version des Markups angezeigt, die der geparten Ansicht der Einträge entspricht (wie wenn ein Report generiert wird).

#### Ziel-H 11 Markup-Editor: Autocompletion

Um Einträge einfacher schreiben zu können, bietet der Editor für die genannten Tags ein Autocompletion-Menü (beim Tippen und per Strg+Leertaste), über das Code-Snippets/Vervollständigungen für die Tags eingefügt werden können, in denen die Cursorposition für die jeweiligen Felder eines Tags automatisch angepasst wird (ähnliches Verhalten wie VS-Code Snippets). Alternativ könnte hier auch über Buttons gearbeitet werden, die Snippets einzufügen, wie im derzeitigen Tool.

#### Ziel-H 12 User Authentifizierung & Autorisierung

Die Benutzer werden beim Aufruf der Applikation von einem Active Directory eingebundenen PC automatisch mit ihrem Windows Benutzer eingeloggt. Falls der Benutzerbrowser den Anwender nicht einloggen kann, müssen Benutzer sich beim Aufruf der Applikation mit ihrem Active Directory Benutzernamen und Passwort anmelden. Es gibt außerdem die Möglichkeit sich ohne Angabe von einem Passwort einzuloggen, wobei man nur „read-only“ Zugang zu der Report-Generierungsfunktionalität bekommt.



**Ziel-H 13**    Logging & Monitoring

Die Applikation berichtet über Events (z.B. „Ein Benutzer hat sich eingeloggt“ oder „Ein Benutzer hat einen Report generiert“) durch „Log-Messages“, welche von Administratoren jederzeit eingesehen werden können. Diverse Statistiken (z.B. „Uptime“ und „RAM-Verbrauch“) können von Administratoren eingesehen werden. Diese Log-Messages und Statistiken werden für einen konfigurierbaren Zeitraum gespeichert.

**Ziel-H 14**    Kommunikation mit der Fabasoft Business Process Cloud

Die Applikation kommuniziert mit der Fabasoft Cloud mittels ihrer SOAP API. Die Applikation selbst hat einen eigenen Fabasoft-Cloud-Benutzer, welcher benutzt wird, um auf die Fabasoft Cloud zuzugreifen.

**Ziel-H 15**    Automatisierter Bau & Testprozess

Die Applikation kann automatisch auf einem aufgesetzten Server vollständig gebaut, getestet, sowie testhalber installiert werden. Nach jedem Sprint soll eine Testinstallation erzeugt werden, um es dem Kunden vorzeigen zu können.

**Ziel-H 16**    Web-App & UI allgemein

Die Funktionalität des gesamten Frontends wird als Web-App bereitgestellt. Diese hat eigene geroutete Bereiche für das Generator Frontend, den Editor, den Task Generator und die Usersettings. Außerdem besitzt sie ein einheitliches modernes Design und ist durch Einsatz von modernen Frontend Frameworks leicht skalier-/erweiterbar (auch später von Fabasoft).

## 2.2 Optionale Ziele

**Ziel-O 1**    Reports generieren: Automatische Ermittlung – Positiv- und Negativliste

Mittels einer Jenkins Anbindung kann das Tool selbstständig herausfinden, welche Stories Teil einer Zielversion sind, die im Report beinhaltet werden sollen. Neben den gebauten Builds werden auch gelieferte Shipments, deren Artefakte sowie Daten der Sprints bei der automatischen Ermittlung berücksichtigt.

**Ziel-O 2**    Reports generieren: Auswertung nach Buildnummern

Es ist möglich, eine Start- und eine Endbuildnummer beim Generieren anzugeben. Diese sorgt dafür, dass nur Stories zwischen diesen zwei Buildnummern ausgewertet werden (benötigt Jenkins Anbindung). Außerdem sind auch branchübergreifende Auswertungen möglich.

**Ziel-O 3**    Reports generieren: PDF-Export

Reports können zusätzlich zu der Webansicht als PDF exportiert werden. Diese können dann heruntergeladen werden.

**Ziel-O 4     Lokalisierte UI**

Damit das Tool in Zukunft auch abteilungsübergreifend von englischsprachigen Usern verwendet werden kann, ist die UI für Deutsch (Standard) und Englisch lokalisiert.

**Ziel-O 5     Markup-Editor: Dictionary**

Der Editor bietet ein Dictionary-Fenster, über das Begriffe zur Übersetzung gesucht werden können. Damit können Übersetzungsfehler und Missverständnisse bei üblichen Begriffen der DACH-Länder verhindert werden.

**Ziel-O 6     Task-Generator**

Aus einer Liste an vordefinierten allgemeinen Tasks (bereits vorhanden) können beliebige ausgewählt, mit Kommentaren versehen und zu einer Userstory über deren COO-Adresse/Story-ID hinzugefügt werden (SOAP oder JSON-basiertes REST).

**Ziel-O 7     Updaten der Applikation ohne Serviceausfall**

Die Applikation kann bei Bedarf teilweise oder gänzlich im laufenden Betrieb von einem Administrator upgedated werden. Dieser Vorgang darf die laufende Arbeit eines Endbenutzers nicht unterbrechen oder behindern.

**Ziel-O 8     Backup- und Restore Konzept**

Damit die Anwendung im Falle eines Totalausfalles des Servers mit wenigen Schritten wieder zum Laufen gebracht werden kann, benötigt es ein Konzept für Backups und Restores. Dabei werden alle notwendigen Daten gesichert, was für einen einwandfreien Betrieb der Anwendung sorgt.

**Ziel-O 9     Reports generieren: Ausgabeformat „TAM-Hotfix“**

Zusätzlich zu Hotfix werden bei TAM-Hotfix auch ISO-Dateien der eGov Suite, Übersetzungstabellen und automatisierte Übersetzungsservices (z.B. DeepL) abgefragt, um deutsche/englische Übersetzungen zu generieren.

## 2.3 NICHT Ziele

### Ziel-N 1 Scrum Prozess abbilden

Das Tool kann den gesamten Scrum Prozess abbilden und dient als primäres Tool, um Scrum zu betreiben. Es besitzt alle notwendigen Funktionen die Scrum unterstützen wie Jira oder Taiga.

### Ziel-N 2 Responsiveness

Das Arbeiten auf mobilen Endgeräten wie Handys oder Tablets ist dank eines responsiven Designs möglich.

### Ziel-N 3 Accessibility

Die Anwendung kann auch von Menschen mit Einschränkungen (z.B. Blindheit) verwendet werden. Dazu werden entsprechende Funktionen für Personen mit Einschränkungen implementiert.

### Ziel-N 4 Editor für Globale Programm-Einstellungen

Das Tool besitzt einen internen Editor, mit dem die globalen Programm-Einstellungen editiert werden können. Nur Administratoren haben Zugriff auf diesen Editor.

## 2.4 Individuelle Aufgabenstellungen der Teammitglieder im Gesamtprojekt

### 2.4.1 Benjamin Komar

Themenschwerpunkt	Markup-Editor Frontend & Report Frontend
Aufgabenstellung Auflistung der einzelnen Ziele und Anforderungen	<ul style="list-style-type: none"> <li>• ZIEL-H 1 Reports generieren: Positivliste an Stories angeben</li> <li>• ZIEL-H 2 Reports generieren: Negativliste an Stories angeben</li> <li>• ZIEL-H 3 Reports generieren: Report Settings</li> <li>• ZIEL-H 8 Markup-Editor</li> <li>• ZIEL-H 9 Markup-Editor: Syntax-Highlighting</li> <li>• ZIEL-H 10 Markup-Editor: Live-Preview</li> <li>• ZIEL-H 11 Markup-Editor: Autocompletion</li> <li>• ZIEL-O 1 Reports generieren: Automatische Ermittlung – Positiv- und Negativliste</li> <li>• ZIEL-O 2 Reports generieren: Auswertung nach Buildnummern</li> <li>• ZIEL-O 5 Markup-Editor: Dictionary</li> </ul>

### 2.4.2 Nick Müllner

Themenschwerpunkt	Implementierung der Fabasoft Cloud API und des Changelog Parsers
Aufgabenstellung Auflistung der einzelnen Ziele und Anforderungen	<ul style="list-style-type: none"> <li>• ZIEL-H 1 Reports generieren: Positivliste an Stories angeben</li> <li>• ZIEL-H 2 Reports generieren: Negativliste an Stories angeben</li> <li>• ZIEL-H 4 Reports generieren: Ausgabeformat „What’s New“</li> <li>• ZIEL-H 5 Reports generieren: Ausgabeformat „Hotfix“</li> <li>• ZIEL-H 6 Reports generieren: Ausgabeformat „Changelog“</li> <li>• ZIEL-H 14 Kommunikation mit der Fabasoft Business Process Cloud</li> <li>• ZIEL-O 1 Reports generieren: Automatische Ermittlung – Positiv- und Negativliste</li> <li>• ZIEL-O 2 Reports generieren: Auswertung nach Buildnummern</li> <li>• ZIEL-O 9 Reports generieren: Ausgabeformat „TAM-Hotfix“</li> </ul>

### 2.4.3 Nick Fischer

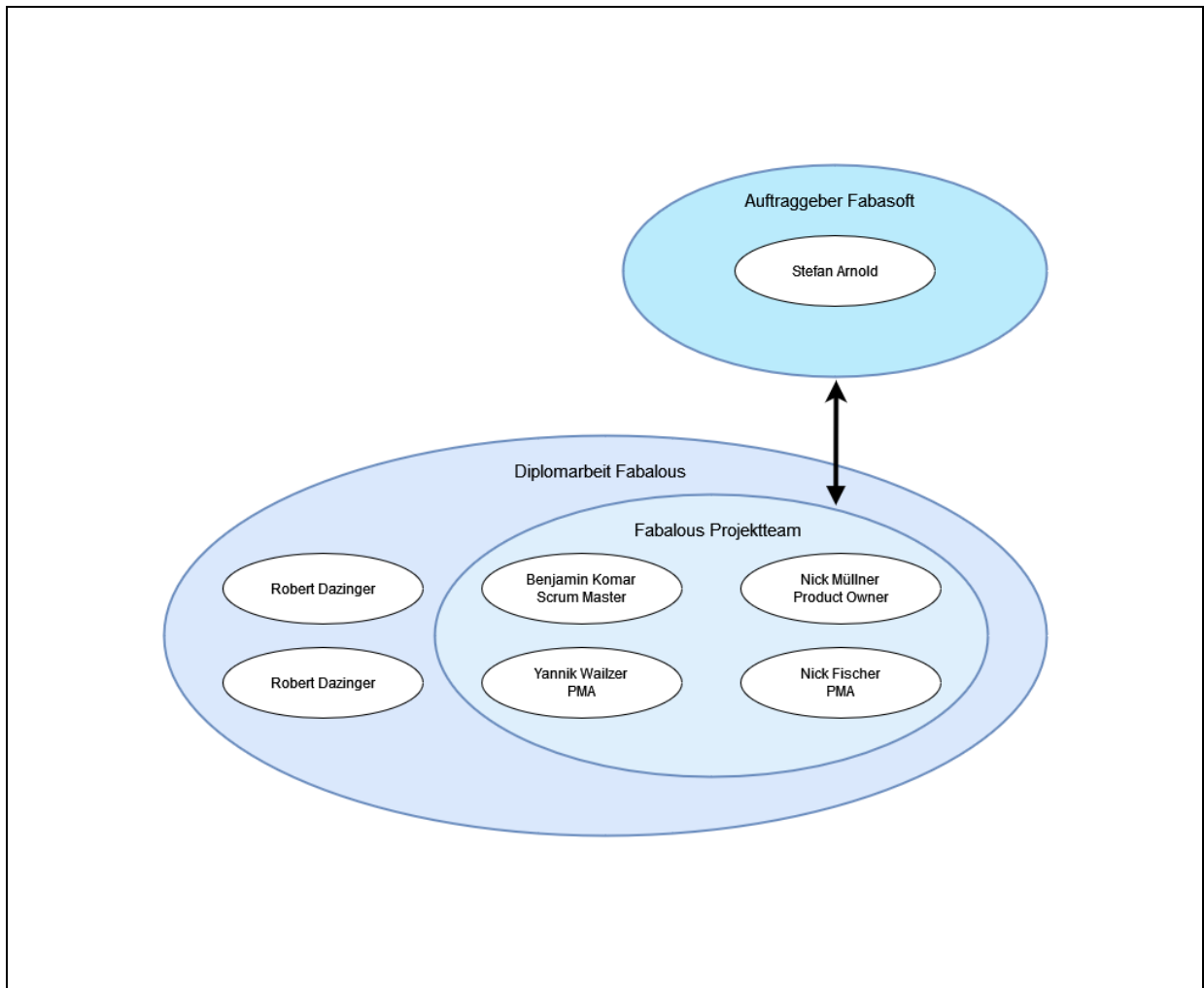
Themenschwerpunkt	Entwicklung der Authentifizierungs- und Managementkomponenten sowie der Entwicklungsinfrastruktur
Aufgabenstellung Auflistung der einzelnen Ziele und Anforderungen	<ul style="list-style-type: none"> <li>• ZIEL-H 7 Globale Programm-Einstellungen</li> <li>• ZIEL-H 12 User Authentifizierung &amp; Autorisierung</li> <li>• ZIEL-H 13 Logging &amp; Monitoring</li> <li>• ZIEL-H 15 Automatisierter Bau &amp; Testprozess</li> <li>• ZIEL-O 7 Updaten der Applikation ohne Serviceausfall</li> <li>• ZIEL-O 8 Backup- und Restore Konzept</li> </ul>

### 2.4.4 Yannik Wailzer

Themenschwerpunkt	UI Allgemein & Generator Frontend
Aufgabenstellung Auflistung der einzelnen Ziele und Anforderungen	<ul style="list-style-type: none"> <li>• ZIEL-H 4 Reports generieren: Ausgabeformat „What's New“</li> <li>• ZIEL-H 5 Reports generieren: Ausgabeformat „Hotfix“</li> <li>• ZIEL-H 6 Reports generieren: Ausgabeformat „Changelog“</li> <li>• ZIEL-O 3 Reports generieren: PDF-Export</li> <li>• ZIEL-O 4 Lokalisierte UI</li> <li>• ZIEL-O 6 Task-Generator</li> <li>• ZIEL-O 9 Reports generieren: Ausgabeformat „TAM-Hotfix“</li> <li>• ZIEL-H 16 Web-App &amp; UI allgemein</li> </ul>

### 3 Projektorganisation

#### 3.1 Grafische Darstellung

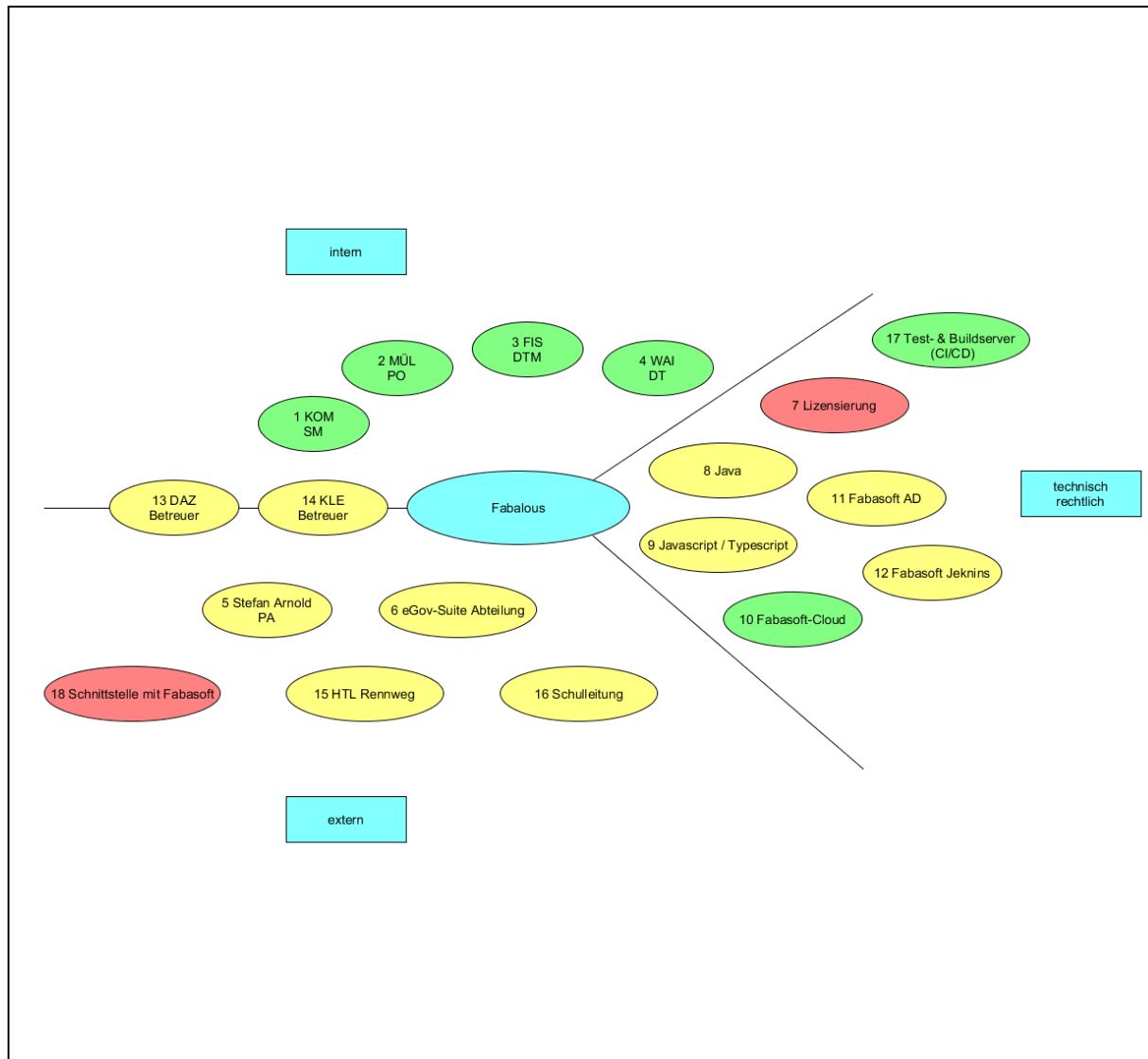


#### 3.2 Projektteam

Funktion	Name	Kürzel	E-Mail
SM	Benjamin Komar	KOM	<a href="mailto:7149@htl.rennweg.at">7149@htl.rennweg.at</a>
PO	Nick Müllner	MÜL	<a href="mailto:7152@htl.rennweg.at">7152@htl.rennweg.at</a>
PMA	Nick Fischer	FIS	<a href="mailto:7150@htl.rennweg.at">7150@htl.rennweg.at</a>
PMA	Yannik Wailzer	WAI	<a href="mailto:7163@htl.rennweg.at">7163@htl.rennweg.at</a>

## 4 Projektumfeldanalyse

### 4.1 Grafische Darstellung



## 4.2 Beschreibung der wichtigsten Umfeldler

#	Bezeichnung	Beschreibung	Bewertung
1	KOM	SM: Kümmert sich um die Projektorganisation und hat gemeinsam mit #3 FIS viel Erfahrung in Web-APIs und kann somit beim Designen der Schnittstellen mithelfen.	+
2	MÜL	PO: Da Nick Müllner bei der Fabasoft AG als Softwareentwickler angestellt ist, hat er am meisten Know-How über die Firmeninternen Tools / Services.	+
3	FIS	DTM: Hat gemeinsam mit #1 KOM viel Erfahrung in Web-APIs und kann somit beim Designen der Schnittstellen mithelfen.	+
4	WAI	DTM: Hat am meisten Know-How bezüglich Frontend und Design und wird sich entsprechend darauf während dem Projekt fokussieren.	+
5	Stefan Arnold	Auftraggeber des Projektes: Unterschreibt mit dem Team den Projektauftrag und führt mit dem Team die Produktabnahme durch. Fordert (gemeinsam mit #6 eGov-Suite Abteilung) das Feature-Set an, welches das Team implementieren soll.	~
6	eGov-Suite Abteilung	Verwendet das jetzige Scrum-Tool und fordert insofern (gemeinsam mit #5 Stefan Arnold) auch das Feature-Set an, welches das Team implementieren soll.	~
7	Lizensierung	Da der Source-Code des Scrum-Tools als closed-source bei der Fabasoft AG bleiben soll, dürfen keine Libraries bzw. Tools verwendet werden, die dieser Richtlinie widersprechen.	-
8	Java	Wird als Backend-Sprache für das Projekt benutzt, da mit dieser alle Teammitglieder bereits ausgiebig Erfahrung haben.	~
9	Javascript / Typescript	Wird als Frontend-Sprache für das Projekt benutzt, da mit dieser alle Teammitglieder bereits ausgiebig Erfahrung haben.	~



10	Fabasoftware-Cloud	<p>1. Dort wird das Team seinen eigenen Scrum-Prozess abbilden (Product-Backlog / Sprints). Da die Fabasoftware-Cloud entsprechend auch von der Fabasoftware selbst verwendet wird, sollten alle Features, die das Team für das Projektmanagement braucht, vorhanden sein.</p> <p>2. Wird durch das Scrum-Tool angesprochen, um die Informationen von den verschiedenen User-Stories für die Reports und Changelogs zu bekommen.</p>	+
11	Fabasoftware AD	Muss vom Scrum-Tool aus erreichbar sein / angesprochen werden können, um eine User-Authentifizierung / Autorisierung zu implementieren.	~
12	Fabasoftware Jenkins	Muss vom Scrum-Tool aus erreichbar sein / angesprochen werden können, um S-IDs (User-Story IDs) von einem geg. Build zu bekommen.	~
13	DAZ	Betreuer unseres Projektes: Erhält alle vier Wochen Management-Summaries bzw. trifft sich mit dem Team alle vier Wochen zu einem Meeting. Ist für die Benotung der Diplomarbeit zuständig.	~
14	KLE	Betreuer unseres Projektes: Erhält alle vier Wochen Management-Summaries bzw. trifft sich mit dem Team alle vier Wochen zu einem Meeting. Ist für die Benotung der Diplomarbeit zuständig.	~
15	HTL Rennweg	Gibt die Rahmenbedingungen für die Durchführung der Diplomarbeit vor. Bei Auftretenden Problemen wird aber seitens der Lehrkräfte Hilfe angeboten.	~
16	Schulleitung	Bewilligt die Diplomarbeit und ist deshalb als potenzielles Anfangsrisiko eingeschätzt, da möglicherweise Teile des Projektes abgeändert werden müssen. Wird das Projekt aber problemlos bewilligt, stellt dies kein Risiko dar.	~
17	Test- & Buildserver (CI/CD)	Wird zwecks Flexibilität und besserem Zugang extern bei Easynome statt bei Fabasoftware gehostet (durch WAI). Der Server muss dadurch in eigener Verantwortung gewartet und z.B. bei Leistungseinbrüchen upgegradet werden. Die anfallenden Kosten übernimmt Fabasoftware.	+

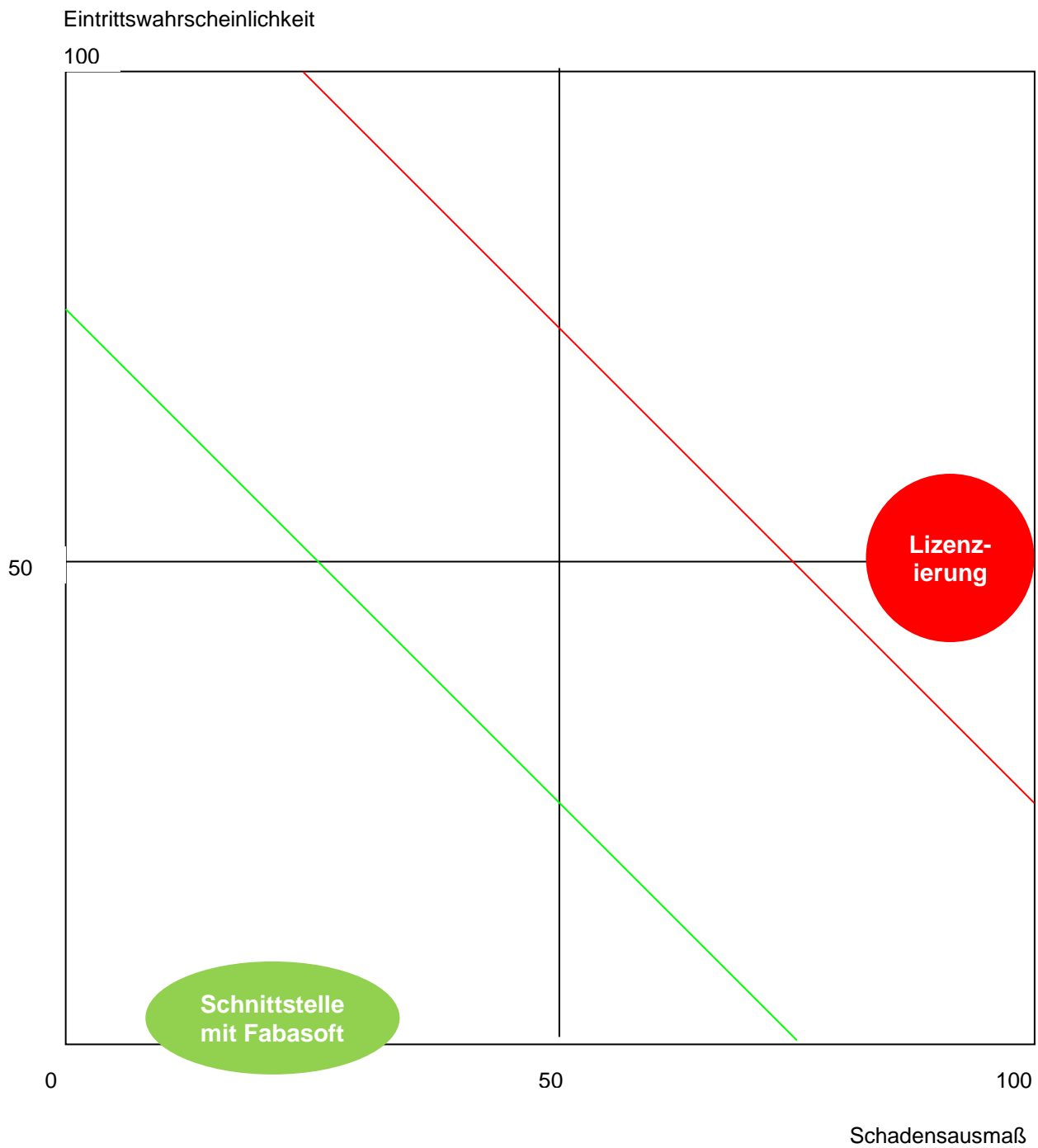
18	Schnittstelle mit Fabasoft	Das Projekt besitzt eine Abhängigkeit von Fabasoft, darunter zu verstehen ist die generelle Kommunikation mit Fabasoft (Vermittlung von Zielen, Änderungswünschen, etc.) und die Einhaltung von Terminen von Seite der Fabasoft. Bei Nichteinhaltung von Terminen oder Aussetzen der Kommunikation kann es zu Problemen im Projekt kommen.	-
----	----------------------------	--	---

## 5 Risikoanalyse

### 5.1 Beschreibung der wichtigsten Risiken

#	Bezeichnung	Beschreibung des Risikos	P	A	RF
7	Lizensierung	<p>Da der Source-Code des Scrum-Tools als closed-source bei der Fabasoft AG bleiben soll, dürfen keine Libraries bzw. Tools verwendet werden, die dieser Richtlinie widersprechen.</p> <p>Unter Umständen könnte eine solche Library / ein solches Tool fälschlicherweise verwendet werden, ohne, dass es einem Teammitglied auffällt. Zudem könnte es für eine Library / ein Tool mit unzureichender Lizenz keine Alternative geben (sehr unwahrscheinlich).</p>	50	100	5000
18	Schnittstelle mit Fabasoft	<p>Das Projekt besitzt eine Abhängigkeit von Fabasoft, darunter zu verstehen ist die generelle Kommunikation mit Fabasoft (Vermittlung von Zielen, Änderungswünschen, etc.) und die Einhaltung von Terminen von Seite der Fabasoft. Bei Nichteinhaltung von Terminen oder Aussetzen der Kommunikation kann es zu Problemen im Projekt kommen.</p>	3	20	60

## 5.2 Risikoportfolio



### 5.3 Risiko Gegenmaßnahmen

#	Bezeichnung	Gegenmaßnahme
7	Lizensierung	Für jede verwendete Library und jedes verwendete Tool muss die Lizenz überprüft werden (automatisch durch CI/CD oder manuell und in einem Dokument niedergeschrieben).
18	Schnittstelle mit Fabasoft	Um Missverständnisse vorzubeugen und die Kommunikation regelmäßig zu halten, besucht Nick Müllner regelmäßig das Fabasoft Büro (er arbeitet dort Geringfügig) und das Team besitzt sowieso die Möglichkeit bei Sprint-Abnahmen ihre Ziele und Termine mit Fabasoft abzugleichen.

## 6 Objektstrukturplan



## 7 Projektstrukturplan

Der Scrum-Backlog wurde elektronisch als Anhang mitgeschickt.

## 8 Meilensteinliste

Datum	Meilenstein
24.09.2021	Projektstart
03.10.2021	Ein Minimal-Prototyp wurde erstellt
31.10.2021	Editor, Parsing und Authentifizierung wurden fertiggestellt
28.11.2021	Report- / Appsettings, Report-Generierung, Cloudadapter und Logging wurden fertiggestellt
26.12.2021	Fertigstellung optionaler Ziele
Ende Dezember	Projektinterne Zwischenpräsentation wurde abgehalten
23.01.2022	End-To-End Testing
Februar 2022	Erste Version des Diplomarbeitsbuch wurde ausgesendet
20.03.2022	Diplomarbeitsbuch wurde fertiggestellt
März 2022	Finale Präsentation wurde gehalten
April 2022	Finale Version des Diplomarbeitsbuch wurde ausgesendet
Mai 2022	Defensio wurde abgehalten



## 9 Projektressourcen

### 9.1 Projektressourcen: Soll – Ist Vergleich

SOLL Bereich	IST	Risiko-Nr.	PSP-Code
KNOW HOW im Bereich Frontend (mit VueJS)	ausreichend	/	/
KNOW HOW im Bereich REST APIs	ausreichend	/	/
KNOW HOW im Bereich SOAP APIs	ausreichend	/	/
KNOW HOW im Bereich Parsing	ausreichend	/	/
KNOW HOW im Bereich Datenbanken	ausreichend	/	/
KNOW HOW im Bereich CI/CD	ausreichend	/	/
KNOW HOW im Bereich Containerization	ausreichend	/	/
Server zum Builden und Hosten vom Produkt	ausreichend	/	/
Development-Software (IDEs, SDKs, ...)	ausreichend	/	/

### 9.2 Personelle Ressourcen

#	Teammitglied	Personenstunden
1	Benjamin Komar	190
2	Nick Müllner	190
3	Nick Fischer	190
4	Yannik Wailzer	190
SUMME		760

### 9.3 Budget

#### 9.3.1 Auflistung der Aufwände für die Durchführung der Diplomarbeit

Pos.	Bezeichnung des Aufwands	Kosten
1	Serverkosten für 1 Jahr	EUR 314,40
2	Domainkosten für 1 Jahr	EUR 15,49
	Gesamtkosten	EUR 329,89

#### 9.3.2 Kostendeckung

Pos.	Beschreibung der Kostendeckung
1	Kostendeckung durch den Kooperationspartner (Fabasoft)
2	Kostendeckung durch das Projektteam

## 10 Geplante externe Kooperationspartner

Wie bereits erwähnt unter [1.1](#) wird das Team in Kooperation mit der Firma Fabasoft die Diplomarbeit durchführen. Dabei unterstützt die Firma Fabasoft das Projektteam in der Verwirklichung und Durchführung der ursprünglich von der Firma Fabasoft bereitgestellten Idee für das Projekt, der Erstellung eines Produktes für ihren Scrum-Prozess. Das Team erhält entsprechende Zugangsdaten zur Fabasoft Cloud, um darin ein Scrum-Projekt mit Testdaten anlegen zu können. Codeverwaltung (SVN / GIT) und notwendige Infrastruktur werden bereitgestellt. Auch kostenpflichtige Services wie Containeranbieter werden, wenn genutzt, von Fabasoft bezahlt.

## 11 Geplante Verwertung der Ergebnisse

Das Produkt soll in der Firma als Ersatz für das momentane Scrum-Tool, *Scrumtool3*, dienen. Außerdem wird es dort, so wie das momentane Tool, weiterentwickelt und gewartet. Das Produkt wird nach der Diplomarbeit an Fabasoft vollständig übergeben, während der Diplomarbeit liegt die Verantwortung ganz in den Händen des Teams. Das Team ist somit nicht zuständig für das Aufsetzen und der Wartung des Produktes.