

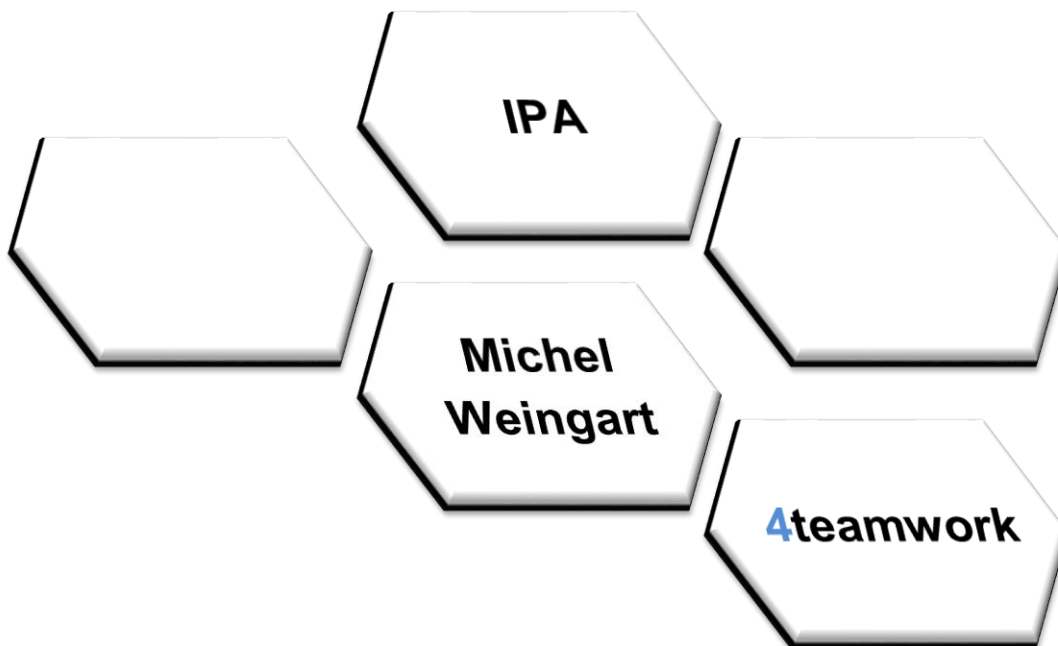
# IPA Dokumentation

---

## Automatische Installation von Demo-Webplattformen

<b>Auftraggeber</b>	4teamwork AG
<b>Projektname</b>	Automatische Installation von Demo-Webplattformen
<b>Autor</b>	Michel Weingart
<b>Ausgabedatum</b>	27.5.2014
<b>Version</b>	V 1.0

<b>Status</b>	In Arbeit	In Prüfung	Genehmigt, zur Nutzung
	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>



# Dokumentinformationen

## Änderungskontrolle, Prüfung, Genehmigung

In der Änderungskontrolle sind nur Punkte aufgelistet, welche vollständig in der Dokumentation enthalten sind, was zur Folge hat, dass der Arbeitsplan nicht zu hundert Prozent mit der Änderungskontrolle übereinstimmt.

Version	Datum	Name	Beschreibung
Vorlage	05.05.2014		IPA_Dokumentation_2014-final
V0.1	05.05.2014	Michel Weingart	Dokumenteröffnung
V0.2	06.05.2014	Michel Weingart	Abschluss der Phase Initialisierung
V0.3	07.05.2014	Michel Weingart	Mehrere Konzepte sind erarbeitet
V0.4	08.05.2014	Michel Weingart	Abschluss der Phase Konzept
V0.5	19.05.2014	Michel Weingart	Konfiguration für die Realisierung ist festgelegt
V0.6	20.05.2014	Michel Weingart	Skript „demo-teamraum.py“ ist erstellt
V0.7	21.05.2014	Michel Weingart	Skript „apply_profile“ ist erarbeitet
V0.8	22.05.2014	Michel Weingart	Profil und die Benutzeranleitung sind erstellt
V0.9	26.05.2014	Michel Weingart	Testfälle sind durchgeführt und Problembehebung implementiert
V1.0	27.05.2014	Michel Weingart	Abschluss der Phase Realisierung

Tabelle 1: Änderungskontrolle

## Verwendete Abkürzungen

Die wichtigsten Abkürzungen werden in der nachfolgenden Tabelle ausgeschrieben.

Abkürzung	Bedeutung
CMS	Content <b>M</b> anagement <b>S</b> ystem
IPA	Individuelle <b>p</b> raktische <b>A</b> rbeit
JSON	<b>J</b> ava <b>S</b> cript <b>O</b> bject <b>N</b> otation
PEP	<b>P</b> ython <b>E</b> nhancement <b>P</b> roposal
VM	<b>V</b> irtual <b>M</b> achine
XML	<b>E</b> xtensible <b>M</b> arkup <b>L</b> anguage

Tabelle 2: Verwendete Abkürzungen

Nachfolgend wird der Personenkreis betreffend die IPA aufgelistet.

Person	Kürzel
Michel Weingart	MW
Pascal Habegger	PH
Manuel J. Schaffner	MS
Verena Meinung,	VM
Yvan Kolly, Validexperte	YK
Josef Müller	JM
Mathias Leimgruber	ML
Thomas Buchberger	TB

**Tabelle 3: Personenkreis**

# Inhaltsverzeichnis

<b>Dokumentinformationen.....</b>	<b>2</b>
<b>Inhaltsverzeichnis.....</b>	<b>4</b>
<b>Abbildungsverzeichnis.....</b>	<b>7</b>
<b>Tabellenverzeichnis.....</b>	<b>7</b>
<b>Management Summary.....</b>	<b>8</b>
<b>Teil 1: Ablauf und Umfeld.....</b>	<b>9</b>
<b>1. Aufgabenstellung.....</b>	<b>10</b>
1.1 Ausgangslage.....	10
1.2 Auftragsformulierung .....	10
1.3 Mittel und Methoden .....	11
1.4 Projektmanagementplan .....	12
1.5 Projektrollen .....	13
<b>2. Vorkenntnisse .....</b>	<b>13</b>
<b>3. Vorarbeiten .....</b>	<b>14</b>
<b>4. Rahmenbedingungen.....</b>	<b>14</b>
<b>5. Firmenstandards .....</b>	<b>14</b>
<b>6. Organisation der IPA.....</b>	<b>15</b>
6.1 Datensicherung der IPA.....	16
<b>7. Meilensteine .....</b>	<b>17</b>
<b>8. Zeitplan .....</b>	<b>18</b>
<b>9. Arbeitsjournal.....</b>	<b>19</b>
9.1 Erster Tag: Montag, 05.05.2014 .....	19
9.2 Zweiter Tag: Dienstag, 06.05.2014 .....	21
9.3 Dritter Tag: Mittwoch, 07.05.2014 .....	23
9.4 Vierter Tag: Donnerstag, 08.05.2014 .....	25
9.5 Fünfter Tag: Montag, 19.05.2014.....	27
9.6 Sechster Tag: Dienstag, 20.05.2014.....	29
9.7 Siebter Tag: Mittwoch, 21.05.2014 .....	30
9.8 Achter Tag: Donnerstag, 22.05.2014 .....	31
9.9 Neunter Tag: Montag, 26.05.2014 .....	33
9.10 Zehnter Tag: Dienstag, 27.05.2014 .....	34
9.11 Arbeitszeit total.....	35
<b>10. Projektjournal .....</b>	<b>36</b>
<b>11. Abschlussbericht .....</b>	<b>42</b>
11.1 Vergleich Ist/Soll.....	42
11.2 Realisierung .....	42
11.3 Testbericht.....	43
11.4 Fazit zum Projekt.....	43
11.5 Persönliches Fazit .....	43

<b>12.</b>	<b>Unterschriften Teil 1.....</b>	<b>44</b>
	<b>Teil 2: Projektdokumentation.....</b>	<b>45</b>
<b>13.</b>	<b>Initialisierung.....</b>	<b>46</b>
13.1	Situationsanalyse .....	46
13.1.1	Plone .....	46
13.1.2	teamraum®.....	46
13.1.3	Github / GitLab .....	47
13.2	Studie Ist-Zustand / Soll-Zustand.....	48
13.3	Zeitraumen / Budget .....	50
13.4	Übergeordnete Ziele .....	50
13.5	Chancen .....	50
13.6	Anforderungen.....	50
13.7	Anwendungsbereich .....	51
13.8	Risikoanalyse .....	51
13.9	Varianten .....	51
13.9.1	Variante 1: VMware-Image .....	52
13.9.2	Variante 2: Skriptbasierte Installation.....	52
13.10	Variantenentscheid.....	53
13.10.1	Begründung .....	53
13.11	Lösungen suchen .....	53
13.12	Freigabe Konzept .....	55
<b>14.</b>	<b>Konzept.....</b>	<b>56</b>
14.1	Betriebskonzept.....	56
14.1.1	Repositories .....	56
14.1.2	demo_teamraum.py.....	57
14.1.3	Implementation in die bestehende Umgebung.....	58
14.1.4	Versionierung demo_teamraum.py .....	58
14.1.5	Betrieb auf dem Server.....	59
14.1.6	Anforderungen.....	59
14.2	Zugriffskonzept .....	59
14.2.1	Berechtigungen .....	60
14.3	Testkonzept.....	60
14.3.1	Ausgangslage.....	60
14.3.2	Testmethode.....	60
14.3.3	Testplanung.....	60
14.3.4	Testziele .....	61
14.3.5	Hilfsmittel.....	61
14.3.6	Testvorbereitung.....	61
14.3.7	Testdurchführung .....	61

14.3.8	Testfälle.....	61
14.4	Einführungskonzept.....	64
14.4.1	Anforderung.....	64
14.4.2	Durchführung.....	65
14.4.3	Betrieb.....	65
14.5	Freigabe Realisierung.....	66
<b>15.</b>	<b>Realisierung .....</b>	<b>67</b>
15.1	Legende .....	67
15.2	Bestehende Umgebung verifizieren .....	67
15.3	Konfiguration festlegen .....	67
15.4	Vorarbeiten.....	68
15.5	Skript „demo_teamraum.py“ erstellen .....	69
15.5.1	Hauptelemente Skript „demo_teamraum.py“ .....	69
15.5.2	Bibliotheken Skript „demo_teamraum.py“ .....	69
15.5.3	Wichtige Elemente des Quellcodes .....	69
15.5.4	Zusammenfassung .....	72
15.6	Profil „teamraum-demo“ definieren im Repository „eGov“ .....	73
15.6.1	Hauptelemente des Profils.....	73
15.6.1.1	teamraum-modules.json .....	73
15.6.1.2	01_organisation.json.....	74
15.6.2	Abhängigkeit.....	74
15.7	Skript „apply_profile.py“ erarbeiten in einem Branch des Repository „buildout- base“	75
15.8	Benutzeranleitung.....	75
15.9	Testprotokoll.....	75
15.10	Testauswertung / Testabschluss.....	76
15.11	Problemanalyse und -behebung .....	76
<b>16.</b>	<b>Quellenverzeichnis .....</b>	<b>78</b>
<b>17.</b>	<b>Glossar .....</b>	<b>78</b>
<b>18.</b>	<b>Anhang .....</b>	<b>81</b>

# Abbildungsverzeichnis

Abbildung 1: Hermes 5 IPA .....	11
Abbildung 2: Organigramm Projekt .....	12
Abbildung 3: Grober Grundriss einiger Büros .....	15
Abbildung 4: Arbeitsplatz Projektleiter .....	15
Abbildung 5: Kanban .....	16
Abbildung 6: Zeitplan.....	18
Abbildung 7: Fehler Zeitplan .....	27
Abbildung 8: Fehlermeldung nach git push.....	39
Abbildung 9: Zusammenhang der Systeme .....	47
Abbildung 10: Erläuterung Git .....	48
Abbildung 11: Releaseprozess Ist.....	49
Abbildung 12: Releaseprozess Soll .....	50
Abbildung 13: Brainstorming .....	54
Abbildung 14: Inhaltsstruktur.....	57
Abbildung 15: Zusammenhang Skripts, Repositories und Branches.....	58
Abbildung 16: Phasenmodell.....	60
Abbildung 17: Struktur .....	67
Abbildung 18: Bibliothek im Skript.....	69
Abbildung 19: Programmablaufplan .....	72
Abbildung 20: Grundaufbau des Profils .....	73
Abbildung 21: Module.....	73
Abbildung 22: Aufbau Inhaltsseite 1 .....	74
Abbildung 23: Aufbau Inhaltsseite 2.....	74

# Tabellenverzeichnis

Tabelle 1: Änderungskontrolle.....	2
Tabelle 2: Verwendete Abkürzungen .....	2
Tabelle 3: Personenkreis.....	3
Tabelle 4: Projektrollen.....	13
Tabelle 5: Vorkenntnisse .....	13
Tabelle 6: Datensicherung Git.....	16
Tabelle 7: Meilensteine .....	17
Tabelle 8: Ist/Soll .....	42
Tabelle 9: Terminalbefehle git.....	47
Tabelle 10: Systemziele .....	51
Tabelle 11: Nicht funktionale Anforderungen .....	51
Tabelle 12: Variantenentscheid.....	53
Tabelle 13: Anforderungen Skript „demo_teamraum.py“ .....	59
Tabelle 14: Nicht funktionale Anforderungen Skript „demo_teamraum.py“ .....	59
Tabelle 15: Berechtigungstabelle .....	60
Tabelle 16: Hauptziele Tests.....	61
Tabelle 17: Erläuterung der Struktur .....	73
Tabelle 18: Wichtigste Elemente der JSON-Datei .....	74
Tabelle 19: Testprotokoll .....	76
Tabelle 20: Wichtigste Inhaltstypen.....	81

# Management Summary

## Ausgangssituation

4teamwork bietet mehrere webbasierte Lösungen an. Um den Kunden die hohe Qualität der Lösungen gewährleisten zu können, werden diese in einem geordneten Releaseprozess immer weiterentwickelt. Die neusten Anpassungen und Features sollen den Kunden zu Testzwecken auf einer Demo-Plattform stets zur Verfügung stehen.

Gefundene Fehler können 4teamwork mitgeteilt und durch die Entwickler behoben werden. Die Demo-Plattform wird aber nicht regelmässig aktualisiert, was dazu führen kann, dass Releases auf produktiven Plattformen eingespielt werden, welche noch nicht genügend getestet worden sind.

Im Rahmen der IPA soll eine Lösung entwickelt werden, um die Demo-Plattform mit dem neusten Release in kürzester Zeit automatisiert aufsetzen zu können.

So soll gewährleistet werden, dass Releases vor der Freigabe besser getestet werden. Dies führt einerseits zu einer zeitlichen Einsparung beim Aufsetzen der Demo-Plattform und andererseits zu einer Qualitätssteigerung der Produkte von 4teamwork.

Die Kundenzufriedenheit steht in direktem Bezug zur Qualität der Produkte. Somit wird bei einer Steigerung der Qualität auch die Kundenzufriedenheit erhöht.

## Umsetzung

Um eine einfache Implementation in die bestehende Umgebung gewährleisten zu können, wurde im Rahmen eines Variantenentscheids, die Umsetzung mittels Skripts gewählt. Um eine realitätsnahe Umgebung für den Kunden bereitzustellen, wurde die Demo-Plattform teilweise nach Abbild der Webseite von der Gemeinde Belp nachgebaut.

## Ergebnis

Die im Rahmen dieser Arbeit entwickelten Skripts erlauben es, automatisiert die Demo-Plattform auf einem MacBook Pro aufzusetzen und Testbenutzer und Testinhalte zu erzeugen. Nach Projektabschluss werden die Skripts auf den Servern von 4teamwork installiert und in Betrieb genommen.



# Teil 1: Ablauf und Umfeld

Auftraggeber: 4teamwork AG

IPA Projektname: **Automatische Installation von Demo-Webplattformen**

Autor: Michel Weingart

# 1. Aufgabenstellung

Für mehrere webbasierte Standardlösungen sollen regelmässig (z.B. wöchentlich) und automatisiert öffentliche Demoinstallationen mit Beispieldaten im Internet verfügbar gemacht werden.

## 1.1 Ausgangslage

4teamwork entwickelt verschiedene webbasierte Standardlösungen: teamraum® Collab, teamraum® Intranet, teamraum® Web und OneGov GEVER. Diese Lösungen basieren auf dem Content Management System Plone und werden in einem geordneten Releaseprozess laufend erneuert und erweitert. Neue Funktionen werden verfügbar gemacht und von den Benutzern gemeldete Fehler und Unzulänglichkeiten werden korrigiert. Es existiert eine Demo-Plattform, über die sich interessierte Personen einloggen und diese unverbindlich testen können. Die Demo-Plattform wird von 4teamwork nur unregelmässig aktualisiert und ist daher meistens nicht auf dem neusten Stand. Zudem enthält sie kaum Beispieldaten.

Entwicklungs- und Testinstallationen der Standardlösungen werden von den Entwicklern von 4teamwork zwar immer wieder neu erstellt, allerdings sind diese nur lokal verfügbar und von aussen nicht zugänglich. Diese Installationen dienen vorwiegend automatisierten Softwaretests ohne Benutzerinteraktionen. Aus diesen Gründen werden neue Releases von teamraum® unter Umständen zu wenig von Menschen getestet (Human Testing), bevor sie freigegeben werden.

## 1.2 Auftragsformulierung

Mit dem Projektabschluss werden folgenden Resultate erwartet:

1. Es wird ein Konzept entwickelt, wie die Plattformen mit Beispieldaten und – benutzern, automatisiert installiert und betrieben werden können. Zwei Hauptvarianten sind dabei denkbar und müssen nach verschiedenen Kriterien bewertet werden:
  - 1.1 Bereitstellen eines VMware-Image mit einem vorbereiteten Release einer Webanwendung inkl. Beispieldaten und -benutzern. Das VMware-Image kann dann regelmässig neu geladen werden.
  - 1.2 Auf einer bestehenden Linux-Umgebung werden per Buildout (d.h. geskriptet) automatisiert neue Webinstallationen erstellt und gestartet.  
Die höher bewertete Variante soll umgesetzt werden (siehe 2).
2. Für eine der oben genannten Standardlösungen von teamraum® soll eine automatisierte Demo-Plattform im Internet errichtet werden, so dass interessierte Personen die Lösung selbständig antesten können. Die Demo-Plattform verwendet immer den neusten, freigegebenen Release und enthält Anwendungsfall bezogene Beispieldaten. Der Entscheid, welche Standardlösung automatisiert installiert werden soll, wird dem Lernenden überlassen. Für die gewählte Standardlösung müssen anwendungsbezogene Beispieldaten und -benutzer definiert werden. Diese Inhalte werden dann bei jeder Neuinstallation der Demo-Umgebung automatisch hinzugefügt.

3. Der Prozess, der verwendet wird, um die Demo-Plattformen von neuen Releases von teamraum© automatisiert zu installieren, soll auch für künftige Entwicklungs- und Testreleases eingesetzt werden können. So können sog. Release Candidates von teamraum© von einem kleineren Benutzerkreis über das Web manuell getestet werden. Damit lässt sich die Qualität eines teamraum© Release weiter steigern. Der geforderte Prozess muss technisch gut dokumentiert sein, damit auch andere Personen in Zukunft derartige Testsysteme bereitstellen können.

### 1.3 Mittel und Methoden

- Die Entwicklung und Dokumentation werden auf einem MacBook Pro durchgeführt. Dies stellt das Arbeitsgerät für den täglichen Gebrauch in der Firma 4teamwork AG dar.
- Die Dokumentation wird in Word2010 unter Mac OS X verfasst.
- Als Programmierungsumgebung wird der Texteditor Textmate verwendet.
- Gute Kenntnisse von X-HTML und XML sind erforderlich.
- Anwendungswissen von Plone und teamraum© ist erforderlich.
- Als Programmiersprachen kommen Python und Python-Skripte zum Einsatz. Das Content Management System „Plone“, welches für die webbasierten Plattformen verwendet wird, basiert auf Python.
- Die Projektdokumentation basiert auf der vereinfachten Projektmethode HERMES 5, welche für die Durchführung von IPAs optimal angepasst wurde. Die Meilensteine sind im Kapitel 7 definiert.

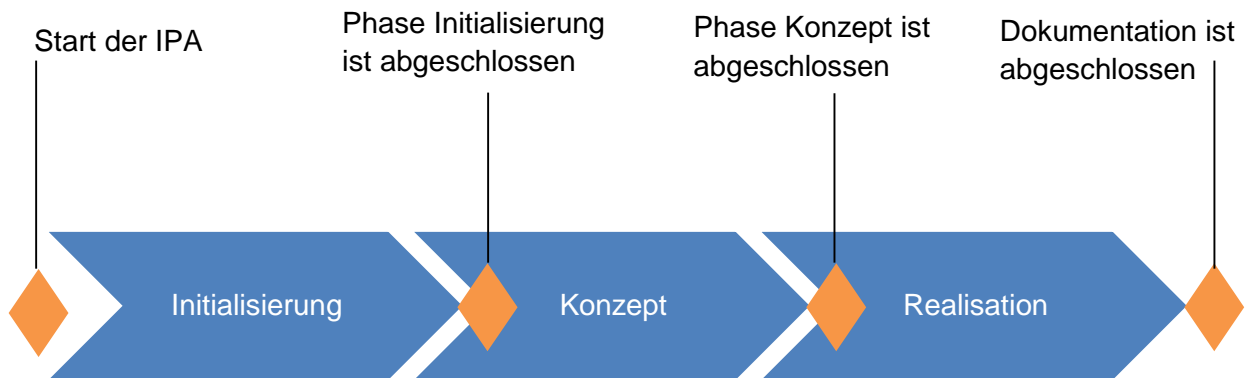


Abbildung 1: Hermes 5 IPA

## 1.4 Projektmanagementplan

Nachfolgend wird der Personenkreis betreffend die IPA in einem Organigramm dargestellt und die verschiedenen Projektrollen in einer Tabelle beschrieben.

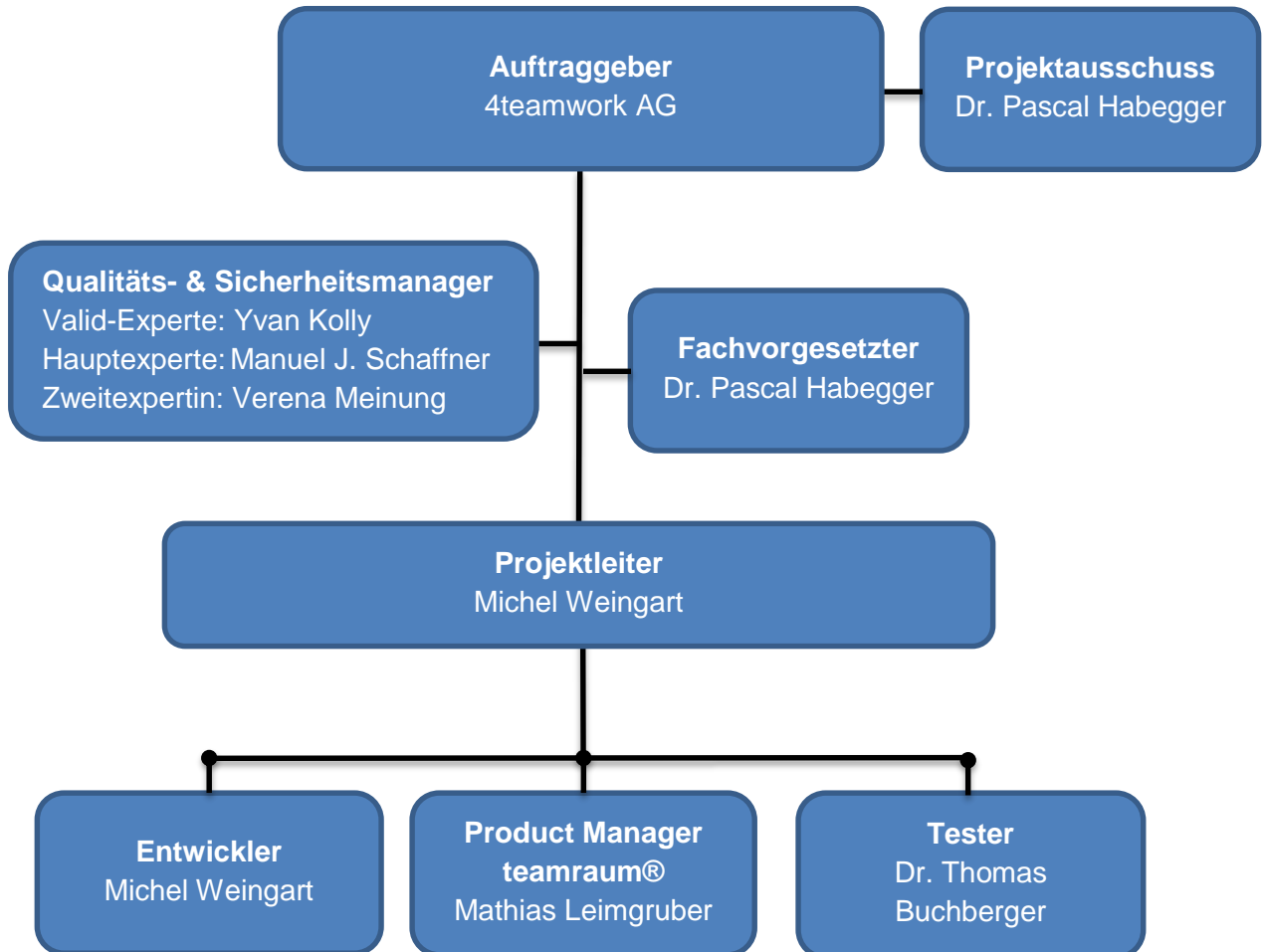


Abbildung 2: Organigramm Projekt

## 1.5 Projektrollen

Die verschiedenen Projektrollen werden in der nachfolgenden Tabelle beschrieben.

Rolle	Beschreibung
Auftraggeber	Er formuliert die Vision. Er ist der Finanzgeber des Projekts und nimmt die fertige Applikation ab.
Projektausschuss	Stellt das oberste beschlussfassende Gremium einer Projektorganisation dar. Er definiert Rahmenbedingungen und Anforderungen.
Valid-Experte	Er ist zuständig für die Validierung des Projekts.
Hauptexperte	Er ist zuständig für die externe Bewertung des Projekts und steht bei Fragen zur Verfügung.
Zweitexpertin	Sie ist bei der Bewertung der Dokumentation beteiligt.
Fachvorgesetzter	Er ist für die Betreuung des Auszubildenden zuständig.
Projektleiter	Er führt und koordiniert das komplette Projekt. Er teilt die Aufgaben in Arbeitspakete auf und ist zuständig für die Kommunikation innerhalb des Teams.
Entwickler	Er ist zuständig für das technische Design und die Implementierung der Software. Er ist somit für die technische Umsetzung der Anforderungen zuständig.
Product Manager teamraum®	Er dient zur Unterstützung des Auszubildenden in Sachen technische Umsetzung und allgemeine Fragen. Er ist ein Spezialist betreffend die Produkte von teamraum®
Tester	Er testet alle Komponenten gemäss den definierten Testfällen und erfasst die Ereignisse in den dazugehörigen Testprotokollen.

Tabelle 4: Projektrollen

## 2. Vorkenntnisse

Eine Übersicht über die Vorkenntnisse, welche für die Durchführung dieses Projekts relevant sind.

Anwendung	Bisheriger Einsatzbereich	Dauer
Webanwendung teamraum®	Schulung und Supportarbeiten	2 Jahre
Browser (Chrome, Firefox, Safari, Internet Explorer)	Alltag, Support	2 Jahre
Serveradministration Linux Grundwissen	Supportarbeiten	1 Jahr
Entwicklung mit Programmiersprache Python	Keine	2 Wochen
Plone Grundwissen	Supportarbeiten	1 Jahr
git	Kleinere Entwicklungen	0.5 Jahr
Microsoft Office	Tägliche Arbeiten	7 Jahre
HERMES 5	Bei kleineren Projekten in der Schule.	0.5 Jahre
X-HTML und XML	Bei kleineren Projekten in der Schule	1 Jahr

Tabelle 5: Vorkenntnisse

### 3. Vorarbeiten

Um die Durchführung des Projekts optimal umsetzen zu können, wurden folgende Vorarbeiten geleistet:

- Diverse Vorbesprechungen mit dem Fachvorgesetzten zur Definition des IPA-Themas.
- Einarbeitung in Plone, wie Webinstallation technisch mit Inhalten abgefüllt werden können.
- Besprechung mit einem Entwickler betreffend die Programmiersprache Python.
- Programmierung von Python, z.T. mit Unterstützung eines Entwicklers.
- Einrichtung des Arbeitsplatzes für die IPA.
- Erstellen eines Kanban-Plakats.
- Studium bestehender IPAs.

### 4. Rahmenbedingungen

Es sind folgende Rahmenbedingungen einzuhalten:

- Die Durchführung des Projekts dauert 10 Tage.
- Die IPA Dokumentation muss bis am 27.05.2014 um 18:00 auf der Plattform extranet.pkorg.ch hochgeladen werden
- Die Dokumentation ist einseitig auszudrucken.
- Eine ausgedruckte Version, mit Deckblatt Blau, ist dem Hauptexperten (A-Post) an die private Wohnadresse zu versenden.
- Eine ausgedruckte Version, mit Deckblatt Blau, ist der Zweitexpertin (A-Post) an die Geschäftsadresse zu versenden.
- Eine ausgedruckte Version, mit Deckblatt Gelb, ist dem Fachvorgesetzten abzugeben.
- Die definierten Termine (Besuchstage und Präsentation) müssen eingehalten werden.

### 5. Firmenstandards

Da bei der Firma 4teamwork AG keine Standards betreffend Vorlagen existieren, wird die Dokumentvorlage, welche von PkOrg zur Verfügung gestellt wird, genutzt. Für die Programmierung gelten partiell die Geltungsrichtlinien von PEP 8, welche unter <http://legacy.python.org/dev/peps/pep-0008/> aufgeführt sind und bei 4teamwork auch eingesetzt werden.

## 6. Organisation der IPA

Um die IPA ungestört durchführen zu können, wurde der Arbeitsplatz mit dem nötigen Zubehör vom Supportzimmer in ein meist ungenutztes Büro verlegt.

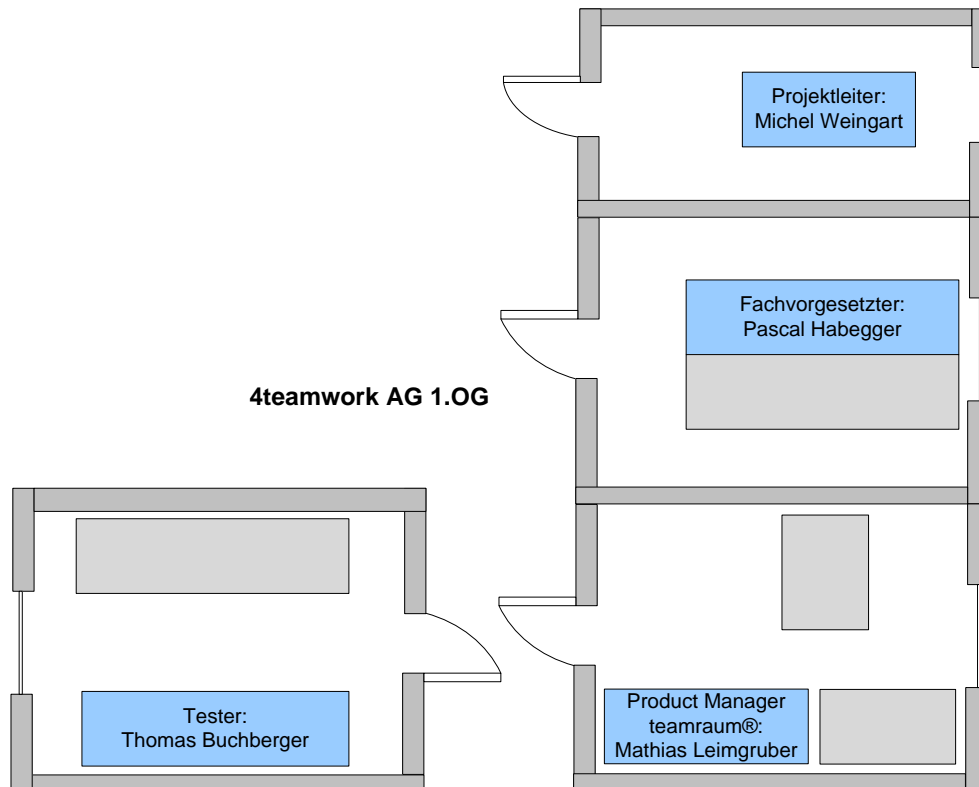


Abbildung 3: Grober Grundriss einiger Büros

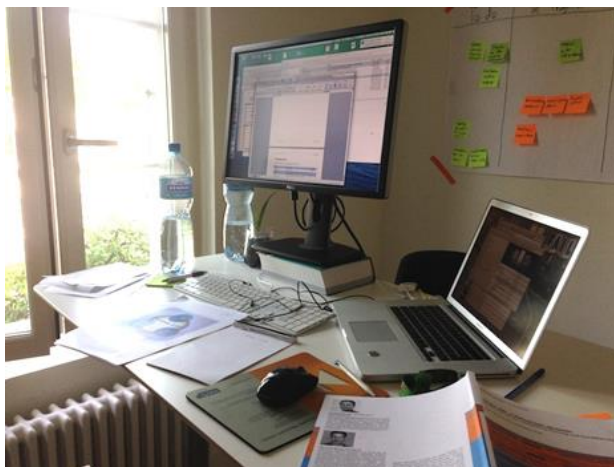


Abbildung 4: Arbeitsplatz Projektleiter

Zusätzlich zum Wechsel des Arbeitsplatzes wurde beschlossen, das Vorgehensmodell „Kanban“ einzusetzen. Dieses wurde bei der Firma 4teamwork vor ca. einem halben Jahr eingeführt und hat sich sowohl bei den Entwicklern als auch bei den Projektmanagern bewährt. Es dient zur vereinfachten Aufgabenteilung und schnelleren Erkennung von Problemen.



Abbildung 5: Kanban

## 6.1 Datensicherung der IPA

Die Datensicherung und die Dokumentablage erfolgen auf zwei Arten.

1. Mittels Repository auf Github.com.

Eine automatische Versionierung ist so gewährleistet, ein Datenverlust wird vermieden und der Datenschutz eingehalten. Zusätzlich werden getätigte Änderungen in einem geregelten Prozess kontrolliert und Fehler somit reduziert. Eine Beschreibung zu Github ist unter dem Kapitel 13.1.3 verfasst.

Name Repository	URL	Beschreibung
IPA_michel.weingart	<a href="https://github.com/4teamwork/IPA_michel.weingart">https://github.com/4teamwork/IPA_michel.weingart</a>	Ablage von: - Dokumenten - Quellcode - Bilder/Grafiken

Tabelle 6: Datensicherung Git

2. Mittels externer Festplatte

Ein Backup wird jeden Abend auf eine externe Festplatte gespeichert.



## 7. Meilensteine

In der IPA sind folgende vier Meilensteine definiert:

Datum	Beschreibung	Erreicht
05.05.2014	Start der IPA	Ja
06.05.2014	Phase Initialisierung ist abgeschlossen und Phase Konzept freigegeben	Ja
19.05.2014	Phase Konzept ist abgeschlossen und Phase Realisierung freigegeben	Ja
22.05.2014	Abschluss Dokumentation	Ja

**Tabelle 7: Meilensteine**

8. Zeitplan

Soll-Arbeitszeit	<div></div>	Mo. 05.05.2014	Di. 06.05.2014	Mi. 07.05.2014	Do. 08.05.2014	Mo. 19.05.2014	Di. 20.05.2014	Mi. 21.05.2019	Do. 22.05.2014	Mo. 26.05.2014	Di. 27.05.2014
Ist-Arbeitszeit	<div></div>										
Meilensteine	<div></div>										
Unterbruch der IPA	<div></div>										
Tätigkeit	Soll / Ist										
Start IPA	7.5 / 8.5										
Kickoff											
Git Repository erstellen											
Dokument eröffnen											
Dokumentstruktur anpassen / Standardeintragungen (Kunde / Projektname...)											
Zeitplan erstellen											
Aufgabenstellung erfassen											
Vorkenntnisse / Vorarbeiten / Firmenstandards / Meilensteine und Organisation der IPA erfassen											
Initialisierung	7.5 / 7.0										
Studie Ist / Soll											
Vorgehensziele / Systemziele / Anwendungsbereich / Anforderungen											
Variantenentscheid											
Informationssicherheit und Datenschutz / Lösungen suchen											
Freigabe Konzept											
Konzept	18.5 / 19.0										
Allgemeine Anforderungen											
Systemstruktur											
Zugriffskonzept											
Betriebskonzept											
Testkonzept											
Einführungskonzept											
Zusammenfassung Konzept											
Freigabe Realisierung											
Expertenbesuch 1											
Realisierung	34.0 / 37.0										
Bestehende Umgebung verifizieren											
Konfiguration festlegen											
Installationsskript erstellen											
Profil definieren in einem Branch des Pakets "eGov"											
Runskript erarbeiten in einem Branch des Pakets "buildout-base"											
Lösung testen / Vorabnahme											
Programmablauf beschreiben											
Anleitung erstellen											
Expertenbesuch 2											
Systemumgebung testen											
Problemanalyse und -behebung											
Abschluss IPA	7.5 / 10.5										
Dokumentation durchlesen/kontrollieren											
Fazit / Management Summary erfassen											
Dokument abschliessen / Drucken											
Sonstiges											
Eintrag im Arbeitsjournal											
Projektjournal(laufend)											
Reserve: 0.5 pro Tag	5										
Gesamtaufwand	80 / 82										

Abbildung 6: Zeitplan

## 9. Arbeitsjournal

Im Arbeitsjournal sind die wichtigsten Informationen chronologisch gesammelt. Täglich werden das Vorgehen, fremde Hilfestellungen und besondere Vorkommnisse dokumentiert.

### 9.1 Erster Tag: Montag, 05.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Kickoff-Meeting</i> Das Kickoff-Meeting habe ich mit dem Product Manager teamraum®, Mathias Leimgruber, durchgeführt. Das Projekt kann beginnen.	MW/ML	0.5	0.5
<i>Git Repository</i> Das Git Repository habe ich unter „ <a href="https://github.com/4teamwork">https://github.com/4teamwork</a> “ mit dem Namen „IPA_michel.weingart.git“ erstellt um die Dokumentversionen und sonstige Daten hochladen zu können.	MW	0.5	0.5
<i>Dokumentation eröffnen</i> Ich habe die Dokumentation eröffnet.	MW	0.5	0.5
<i>Standardeintragungen in der Dokumentation</i> Ich habe die ersten kleinen Eintragungen in der Dokumentation vorgenommen, wie z.B. Kopf- und Fusszeile.	MW	0.5	0.5
<i>Zeitplan erstellen</i> Den Zeitplan habe ich erstellt und die Soll-Zeiten eingetragen.	MW	2.0	2.5
<i>Aufgabenstellung erfassen</i> Die Aufgabenstellung zum Projekt habe ich erfasst.	MW	1.0	1.5
<i>Vorkenntnisse / Vorarbeiten / Firmenstandards / Organisation des Projekts erfassen</i> Elemente habe ich der Dokumentation hinzugefügt.	MW	1.0	1.0
<i>Studie Ist / Soll</i> Eine kurze Reflektion betreffend des Ist- und Soll-Zustandes habe ich gemacht.	MW	0.5	0.5
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>8.5</b>
<b>Tagesverlauf</b>			
Das Projekt konnte ich nach dem Kickoff-Meeting erfolgreich starten. Nach kleineren Problemen betreffend das Git Repository, habe ich die Dokumentation eröffnet.			

Einige Probleme bereitete mir auch das Erstellen des Zeitplans. Dieser beanspruchte mehr Zeit als eingeplant, da ich Excel zwar von Windows her sehr gut kenne, jedoch selten auf dem MacBook Pro Excel verwende, welches anders aufgebaut ist. Das Schätzen des Zeitaufwands für die verschiedenen Phasen hatte ich mir einfacher vorgestellt, hoffe aber, dass die Schätzungen realistisch sind.

### Hilfestellungen

Ich habe zu Beginn lokal eine Ordnerstruktur im neuen Git Repository erstellt. Mit dem Befehl „git status“ sind keine Änderungen erkannt worden. Also habe ich bei einem Entwickler nachgefragt, ob Ordner in einem Git Repository nicht als neue Eintragungen erkannt werden. Nun habe ich erfahren, dass diese nicht erkannt werden, sondern nur Dateien.

### Reflexion

Die heutigen Ziele habe ich grundsätzlich erreicht und der Start verlief sehr gut. Ich konnte ohne grosse Probleme mit der Dokumentation beginnen.

Das Einarbeiten in das Programm Excel auf dem Mac hat Zeit benötigt, welche ich nicht zur Verfügung habe. Die bestehenden Unterschiede von Software unter Windows und unter Mac sollte ich in Zukunft besser berücksichtigen.

### Nächste Schritte

Aufgaben, welche ich für morgen geplant habe:

- Fertigstellung der Voranalyse
- Variantenentscheid durchführen
- Risikoanalyse durchführen
- Freigabe für das Konzept einholen

## 9.2 Zweiter Tag: Dienstag, 06.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Studie Ist / Soll</i> Den aktuellen und zu erwartenden Zustand habe ich aufgezeigt und den Prozess abgebildet.	MW	1.5	1.0
<i>Vorgehensziele / Systemziele / Anforderungen</i> Ich habe mehrere Elemente der Dokumentation hinzugefügt.	MW	1.5	1.5
<i>Variantenentscheid</i> Die bessere Variante konnte ich anhand des Variantenentscheids eruieren.	MW	1.5	1.5
<i>Informationssicherheit und Datenschutz / Lösungen suchen</i> Ich habe weitere Elemente der Dokumentation hinzugefügt.	MW	1.5	1.5
<i>Freigabe Konzept</i> Die Phase Konzept wurde freigegeben.	MW/PH	0.5	0.5
<i>Eintrag Arbeitsjournal</i> Ich habe das Arbeitsjournal nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>7.0</b>

### Tagesverlauf

Den Ist / Soll Vergleich konnte ich gut abschliessen und grafisch darstellen. Die bessere Variante hat sich deutlicher herauskristallisiert, als ich dies angedacht hatte, was den Entscheid für die Variante per Skript sehr vereinfachte. Auch war die Lösungssuche mit der Methode „Brainstorming“ sehr aufschlussreich und ich konnte die wichtigsten Kriterien daraus gewinnen. Hingegen bereitete mir das Erfassen der Vorgehensziele / Systemziele und der Anforderungen Mühe, da die Anforderungen dazu für mich nicht klar ersichtlich waren.

### Hilfestellungen

Für das Einfüllen der Vorgehensziele / Systemziele und Anforderungen konnte ich mir einige Tipps vom Product Manager teamraum®, Mathias Leimgruber, holen. Auch für die Auswahl der Kriterien für den Variantenentscheid habe ich kurz seine Hilfe beansprucht.

Meinem Hauptexperten habe ich eine Fragestellung betreffend Änderungen des Zeitplans zugestellt. Mir war nicht klar, wie eine Änderung erfasst werden soll und ob der Zeitplan überhaupt angepasst werden darf. Falls der Hauptexperte die Frage nicht beantworten kann, werde ich diese beim ersten Expertenbesuch stellen.

### Reflexion

Trotz des Zeitgewinns bei der Studie zum Ist- / Soll- Vergleich habe ich dort folgende Idee zur Verbesserung: Die Aufteilung der Arbeiten über mehrere Tage habe ich nicht als förderlich empfunden und werde dies in Zukunft wenn möglich vermeiden. Die Studie habe ich am Montag angefangen und am Dienstag beendet. Der Unterbruch verlangte, dass ich mich zuerst

wieder in das Thema einarbeitete.

### **Nächste Schritte**

Ich werde mich auf den Erstbesuch des Hauptexperten vorbereiten und Fragen bezüglich der IPA aufschreiben um ihm diese stellen zu können.

Weitere Aufgaben, welche ich für morgen geplant habe:

- Situationsanalyse
- Allgemeine Anforderungen
- Systemarchitektur ermitteln
- Zugriffskonzept erstellen
- Erarbeitung des Betriebskonzepts

### 9.3 Dritter Tag: Mittwoch, 07.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Situationsanalyse</i> Die Situationsanalyse betreffend das Konzept habe ich erstellt.	MW	1.5	1.5
<i>Allgemeine Anforderungen</i> Die Anforderungen, damit das Skript lauffähig ist, konnte ich definieren.	MW	0.5	0.5
<i>Systemarchitektur ermitteln</i> Die Systemarchitektur vom Server Poseidon habe ich ermittelt und aufgezeigt.	MW	1.0	1.0
<i>Zugriffskonzept erstellen</i> Die Erarbeitung habe ich begonnen, jedoch noch nicht abgeschlossen.	MW	1.5	1.5
<i>Erarbeitung des Betriebskonzepts</i> Das Betriebskonzept habe ich fertiggestellt.	MW	1.0	1.0
<i>Expertenbesuch</i> Der Expertenbesuch hat stattgefunden. Details dazu, können dem Projektjournal entnommen werden.	MW/MS/VM/PH	1.0	2.0
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>8.5</b>
<b>Tagesverlauf</b>			
<p>Die Erarbeitung des Zugriffskonzeptes konnte ich nicht fertigstellen, da ich meinem Fachvorgesetzten, Pascal Habegger, noch einige Fragen betreffend die Datenstruktur (Beispielinhalt auf der Plattform) stellen musste. Dieser war aber am Morgen in einer externen Besprechung. Um die Zeit sinnvoll zu nutzen, habe ich deshalb die Erarbeitung des Betriebskonzepts auf den Vormittag verschoben.</p> <p>Für die Erarbeitung des Zugriffskonzepts habe ich 1.5 Stunde aufgewendet, konnte dies aber noch nicht ganz abschliessen und werde dies morgen erledigen und als Punkt im Arbeitsjournal vermerken.</p> <p>Heute erfolgte der Besuch des Hauptexperten Manuel J. Schaffner und der Zweitexpertin Verena Meinung. Weitere Informationen zu dieser Sitzung sind im Projektjournal festgehalten.</p>			
<b>Hilfestellungen</b>			
<p>Heute Morgen habe ich noch die Rückmeldung des Experten betreffend die gestrige Fragestellung (Anpassung Zeitplan) erhalten. Er hat mich darauf hingewiesen, dass die Soll-Zeiten angepasst werden können, hingegen die Ist-Zeiten so belassen werden sollten.</p>			

Murat Tokmak, Projektleiter bei 4teamwork, hat mir betreffend den Aufbau eines Betriebskonzeptes Hilfestellung gegeben.

Betreffend Sicherungsmechanismen habe ich beim Product Manager teamraum®, Mathias Leimgruber, Hilfestellung zu folgenden Themen erhalten:

- Auf welchem Master wird der Branch erstellt?
- Werden weitere Branches von bestehenden Branches erstellt oder ist dies mit einzelnen Commits umzusetzen?

### **Reflexion**

Das Zusammentragen der gewünschten Informationen hat gut funktioniert. Aufgrund des Expertenmeetings habe ich bemerkt, dass ich zu wenig Zeit für die Überarbeitung der Dokumentation und die Überarbeitung des Zeitplans eingeplant habe. Ich schätze, dass dies noch einen Mehraufwand von 3 Stunden mit sich bringen wird.

### **Nächste Schritte**

Am Donnerstag werde ich den Zeitplan anpassen, wie dies an der Expertensitzung besprochen wurde. Auch werde ich Zeit investieren müssen, die Dokumentation grundlegend neu aufzubauen.

Weitere Aufgaben, welche ich für morgen geplant habe:

- Erarbeitung des Testkonzepts
- Abschluss der Konzeptphase und Freigabe der Realisierungsphase



## 9.4 Vierter Tag: Donnerstag, 08.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Anpassung des Zeitplans</i> Den Zeitplan habe ich teilweise überarbeitet.	MW	0.0	1.0
<i>Erarbeitung des Zugriffskonzepts</i> Die Erarbeitung des Zugriffskonzepts habe ich abgeschlossen.	MW	0.0	1.0
<i>Erarbeitung des Testkonzepts</i> Das Testkonzept mit den verschiedenen Testfällen dazu habe ich erstellt.	MW	2.0	1.5
<i>Erarbeitung des Einführungskonzepts</i> Das Konzept zur Einführung habe ich erarbeitet.	MW	2.0	1.0
Zusammenfassung des Konzepts Die Phase Konzept habe ich zusammengefasst.	MW	2.0	0.5
<i>Überarbeitung Dokumentation</i> Einige Punkte der Dokumentation habe ich überarbeitet.	MW	0.0	1.0
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.5	1.0
<b>Total:</b>		<b>7.5</b>	<b>7.0</b>
<b>Tagesverlauf</b>			
<p>Das Testkonzept ist mir sehr gut gelungen und ich konnte eine halbe Stunde bei der Erarbeitung einsparen. Auch bei der Zusammenfassung des Konzepts habe ich zu viel Zeit eingeplant und konnte diese schon nach einer halben Stunde abschliessen. Den Zeitgewinn habe ich in die Überarbeitung der Dokumentation und des Konzepts investiert. Die Überarbeitung konnte ich noch nicht ganz abschliessen.</p> <p>Die Entscheidung, dass das Skript auf meinem lokalen MacBook Pro funktionieren muss, hat dazu beigetragen, dass das Einführungskonzept nicht mehr in demselben Masse benötigt wird wie vorgesehen, weshalb ich mich dort recht kurz gefasst habe.</p>			
<b>Hilfestellungen</b>			
Heute habe ich keine Hilfestellung benötigt.			
<b>Reflexion</b>			
<p>Heute konnte ich produktiv arbeiten, da ich mich über den Mittag kurz sportlich betätigt habe. Ich konnte am Nachmittag die Produktivität auf einem hohen Niveau halten.</p>			
<b>Nächste Schritte</b>			
<p>Der nächste Tag, an welchem ich für die IPA arbeiten kann, ist erst der Montag, 19.05.2014, da ich während einer Woche die Computerschule Bern CSBE besuche. Ich hoffe, dass ich nach diesem Unterbruch gut an das Projekt anschliessen kann. Die nächsten Schritte, welche ich</p>			

geplant habe, sind:

- Freigabe der Phase Realisierung
- Bestehende Umgebung verifizieren
- Konfiguration festlegen
- Skript „demo\_teamraum.py“ erstellen (noch nicht fertig)

## 9.5 Fünfter Tag: Montag, 19.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Anpassung des Zeitplans</i> Die letzten Anpassungen, welche am Expertenmeeting besprochen wurden, habe ich vorgenommen.	MW	0.0	0.5
<i>Freigabe der Phase Realisierung</i> Die Phase Realisierung wurde freigegeben.	MW	3.0	0.5
<i>Verifizierung der bestehenden Umgebung</i> Die Verifizierung der bestehenden Umgebung konnte ich abschliessen.	MW	1.0	1.0
<i>Festlegen der Konfiguration für die Realisierung</i> Die Konfiguration für die Realisierung habe ich festgelegt.	MW	1.0	1.0
<i>Überarbeitung der Dokumentation</i> Mehrere Punkte der Dokumentation habe ich überarbeitet.	MW	0.0	2.0
<i>Erarbeitung des Skripts „demo_teamraum.py“</i> Ich habe die Erarbeitung des Skripts begonnen.	MW	1.5	1.5
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total</b>		<b>7.5</b>	<b>7.5</b>

### Tagesverlauf

Ein kleiner aber wirkungsvoller Fehler in meinem Zeitplan hat dazu geführt, dass am Morgen nur die Freigabe der Realisierung durchgeführt werden musste.

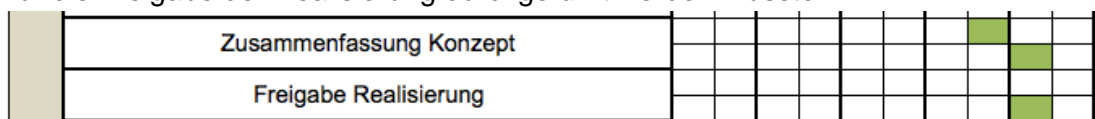


Abbildung 7: Fehler Zeitplan

Deshalb habe ich die Verifizierung der Umgebung und das Festlegen der Konfiguration auf den Vormittag verschoben. Für beide Beschreibungen konnte ich je eine halbe Stunde gutmachen. Bis am Mittag habe ich deshalb die überschüssige Zeit genutzt um die Dokumentation zu überarbeiten.

Am Nachmittag habe ich begonnen, das Skript demo-teamraum.py abzufüllen. Recht schnell wurde folgender Fehler geworfen:

SyntaxError: Non-ASCII character '\xc3' in file demo-teamraum.py on line 1, but no encoding declared.

Nach Konsultation verschiedener Google-Quellen war schnell klar, dass auf der ersten Zeile der verwendete Zeichensatz deklariert werden muss.

### Hilfestellungen

Betreffend optionalen Parameter zum Skript habe ich beim Entwickler, Jonas Baumann, eine

kleine Hilfestellung erhalten. Morgen werde ich versuchen, dem Skript eine Option mitzugeben.

### **Reflexion**

Bei künftigen Projekten muss ich den Zeitplan besser kontrollieren. Der Fehler war recht offensichtlich und hätte vermieden werden können. Das Gute des Fehlers ist, dass ich viele Punkte der Dokumentation überarbeiten konnte. Der Start der Erarbeitung des Skripts ist mir geglückt.

### **Nächste Schritte**

Morgen werde ich mich ausschliesslich mit dem Skript befassen und versuchen, dies wenn möglich fertigzustellen.

## 9.6 Sechster Tag: Dienstag, 20.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Erarbeitung des Skripts „demo-teamraum.py“</i> Das Skript konnte ich fertigstellen und die Hauptfunktionen sind gewährleistet. Falls Zeit bleibt, werde ich das Skript noch verbessern.	MW	6.5	6.0
<i>Überarbeiten der Systemstruktur und des Testkonzepts</i> Die Systemarchitektur habe ich in Systemstruktur umbenannt. Einige Punkte musste ich der Systemstruktur hinzufügen, da diese für das Verständnis des Projekts notwendig sind. Das Testkonzept habe ich nochmals überarbeitet.	MW	0.0	1.5
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>8.5</b>
<b>Tagesverlauf</b>			
Heute habe ich mich fast ausschliesslich mit der Erarbeitung des Skripts befasst. Die Arbeit hat mir sehr gut gefallen. Ich habe nicht gedacht, dass diese so gut voranschreiten würde. Der Systemstruktur musste ich, zum allgemeinen Verständnis, noch einige Punkte hinzufügen.			
<b>Hilfestellungen</b>			
Trotz gestriger kurzer Hilfestellung betreffend Skript-Optionen, musste ich nochmals bei Jonas Baumann nachfragen. Ich wusste nicht, wie ich dem OptionParser einen Standardwert mitgeben kann.			
<b>Reflexion</b>			
Es war sehr hilfreich, dass ich mich, als Vorbereitung auf diese IPA, mit Entwicklern und alleine mit der Programmiersprache Python befasst habe. Dies hat den Arbeitsaufwand am Skript deutlich reduziert.			
<b>Nächste Schritte</b>			
Morgen werde ich ein Profil im Branch des Repository „eGov“ definieren. Zusätzlich werde ich das Skript „apply_profile.py“ erstellen, wobei die meisten Konfigurationen diesbezüglich vom Product Manager teamraum®, Mathias Leimgruber, durchgeführt werden müssen. Weitere Schritte, welche ich für morgen geplant habe sind: <ul style="list-style-type: none"> <li>• Vorbereitung des zweiten Expertenbesuchs</li> <li>• Lösung kurz testen</li> </ul>			

## 9.7 Siebter Tag: Mittwoch, 21.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Profil definieren im Branch des Repository „eGov“</i> Profil und Inhalte konnte ich definieren, Benutzer noch nicht.	MW	2.5	2.5
<i>Skript „apply_profile.py“ im Branch des Repository buildout-base erstellen.</i> Das Skript habe ich mit dem Product Manager teamraum®, Mathias Leimgruber, erarbeitet.	MW/ML	3.0	3.5
<i>Lösung testen</i> Die Lösung konnte ich aus Zeitmangel nicht testen.	MW	0.5	0.0
<i>Vorbereitung des zweiten Expertenbesuchs</i> Ich habe Fragen aufgeschrieben, welche ich dem Experten morgen stellen werde.	MW	0.5	0.5
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>7.5</b>

### Tagesverlauf

Heute konnte ich nicht alle Arbeiten abschliessen, welche ich geplant habe. Das Profil habe ich definiert und Inhalte abgefüllt. Leider hat sich herausgestellt, dass Benutzer nicht auf dieselbe Weise erstellt werden können. Deshalb werde ich morgen das Erstellen der Benutzer noch mit dem Product Manager teamraum®, Mathias Leimgruber, erarbeiten. Das Skript „apply\_profile.py“ habe ich auch mit ihm erarbeitet und implementiert, da dies technisch zu anspruchsvoll war. Das erste Testen konnte ich aus Zeitmangel nicht durchführen. Deshalb habe ich dies weggelassen und werde mich bei der Einführungsphase auf das Testen konzentrieren.

### Hilfestellungen

Der Product Manager teamraum®, Mathias Leimgruber, hat mich beim Erarbeiten des Skripts „apply\_profile.py“ geführt.

### Reflexion

Ich habe zu wenig Zeit für die Realisierung eingeplant und konnte keine Tests durchführen. Ich hoffe aber, dass ich diesen Zeitverlust wieder aufholen kann.

### Nächste Schritte

Für morgen habe ich folgende Schritte geplant:

- Fertigstellen des Profils
- Programmablaufplan und Benutzeranleitung erstellen
- Zweiter Expertenbesuch
- Erste Tests auf der Systemumgebung

## 9.8 Achter Tag: Donnerstag, 22.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Programmablaufplan erstellen</i> Den Programmablaufplan konnte ich erstellen.	MW	1.0	1.0
<i>Zweiter Expertenbesuch</i> Der zweite Expertenbesuch hat stattgefunden. Details dazu können dem Projektjournal entnommen werden.	MW/MS/ PH	1.5	1.0
<i>Benutzeranleitung erstellen</i> Die Benutzeranleitung habe ich erstellt.	MW	2.5	2.0
<i>Überarbeitung der Dokumentation</i> Mehrere Punkte der Dokumentation habe ich überarbeitet.	MW	0.0	1.5
<i>Fertigstellen des Profils</i> Das Profil konnte ich mit Hilfe des Product Managers teamraum®, Mathias Leimgruber, fertigstellen.	MW/ML	0.0	1.0
<i>Freigabe der Phase Einführung</i> Die Phase Einführung wird nicht mehr benötigt.	MW	0.5	0.0
<i>Systemumgebung testen</i> Zwei Tests wurden durchgeführt und dokumentiert.	MW/TB	1.0	1.0
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	0.5
<b>Total:</b>		<b>7.5</b>	<b>8.0</b>
<b>Tagesverlauf</b>			
<p>Den Programmablaufplan habe ich abgeschlossen und werde eventuell noch einen kurzen Text dazu verfassen.</p> <p>Am Morgen hat der zweite Besuch des Hauptexperten wie geplant stattgefunden. Nach der Besprechung habe ich die abgearbeiteten Punkte im Projektjournal festgehalten.</p> <p>Den restlichen Morgen habe ich damit verbracht, die Benutzeranleitung zu erstellen und die Dokumentation zu überarbeiten. Am Nachmittag konnte ich mit dem Product Manager teamraum®, Mathias Leimgruber, in kurzer Zeit das Profil fertigstellen. Die ersten Tests konnten vom Tester, Thomas Buchberger, durchgeführt werden.</p>			
<b>Hilfestellungen</b>			
Den grössten Teil der Erarbeitung des Profils hat der Product Manager teamraum®, Mathias Leimgruber, übernommen, da dies für mich technisch zu anspruchsvoll war.			
<b>Reflexion</b>			
Die Benutzeranleitung ist mir gut gelungen und ich konnte diese vor dem geplanten Termin abschliessen.			

Wahrscheinlich werde ich am Montag mehr Zeit als vorgesehen in das Projekt investieren, damit ich dieses am Dienstag abschliessen kann. Die Dokumentation werde ich über das Wochenende mehreren Personen zum Durchlesen geben. Die Korrekturen werde ich am Montag übernehmen.

#### **Nächste Schritte**

Für den nächsten Montag habe ich folgende Arbeiten geplant:

- Fertigstellung der Tests
- Problemanalyse und –behebung in Bezug auf die Testfälle
- Korrekturen übernehmen.



## 9.9 Neunter Tag: Montag, 26.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Abschliessen der Tests</i> Die Tests wurden alle durchgeführt und die Testauswertung konnte ich erstellen.	MW/TB	3.0	2.5
<i>Problemanalyse und -behebung</i> Probleme habe ich behoben und Verbesserungen wahrgenommen.	MW	3.5	1.5
<i>Überarbeitung der Dokumentation</i> Mehrere Punkte der Dokumentation habe ich überarbeitet und die Korrekturen eingetragen.	MW	0.0	4.0
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgeführt.	MW	1.0	1.0
<b>Total:</b>		<b>7.5</b>	<b>9.0</b>
<b>Tagesverlauf</b>			
<p>Die Testfälle wurden alle gut abgearbeitet und waren, wie erwartet, nicht alle erfolgreich. Für die Durchführung wurde weniger Zeit beansprucht, als ich dies eingeplant hatte.</p> <p>Die JSON- Datei, welche die Inhalte vorgibt, konnte ich innert kurzer Zeit anpassen und einige Verbesserungen einbringen. Die Dokumentation habe ich noch an vielen Stellen angepasst und die gemeldeten Korrekturen eingetragen. Nun bin ich fast zufrieden mit dem Inhalt.</p>			
<b>Hilfestellungen</b>			
<p>Heute hatte ich eine kleine Hilfestellung von Jonas Baumann betreffend Vorgehen eines fehlgeschlagenen Tests. Die Erläuterungen dazu sind im Projektjournal nachzulesen.</p>			
<b>Reflexion</b>			
<p>Mein Vorgehen bei der Problemanalyse und Problembehebung war gut. Ich habe die Probleme schnell erkannt, dokumentiert und konnte diejenigen, welche ich beheben musste, beheben.</p>			
<b>Nächste Schritte</b>			
<p>Morgen ist der letzte Tag, an welchem ich für die IPA arbeite. Als Erstes werden ich die offenen Punkte der Dokumentation (Fazit und Management Summary) erarbeiten und anschliessend das Dokument abschliessen und drucken lassen. Um nicht in Verzug zu kommen, werde ich früh am Morgen an meinem Arbeitsplatz sein. Da mir noch mehrere Stunden, welche ich als Reserve eingeplant habe, zur Verfügung stehen, werde ich die Dokumentation nochmals auf Fehler überprüfen und diese korrigieren.</p>			

## 9.10 Zehnter Tag: Dienstag, 27.05.2014

Tätigkeiten	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
<i>Management Summary und Fazit erfassen</i> Die beiden Punkte habe ich erfasst.	MW	1.5	2.0
<i>Überarbeitung der Dokumentation</i> Ich habe die Dokumentation nochmals durchgelesen und Anpassungen vorgenommen	MW	3.0	6.0
<i>Dokument abschliessen und Drucken</i> Ich habe die Dokumentation abgeschlossen und drucke diese nun.	MW	2.0	2.0
<i>Eintrag Arbeitsjournal</i> Das Arbeitsjournal habe ich nachgetragen.	MW	1.0	0.5
<b>Total:</b>		<b>7.5</b>	<b>10.5</b>
<b>Tagesverlauf</b>			
Das Erfassen des Management Summaries hat mehr Zeit beansprucht, als ich eingeplant habe. Es war schwierig, die ganze Arbeit auf einer Seite zusammenzufassen. Die Dokumentation konnte ich gut abschliessen und ich bin zufrieden mit dem Endresultat.			
<b>Hilfestellungen</b>			
Heute benötigte ich keine Hilfestellungen.			
<b>Reflexion</b>			
Ich bin froh, dass ich eine Reservezeit eingeplant habe und ich kann dies unbedingt empfehlen. Mehrmals konnte ich davon Gebrauch machen und gerade für heute konnte ich die Zeit nutzen, um die Dokumentation nochmals durchzugehen und anzupassen.			
<b>Nächste Schritte</b>			
Heute muss ich noch die Dokumentation auf das Extranet von PkOrg hochladen. Anschliessend werde ich mich auf den Vortrag, welcher am 10.06.2014 stattfinden wird, vorbereiten.			

## 9.11 Arbeitszeit total

Die effektiv geplante Arbeitszeit beträgt laut Zeitplan 80 Stunden. Da ich Reservezeit einplant habe, beläuft sich der geplante Aufwand laut Arbeitsjournal nur auf 75 Stunden.

Totaler Zeitaufwand	Person	Aufwand geplant (Std)	Aufwand effektiv (Std)
	MW	75 +5	82
<b>Reflexion</b>			
Um eine Inhaltlich gute Dokumentation abzugeben, bedarf es mehrerer Korrekturen und Anpassungen. Die dafür benötigte Zeit habe ich ein bisschen unterschätzt, weshalb sich der effektive Aufwand nicht mit dem geplanten Aufwand deckt.			

## 10. Projektjournal

Im Projektjournal werden Informationen chronologisch gesammelt, welche im Verlauf des Projekts eine Rolle spielen. Besprechungsprotokolle mit Entscheiden und Abmachungen sind besonders wichtig. Das Projektjournal wird laufend ausgefüllt.

<b>Datum:</b>	05.05.2014
<b>Typ:</b>	Meilenstein
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Product Manager teamraum® Mathias Leimgruber</li> </ul>
<b>Themen:</b>	Start der IPA
<b>Ergebnis / Entscheide:</b>	IPA wurde nach dem Kickoff-Meeting gestartet.

<b>Datum:</b>	05.05.2014
<b>Typ:</b>	Telefonbesprechung
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Hauptexperte Manuel J. Schaffner</li> <li>• Fachvorgesetzter Pascal Habegger</li> </ul>
<b>Themen:</b>	Telefonische Besprechung betreffend Meetings (Expertenbesuch 1, Expertenbesuch 2, Präsentation IPA).
<b>Ergebnis / Entscheide:</b>	Meetings festgelegt auf: 07.05.2014 / 22.05.2014 und 10.06.2014

<b>Datum:</b>	05.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>- Projektleiter Michel Weingart</li> <li>- Product Manager teamraum® Mathias Leimgruber</li> </ul>
<b>Themen:</b>	Analyse / Auswahl für Konzepterarbeitung
<b>Ergebnis / Entscheide:</b>	<p>Analyse für die verschiedenen Konzeptthemen. Die Entscheidung ist auf folgende Konzeptthemen gefallen:</p> <ul style="list-style-type: none"> <li>• Systemanforderung</li> <li>• Systemarchitektur</li> <li>• Zugriffskonzept</li> <li>• Betriebskonzept</li> <li>• Testkonzept</li> <li>• Einführungskonzept</li> </ul>

<b>Datum:</b>	06.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Product Manager teamraum® Mathias Leimgruber</li> </ul>
<b>Themen:</b>	Analyse / Auswahl Kriterien für Variantenentscheid
<b>Ergebnis / Entscheide:</b>	<p>Eine Auswahl von verschiedenen Kriterien für den Variantenentscheid wurde getroffen. Bei folgenden Kriterien macht es Sinn, diese für den Variantenentscheid einzusetzen:</p> <ul style="list-style-type: none"> <li>• Usability für Entwickler</li> <li>• Usability für weitere Benutzer</li> <li>• Flexibilität</li> <li>• Unabhängigkeit</li> <li>• Aufwand für: <ul style="list-style-type: none"> <li>- Entwicklung</li> <li>- Einführung</li> <li>- Testing</li> </ul> </li> </ul>

<b>Datum:</b>	06.05.2014 / 07.05.2014
<b>Typ:</b>	Mailing
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Hauptexperte Manuel J. Schaffner</li> </ul>
<b>Themen:</b>	Fragen betreffend Anpassung Zeitplan
<b>Ergebnis / Entscheide:</b>	Rückmeldung erhalten, inwiefern der Zeitplan angepasst werden darf. Soll-Zeiten werden belassen, hingegen bei den Ist-Zeiten kann es Verschiebungen geben.

<b>Datum:</b>	07.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Interner Projektleiter Murat Tokmak</li> </ul>
<b>Themen:</b>	Fragen zum Betriebskonzept
<b>Ergebnis / Entscheide:</b>	Der Aufbau des Betriebskonzepts wurde erläutert und ist klar.

<b>Datum:</b>	07.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Fachvorgesetzter Pascal Habegger</li> </ul>
<b>Themen:</b>	Fragen betreffend Dateistruktur und Berechtigungen
<b>Ergebnis / Entscheide:</b>	Die vorgeschlagene Dateistruktur muss noch überarbeitet werden. Die Struktur der Berechtigung wurde angenommen.

<b>Datum:</b>	07.05.2014
<b>Typ:</b>	Meeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Hauptexperte Manuel J. Schaffner</li> <li>• Fachvorgesetzter Pascal Habegger</li> </ul>
<b>Thema:</b>	<ul style="list-style-type: none"> <li>• Vorstellungsrunde</li> <li>• Aufgabenstellung besprechen</li> <li>• Ablauf der IPA besprechen</li> <li>• Besprechen der Dokumentation</li> <li>• Fragen beantworten:</li> </ul>
<b>Ergebnis / Entscheide:</b>	<p><b>Redundanzen</b></p> <ul style="list-style-type: none"> <li>• Redundanzen sollten in der Dokumentation vermieden werden. Stattdessen soll mit Verweisen gearbeitet werden.</li> </ul> <p><b>Roter Faden</b></p> <ul style="list-style-type: none"> <li>• Ein roter Faden soll sich durch die Dokumentation ziehen.</li> </ul> <p><b>Zeitplan</b></p> <ul style="list-style-type: none"> <li>• Die Programmierung zur Realisierung muss aufgeteilt werden.</li> <li>• Ist / Soll muss bei den Phasen klarer dargestellt sein.</li> <li>• Unterbruch im Zeitplan soll besser ersichtlich sein.</li> <li>• Meilensteine müssen im Zeitplan angegeben werden.</li> </ul> <p><b>Allgemeine Anpassungen / Punkte</b></p> <ul style="list-style-type: none"> <li>• Die Ich-Form wird ausschliesslich im Arbeitsjournal verwendet</li> <li>• Die Versionskontrolle soll kürzer ausgelegt sein.</li> <li>• Der Projektmanagementplan muss angepasst werden.</li> <li>• Bei den Projektrollen müssen die <b>Rollen</b> beschrieben werden.</li> <li>• Meilensteine sollen in die Umgebung des Zeitplans verschoben werden</li> <li>• Beteiligte Personen sollen zu den Personen verschoben werden</li> <li>• HERMES5 Grafik gehört zu den Mitteln und Methoden</li> <li>• Beschriftung beim Ist / Soll –Vergleiche muss klarer abgegrenzt sein</li> <li>• Systemziele sollen nummeriert sein.</li> <li>• Für die Wirtschaftlichkeit ist eine Grafik wünschenswert (Stunden Kosten...)</li> <li>• Eine Situationsanalyse soll im Konzept nicht vorhanden sein</li> <li>• Für die Systemarchitektur wird eine Grafik gewünscht</li> <li>• Plone soll erklärt werden.</li> <li>• Bei Hilfestellungen soll die Person namentlich erwähnt werden.</li> <li>• Ein stellvertretender Fachvorgesetzter ist nicht nötig.</li> <li>• Im Anhang wird der Code aufgeführt, welcher selber erarbeitet wurde.</li> <li>• Richtlinien betreffend Coding sind bei 4teamwork vorhanden (Pep8).</li> </ul>

	<ul style="list-style-type: none"> <li>• Implementationsschritte des Skripts auf den Server sind erwünscht.</li> <li>• Testprotokoll und Testplan erarbeiten für das Testing.</li> <li>• Bei der Realisierung kann auch mit Screenshots gearbeitet werden um etwas darzustellen.</li> <li>• Eventuell einen Link in der Dokumentation mitgeben auf eine existierende Umgebung, damit sich die Experten ein Bild davon machen können.</li> </ul>
--	---

<b>Datum:</b>	08.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Fachvorgesetzter Pascal Habegger</li> </ul>
<b>Themen:</b>	Diskussion betreffend Freigabekonzept
<b>Ergebnis / Entscheide:</b>	Das Einführungskonzept wird nicht mehr in demselben Masse benötigt wie vorgesehen, da das Skript erst nach Fertigstellung des Projekts auf einer produktiven Umgebung eingeführt wird.

<b>Datum:</b>	20.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Entwickler Jonas Baumann</li> </ul>
<b>Themen:</b>	OptionParser
<b>Ergebnis / Entscheide:</b>	OptionParser soll verwendet werden, um dem Skript „demo-teamraum.py“ Argumente und Optionen mitzugeben.

<b>Datum:</b>	21.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Product Manager teamraum® Mathias Leimgruber</li> </ul>
<b>Themen:</b>	<p>Nach einem git push auf dem Repository eGov wird folgende Fehlermeldung ausgegeben:</p> <pre>fatal: The current branch mwe-ipa_profile_demo-content has no upstream branch. To push the current branch and set the remote as upstream, use      git push --set-upstream origin mwe-ipa_profile_demo-content</pre> <p><a href="#">Abbildung 8: Fehlermeldung nach git push</a></p> <p>Da die Bedeutung der Ausgabe nicht ganz klar war, wurde nachgefragt.</p>
<b>Ergebnis / Entscheide:</b>	Der Branch wurde noch nicht mit dem Master verknüpft. Über den Befehl „git push -u origin mwe-ipa_profile_demo-content“ wird diese Verknüpfung hinzugefügt. Wäre der vorgeschlagene Befehl der Fehlermeldung befolgt worden, wäre der branch „mwe-ipa_profile_demo-content“ als neuer Master definiert worden.
<b>Datum:</b>	22.05.2014
<b>Typ:</b>	Meeting

<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Hauptexperte Manuel J. Schaffner</li> <li>• Fachvorgesetzter Pascal Habegger</li> </ul>
<b>Themen:</b>	<ul style="list-style-type: none"> <li>• Stand der Dokumentation</li> <li>• Allgemeine Verbesserungen in der Dokumentation</li> <li>• Verbesserungen Skript</li> <li>• Besprechung Präsentation</li> <li>• Besprechung der kommenden Termine</li> </ul>
<b>Ergebnis / Entscheide:</b>	<p><b>Zeitplan</b></p> <ul style="list-style-type: none"> <li>• Phase Einführung findet nicht statt, Punkte werden in Phase Realisierung verschoben</li> </ul> <p><b>Dokumentation</b></p> <ul style="list-style-type: none"> <li>• Grafik für Plone -&gt; erläutern, wie dies mit den anderen Systemen zusammenhängt</li> <li>• Abbildungen alle beschriften</li> <li>• Eingefügte Bilder besser beschreiben (was ist das Ticketingsystem...)</li> <li>• Für Tabellen soll eine andere Kopffarbe genommen werden</li> <li>• Was ist ein Profil? -&gt; Erklären</li> <li>• Einleitungssatz bei der Auflistung der Konfiguration</li> <li>• Link auf Plattform oder hilfreiche Dokumente (z.B. für weitere Informationen siehe...)</li> <li>• Teil 2 der Dokumentation: Die Beschreibung muss für aussenstehende Personen verständlich sein. Vom groben Detail ins Feine.</li> </ul> <p><b>Testprotokolle</b></p> <ul style="list-style-type: none"> <li>• Effektives Ergebnis gehört in das Testprotokoll</li> <li>• Testnummern ersetzen mit aussagekräftigem Stichwort</li> <li>• Mit Verweisen arbeiten</li> </ul> <p><b>Skripts</b></p> <ul style="list-style-type: none"> <li>• Warum JSON?</li> <li>• Wurden schon andere Profile definiert?</li> <li>• Runskript umbenennen und sinnvollen Name vergeben</li> </ul> <p><b>Präsentation</b></p> <ul style="list-style-type: none"> <li>• Teil 1 dauert 15 – 20 Minuten und ist in Deutsch abzuhalten. Schwerpunkte setzen auf spannendsten Themen mit den Präsentationsmittel welche benutzen werden.</li> <li>• Teil 2 dauert 10 Minuten und beinhaltet eine Demo</li> <li>• Teil 3 ist das Fachgespräch</li> </ul> <p><b>Websummary</b></p> <ul style="list-style-type: none"> <li>• Das Websummary ist 24 Stunden vor der Präsentation auf PkOrg hochzuladen</li> </ul>



	<ul style="list-style-type: none"> <li>• Das Websummary muss ausgedruckt bei der Präsentation vorliegen.</li> </ul>
--	---

<b>Datum:</b>	26.05.2014
<b>Typ:</b>	Kurzmeeting
<b>Beteiligte Personen:</b>	<ul style="list-style-type: none"> <li>• Projektleiter Michel Weingart</li> <li>• Entwickler Jonas Baumann</li> </ul>
<b>Themen:</b>	Verbesserungsmöglichkeiten, um das "Skript demo-teamraum.py" immer mit einem funktionierendem Python-Interpreter anzustossen, falls keine Option mitgegeben wird.
<b>Ergebnis / Entscheide:</b>	Es kann nicht garantiert werden, dass ein funktionierender Python-Interpreter vorhanden ist, weshalb die ausführende Person die benötigten Grundlagen schaffen muss, um das Skript ohne Konflikte auszuführen können. Das Skript kann nicht jede Eventualität abdecken.

## 11. Abschlussbericht

Im Abschlussbericht wird erläutert, inwiefern die geforderten Ziele des Projekts erreicht wurden. Dazu wird zuerst aufgezeigt, welche Kriterien erfüllt werden konnten und welche nicht. Anschliessend werden ein Fazit und ein persönliches Fazit zum Verlauf des Projekts erfasst.

### 11.1 Vergleich Ist/Soll

In der untenstehenden Tabelle sind die funktionalen und nicht funktionalen Anforderungen betreffend das Projekt aufgelistet und bewertet.

Kriterien	Aktueller Status	Erfüllt
Plattform soll automatisiert aufgesetzt sein.	Die Plattform ist nach Ausführen des Skripts „demo_teamraum.py“ automatisch aufgesetzt.	Ja
Inhalt soll automatisch erfasst sein.	Inhalt ist nach Ausführen des Skripts „demo_teamraum.py“ vorhanden.	Ja
Benutzer sollen automatisch erfasst sein.	Die drei definierten Benutzer sind nach Ausführen des Skripts „demo_teamraum.py“ vorhanden.	Ja
Aufwand für Entwickler wird vermindert.	Für das Aufsetzen der Demo-Plattform muss nur eine Konsoleneingabe durchgeführt werden, was den Aufwand für die Entwickler deutlich vermindert.	Ja
Die Lösung kann ohne grossen Aufwand erweitert / auf einen Server portiert werden.	Das Skripts „demo_teamraum.py“ kann ohne grossen Aufwand erweitert / auf einen Server portiert werden. Dies wird durch den einfachen Aufbau, den Einsatz von PEP8 und die umfangreichen Kommentare sichergestellt.	Ja
Die einfache Bedienung des Skripts „demo_teamraum.py“ ist gewährleistet.	Die Benutzeranleitung garantiert das erfolgreiche Ausführen des Skripts „demo_teamraum.py“.	Ja
Das Skript muss gewisse Sicherheitsmechanismen besitzen, um die unabsichtliche Überschreibung zu verunmöglichen.	Die Sicherheitsmechanismen sind in das Skript Skripts „demo_teamraum.py“ implementiert worden.	Ja
Fehler sollen dokumentiert werden.	Alle erkannten Fehler werden zur Laufzeit im Terminal ausgegeben. Gegenwärtig wird noch keine Logdatei erstellt.	Teilweise
Qualitätsgewinn der Produkte.	Der Qualitätsgewinn der Produkte wird sich nach einer gewissen Zeit bemerkbar machen.	Noch nicht
Supportaufwand wird vermindert.	Der Supportaufwand wird vermindert, indem weniger redundante Tickets von Kunden erstellt werden.	Noch nicht

Tabelle 8: Ist/Soll

### 11.2 Realisierung

Für das Erstellen des Skripts „apply\_profile.py“ und der automatisch erzeugten Benutzer wurde mehr Hilfe beansprucht als dies geplant war, da für die Erstellung dieses Skripts tiefgehende Kenntnisse des Frameworks von Plone erforderlich sind.

## **11.3 Testbericht**

Die Tests verliefen grundsätzlich erfolgreich und haben dabei geholfen, die Skripts weiter zu verbessern.

## **11.4 Fazit zum Projekt**

Das Projekt ist gut verlaufen. Die geplanten Zeitfenster konnten mehrheitlich eingehalten werden, und die Arbeiten wurden korrekt abgeschlossen. Die Ziele wurden erreicht. Trotz der genauen Planung hat das Projekt eine Überschreitung der Gesamtzeit von zwei Stunden erfahren, was aber bei einer Vorgabe von 80 Stunden nicht sonderlich ins Gewicht gefallen ist. Leider konnte aus Zeitmangel kein Logfile implementiert werden, welches die Aktionen aufzeichnen würde.

## **11.5 Persönliches Fazit**

Die geforderten Kriterien konnte ich erfüllen und das Projekt rechtzeitig abschliessen. Die Automation zum Aufsetzen von Demo-Plattformen, welche mit den erarbeiteten Skripts erbracht werden konnte, kann und wird in einem Produktivsystem bei 4teamwork zum Einsatz kommen. Das Skript wird die Arbeit der Entwickler darin erleichtern, die Demo-Plattform mit dem neusten Release bereitzustellen.

Mit dem Projekt konnte ich in den Bereichen der Programmierung und der internen Abläufe viele neue Erfahrungen sammeln. Zudem konnte ich mein Wissen zu Plone und Projektmanagement erweitern.

Die Durchführung des Projekts als Ganzes hat mir persönlich zugesagt, und es freut mich, dass mein Produkt in der Firma 4teamwork AG zum Einsatz kommen wird.

Die grösste Schwierigkeit bereitete mir der Start in die IPA. Am Anfang hatte ich noch nichts ausser der Dokumentvorlage und sah mein Ziel in weiter Ferne. Doch schon zwei Tage später konnte ich ein Gefühl für mein Projekt entwickeln. Eine weitere Schwierigkeit bestand darin zu verstehen, wo die benötigten Repositories (siehe Kapitel 14.1.1) einzuordnen sind und wie diese als Teil eines Ganzen, funktionieren. Auch habe ich zu viele Konzepte und Kapitel erstellt, welche ich eigentlich nicht benötigte und nach mehrmaligem Überarbeiten der Dokumentation wieder entfernte oder zusammenfasste.

Für künftige Projekte werde ich auf jeden Fall mehr Zeit für die Überarbeitung der Dokumentation einplanen. Es ist wichtig, dem Kunden letztendlich eine gute Projektdokumentation übergeben zu können.

## 12. Unterschriften Teil 1

Die lernende Person bestätigt mit ihrer Unterschrift diese IPA aus Eigenleistung erbracht und nach den Vorgaben der Prüfungskommission Informatik Kanton Bern erstellt zu haben. Die Angaben im Arbeitsjournal entsprechen dem geleisteten Arbeitsaufwand.

Datum	Name / OE	Unterschrift
27.05.2014	Michel Weingart	
27.05.2014	Pascal Habegger	

## Teil 2: Projektdokumentation

Auftraggeber: 4teamwork AG

IPA Projektname: **Automatische Installation von Demo-Webplattformen**

Autor: Michel Weingart

## 13. Initialisierung

Die Initialisierung ist ein Klärungsprozess, der mit vertretbarem Aufwand eine Entscheidung über die grundsätzliche Art der Systemrealisierung herbeiführt.

In dieser Phase wird beurteilt, ob das Projekt überhaupt realisiert werden kann.

### 13.1 Situationsanalyse

Um ein Grundverständnis für dieses Projekt zu erhalten, wird zuerst aufgezeigt, welche Applikationen und Entwicklungstools bei 4teamwork zum Einsatz kommen.

#### 13.1.1 Plone

Plone ist im eigentlichen Sinn ein Content Management System, welches vorwiegend auf den Einsatz in Organisationen und Unternehmen ausgerichtet ist. Folgende Eigenschaften zeichnen dieses CMS u.a. aus:

- Basiert auf Zope, einem Web-Framework (siehe auch [www.zope.org](http://www.zope.org))
- Ist quelloffen (Open-Source)
- Wurde in der objektorientierten Programmiersprache Python entwickelt
- Unterstützt rund 50 Sprachen
- Ist in einer weltweiten und professionellen Community integriert

Das CMS Plone wird seit der Firmengründung von 4teamwork zur Bereitstellung von Webplattformen verwendet. Es hat sich als verlässliches und dynamisches System erwiesen.

#### 13.1.2 teamraum®

Die modulare teamraum® Produktlinie wird von zahlreichen Kunden als Alternative zu SharePoint eingesetzt. Die Lösungen werden mit HTML5 Technologie entwickelt. Als Basis des Produkts teamraum® dient Plone als Content Management System. Für weitere Informationen zu teamraum® siehe <http://www.4teamwork.ch/loesungen/teamraum>.

Die zwei Applikationen stehen folgendermassen in Zusammenhang zueinander.

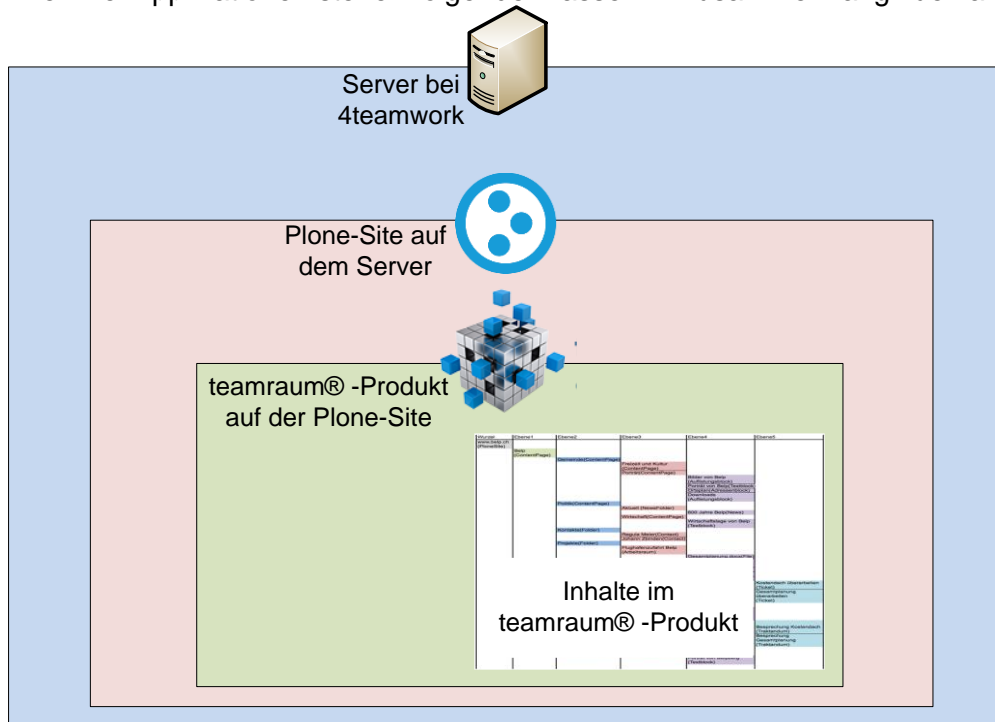


Abbildung 9: Zusammenhang der Systeme

### 13.1.3 Github / GitLab



Github und GitLab sind webbasierte Plattformen und werden bei 4teamwork verwendet, um Entwicklungsprojekte ablegen und zu versionieren. Auch wird ein Review-Prozess durchgeführt, um Fehler ausfindig zu machen. Beide Systeme können wahlweise über ein Terminal oder über einen Browser bedient werden.

Die wichtigsten Befehle in Bezug auf das Projekt sind in der nachfolgenden Tabelle eingetragen.

Terminalbefehl	Auswirkung
git clone <a href="https://git.4teamwork.ch/egov/buildout-base.git">git@git.4teamwork.ch:egov/buildout-base.git</a> demo-teamraum	Der Klon eines Git Repository wird lokal in den Ordner „demo-teamraum“ geklont / Das Repository ist somit ausgecheckt.
git checkout -b ipa-demo-teamraum	Der Branch (Zweig) ipa-demo wird innerhalb des heruntergeladenen Repository erstellt. Automatisch wird nach diesem Befehl auf den Branch gewechselt. Der Master des Repository wird somit verlassen um auf einem Zweig des Masters weiter zu arbeiten.
git add .	Alle Änderungen, welche innerhalb des heruntergeladenen Repository vorgenommen wurden, sind dem Branch hinzugefügt worden.
git commit	Der Änderung wird eine Beschreibung hinzugefügt.
git push origin -u ipa-demo-teamraum	Die Änderung wird auf das Git Repository hochgeladen.

Tabelle 9: Terminalbefehle git

Die folgende grafische Darstellung dient zur Erläuterung der Anwendungsfälle von Github und GitLab für 4teamwork.

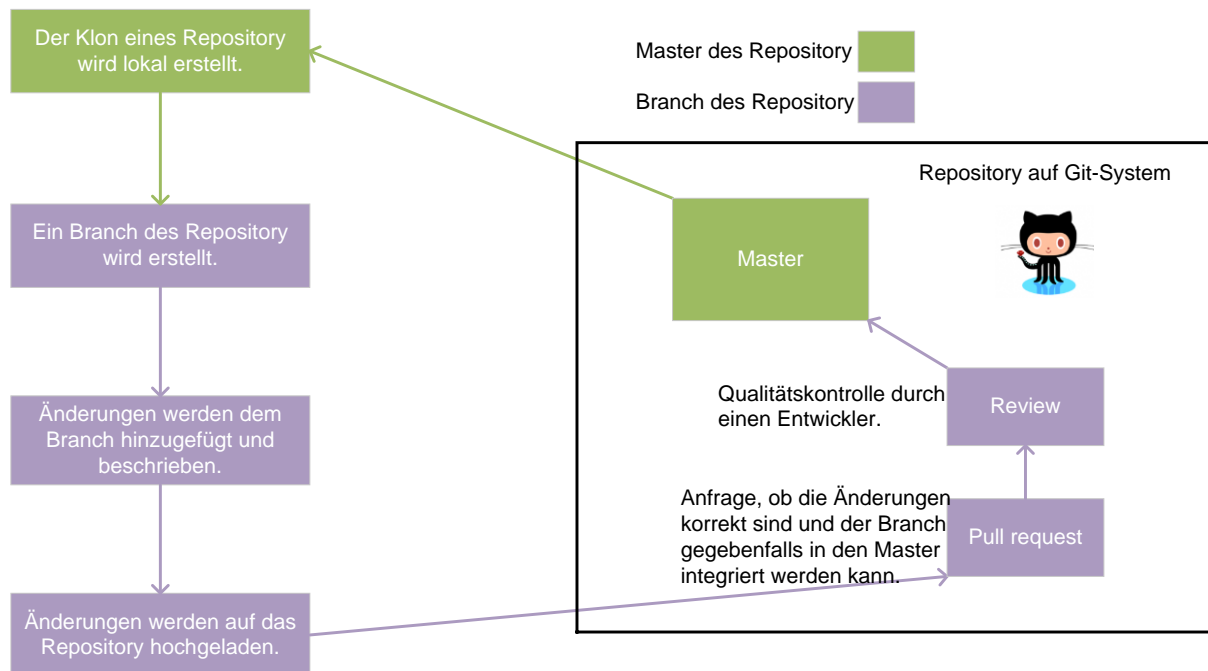


Abbildung 10: Erläuterung Git

## 13.2 Studie Ist-Zustand / Soll-Zustand

### Ist:

teamraum® Collab, teamraum® Intranet und teamraum® Web sind die webbasierten Standardlösungen, welche durch 4teamwork entwickelt wurden und immer mit neuen Funktionen erweitert und verbessert werden. Um diesen Prozess optimal umsetzen zu können, sind die neuen Funktionen in einen geordneten Releaseprozess eingebettet.

Der Releaseprozess kann in folgende Einzelschritte unterteilt werden:

- Gemeldete Probleme werden in Tickets bei 4teamwork erfasst
- Pro Ticket wird ein Paketrelease erstellt
- Ein neuer Release wird aus den Paketreleases erstellt
- Der Release wird eingespielt

Pro Jahr werden viele kleinere Paketreleases und 4-6 Releases für die Produkte von 4teamwork eingespielt. Die nachfolgende Darstellung zeigt auf, wie bei 4teamwork ein Release überhaupt entsteht.





**Abbildung 11: Releaseprozess Ist**

Ein neuer Release wird vom zuständigen Entwickler zwar lokal getestet, doch ist dieser für externe Personen nicht zugänglich. Diese können aber Tests auf der Demo-Plattform von 4teamwork durchführen und die gefundenen Fehler oder Mängel melden.

Diese Plattform wird weder regelmässig neu aufgesetzt, da der Aufwand doch relativ zeitraubend ist, noch sind Beispieldaten vordefiniert. Dies kann dazu führen, dass ein Release freigegeben wird, ohne dass dieser genügend getestet wurde. Mehr Informationen zu den Releases und was diese beinhalten siehe <http://www.teamraum.ch/release>.

### **Soll:**

Das Ziel des Projekts ist die automatische Bereitstellung der Demo-Plattform auf welcher neue Releases getestet werden können und automatisch Benutzer und Inhalte erfasst sind. Eine einfache Implementation der Lösung auf den Servern von 4teamwork wäre zudem wünschenswert.

Um eine realitätsnahe Umgebung für den Kunden bereitzustellen, wird die Demo-Plattform teilweise nach Abbild der Webseite von der Gemeinde Belp (siehe <http://www.belp.ch/>) nachgebaut.

Die nachfolgende Abbildung zeigt auf, wie nach Ablauf des Projekts ein Release bei 4teamwork entstehen soll.

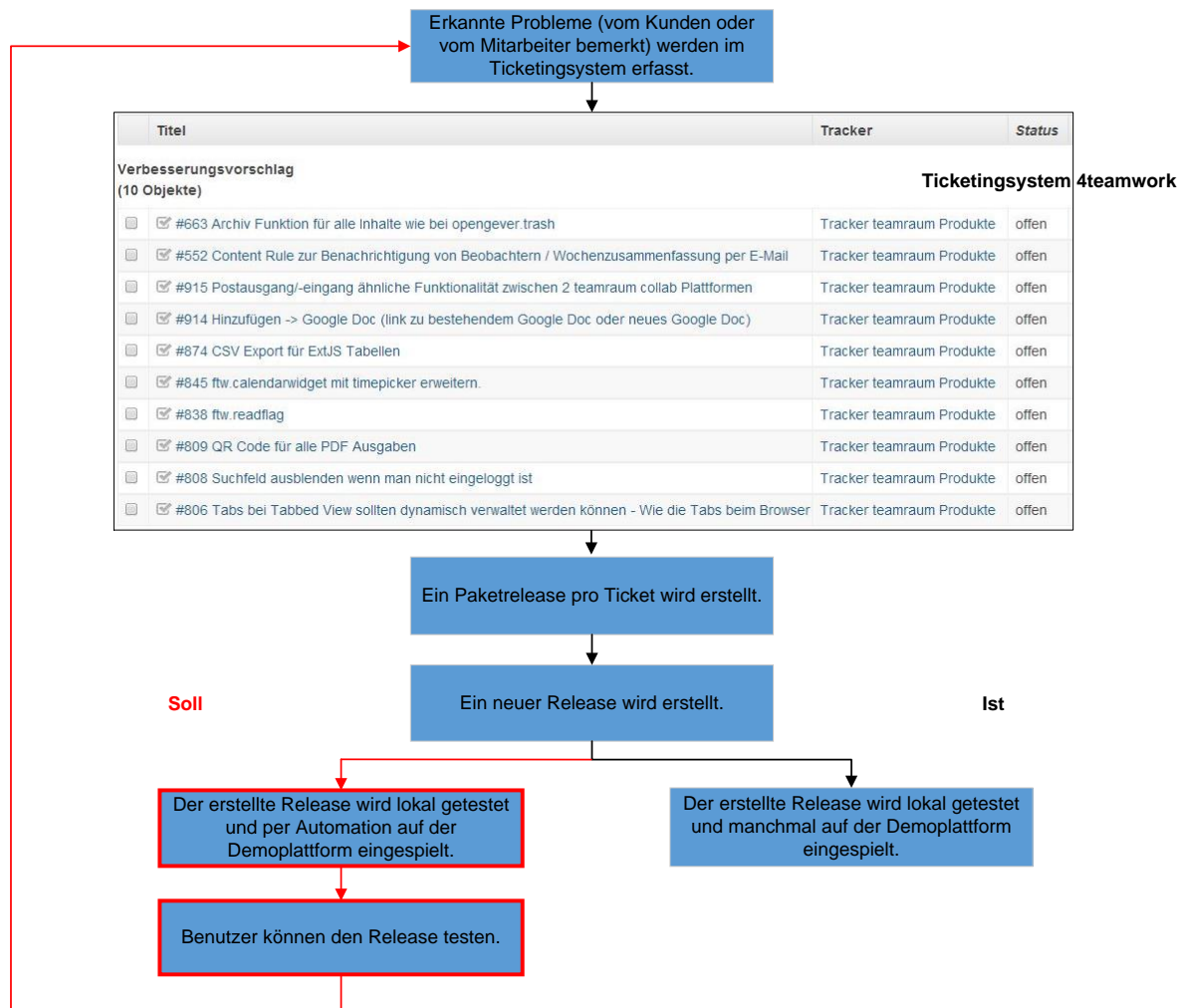


Abbildung 12: Releaseprozess Soll

### 13.3 Zeitrahmen / Budget

Der Zeitrahmen umfasst 10 Arbeitstage, das Projekt wird am 27.05.2014 abgeschlossen. Das Budget beläuft sich auf 80 Mitarbeiterstunden.

### 13.4 Übergeordnete Ziele

Fehler in den Releases sollen besser erkannt werden. Somit sollen die Qualität der Releases und auch die Qualität der Produkte von 4teamwork verbessert werden.

### 13.5 Chancen

Benutzer erwarten eine konsistente und fehlerfreie Applikation. Die Durchführung des Projekts ermöglicht eine weitere Annäherung an diese Erwartungshaltung. Neue Releases werden dank „Human Testing“ verbessert und Fehler erkannt und ausgemerzt.

### 13.6 Anforderungen

In der untenstehenden Tabelle sind die wichtigsten funktionalen Anforderungen an das zu erarbeitende Produkt erläutert.

Nr.	Anforderungen	Erreicht bis
1	Plattform soll automatisiert aufgesetzt werden.	26.05.2014
2	Inhalt soll automatisch erfasst werden.	21.05.2014
3	Benutzer sollen automatisch erfasst werden.	21.05.2014
4	Qualitätsgewinn der Produkte.	Nach Projektabschluss
5	Aufwand für Entwickler wird vermindert.	Nach Projektabschluss
6	Supportaufwand wird vermindert.	Nach Projektabschluss

**Tabelle 10: Systemziele**

Zusätzlich zu den funktionalen Anforderungen können folgende nicht funktionalen Anforderungen an das Projekt gestellt werden.

Nr.	Nicht funktionale Anforderungen
1	Eine einfache Bedienung der Lösung ist gewährleistet
2	Die Lösung kann ohne grossen Aufwand erweitert werden.
3	Die Lösung kann ohne grossen Aufwand portiert werden.

**Tabelle 11: Nicht funktionale Anforderungen**

## 13.7 Anwendungsbereich

Die Lösung wird einerseits von Entwicklern von 4teamwork genutzt, um die Demo-Plattform mit dem neusten Release aufzusetzen, andererseits von externen Benutzern, welche sich auf der Demo-Plattform anmelden, um die neusten Features und somit die neusten Releases zu testen.

## 13.8 Risikoanalyse

Die Demo-Plattform kann jederzeit auch manuell durch die Entwickler mit den neusten Releases aufgesetzt werden. Deshalb besteht kein Risiko, falls dieses Projekt nicht durchgeführt werden kann.

## 13.9 Varianten

Es bestehen mehrere Möglichkeiten, die gewünschte Demo-Plattform bereitzustellen. Entweder wird diese auf einer virtuellen Maschine installiert und das Image bei jedem Release neu konfiguriert oder es wird ein Skript erarbeitet, welches ermöglicht, die Demo-Plattform ohne grossen Aufwand aufzusetzen. Anhand folgender Beurteilungskriterien wird eine der zwei Varianten ausgewählt:

- **Usability:** Wie bedienerfreundlich ist das Aufsetzen der Demo-Plattform für die Entwickler? Kann die Demo-Plattform auch von weiteren Personen benutzt werden?
- **Flexibilität:** Welcher Aufwand wird benötigt, um die entsprechende Variante zu erweitern oder anzupassen?
- **Unabhängigkeit:** Kann das Aufsetzen der Demo-Plattform unter verschiedenen Betriebssystemen gestartet werden?
- **Entwicklungsaufwand:** Welcher Aufwand wird benötigt, um die Variante zu entwickeln?

### 13.9.1 Variante 1: VMware-Image

Auf einer virtuellen Maschine wird ein vorbereiteter Release einer Webanwendung, inklusiv Beispielbenutzer und -daten, installiert. Diese virtuelle Maschine kann verteilt und zwecks Tests zur Verfügung gestellt werden. Das Image kann regelmässig mit einem neuen Release neu geladen werden. Folgende Vorteile und Nachteile konnten zu dieser Variante eruiert werden:

- **Usability für Entwickler:** Recht umständlich, da die Entwickler bei 4teamwork nicht mit Images arbeiten und dies ein Mehraufwand bedeuten würde.

+ **Usability für weitere Benutzer:** Ein externer Benutzer kann ein Image einbinden und die Plattform nutzen.

- **Flexibilität:** Ein Image ist recht starr und eine Erweiterung recht aufwändig. Um einen neuen Release zu erfassen, müsste das Image jedes Mal angepasst werden.

+ **Unabhängigkeit:** Ein Image kann auf verschiedensten Betriebssystemen aufgesetzt werden.

+ **Entwicklungsaufwand:** Der Aufwand, um eine virtuelle Maschine mit Unix-basiertem Betriebssystem aufzusetzen und ein Skript zur Automatisierung zu erstellen, ist relativ gering. Wobei beachtet werden sollte, dass innerhalb der VMware auch ein Linux installiert und gepflegt/aktualisiert werden muss.

### 13.9.2 Variante 2: Skriptbasierte Installation

Auf einer bestehenden Linux-Umgebung wird per Skript die Webanwendung automatisch installiert und Beispielbenutzer und -inhalte erfasst. Folgende Vorteile und Nachteile konnten zu dieser Variante eruiert werden:

+ **Usability Entwickler:** Die Ausführung des Skripts bedeutet für den Entwickler praktisch keinen Aufwand, da dies auf einer bekannten Umgebung ausgeführt wird.

- **Usability für weitere Benutzer:** Ein externer Benutzer kann das Skript ohne Vorkenntnisse wahrscheinlich nicht ausführen.

- **Flexibilität:** Ein Skript kann schnell angepasst und erweitert werden. Ein Entwickler kann in kürzester Zeit eine neue Version eintragen.

+ **Unabhängigkeit:** Ein Python-Skript kann zwar auf verschiedenen Betriebssystemen ausgeführt werden, doch wäre dieses Skript abhängig von benutzerdefinierten Einstellungen und nicht einfach ohne Weiteres lauffähig.

+ **Entwicklungsaufwand:** Der Aufwand, um ein Skript zu erstellen, ist gering.

## 13.10 Variantenentscheid

Anhand des nachfolgenden Variantenentscheids werden die Beurteilungskriterien zu den beiden Varianten gewertet und die bessere Variante ausgewählt.

Kriterien	Variante I Bew.	Variante II Bew.	Gewichtung
Usability für Entwickler	3	6	30% (3x)
Usability für weitere Benutzer	5	2	10% (1x)
Flexibilität	3	4	20% (x2)
Unabhängigkeit	5	2	10% (1x)
Aufwand	4	5	30% (3x)
<b>Total</b>	<b>37</b>	<b>45</b>	<b>100%</b>

Bewertung: Unbrauchbar **0**- bis ausgezeichnet **6**

Tabelle 12: Variantenentscheid

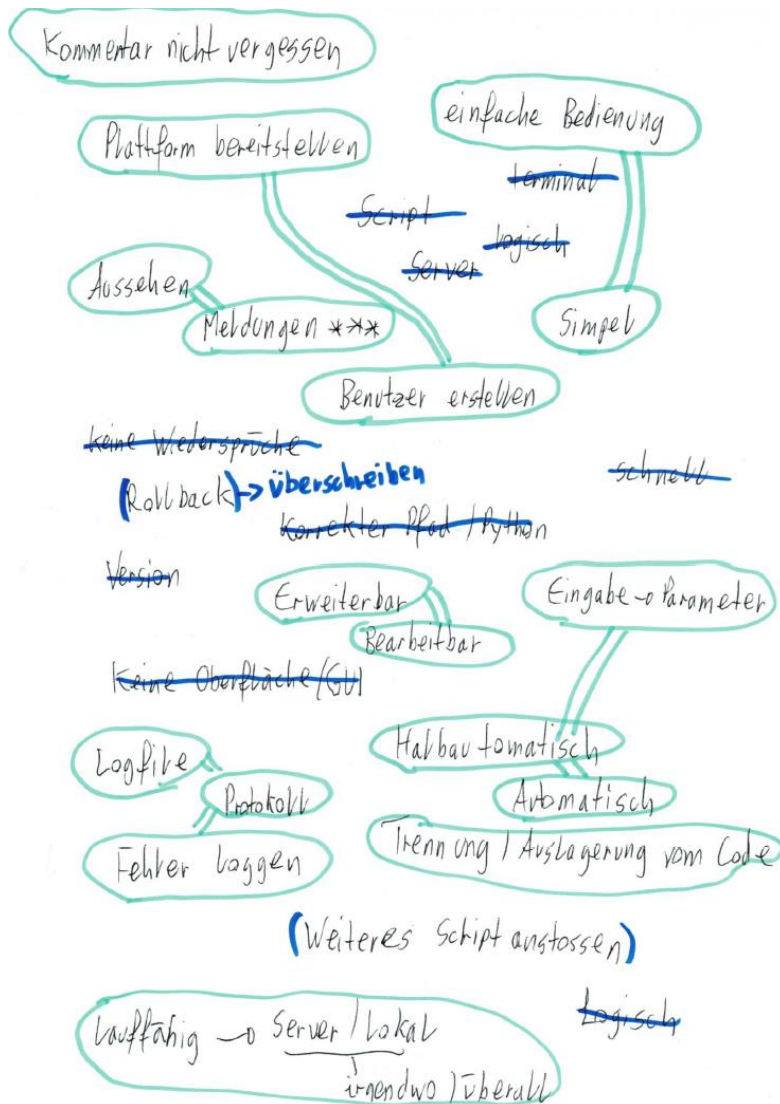
### 13.10.1 Begründung

Aufgrund des Variantenentscheids wurde die Umsetzung für Variante zwei entschieden. Ein sehr grosser Vorteil der Skript-Variante ist die Kombination der Flexibilität und Usability für die Entwickler. Das Skript könnte auch Eingabeparameter entgegennehmen, welche für die Variante mit virtueller Maschine nicht definiert werden könnten.

## 13.11 Lösungen suchen

Um Lösungsansätze zu finden, wie das Skript funktionieren / aussehen und aufgebaut sein soll, wurde die Kreativitätstechnik „Brainstorming“ gewählt. Während fünf Minuten wurden alle Ideen zu folgender Fragestellung gesammelt:

### Was muss das ideale Skript beinhalten und wie funktioniert es?



**Abbildung 13: Brainstorming**

Die wichtigsten Kriterien wurden dem Brainstorming extrahiert und werden teilweise für den Lösungsentscheid verwendet.

Das Skript kann somit folgende Elemente beinhalten:

- Automation soll gewährleistet sein, eventuell mit Parameterübergabe
- Die Benutzererstellung soll gewährleistet sein
- Die Inhaltserstellung soll gewährleistet sein
- Berechtigung, damit das Skript ausgeführt werden kann, soll gewährleistet sein
- Das Skript soll über das Terminal angestossen werden
- Kommentare sollen im Code enthalten sein
- Das Skript soll erweiterbar sein
- Eventuell soll ein Logfile / Protokoll implementiert werden
- Im Skript sollen eventuell weitere Skripts angestossen werden können
- Wichtige Punkte sollen im Skript mit Markierungen aufgezeigt werden

## 13.12 Freigabe Konzept

Die Phase Konzept wird freigegeben:

Datum	Name / Auftraggeber	Unterschrift
06.05.2014	Pascal Habegger	



## 14. Konzept

In dieser Phase steht die Erarbeitung der Grundlagen betreffend die Realisierung des Projekts im Vordergrund. In dieser Phase werden die Konzepte „Betriebskonzept“, „Systemkonzept“ und „Testkonzept“ behandelt, welche für die korrekte Realisierung des Projekts nötig sind.

### 14.1 Betriebskonzept

Im folgenden Abschnitt wird beschrieben wie das Skript, nachfolgend „**demo\_teamraum.py**“ genannt, effektiv betrieben und implementiert werden soll. Dazu werden zuerst die Zusammenhänge der benötigten Repositories aufgezeigt, das Skript „demo\_teamraum.py“ beschrieben, die Implementation der verschiedenen Skripts erläutert und anschliessend die Anforderung an die Lauffähigkeit zusammengefasst.

#### 14.1.1 Repositories

Der Aufbau der webbasierten Standardlösungen teamraum® ist sehr umfangreich und unterliegt vielen Abhängigkeiten. Um eine Plattform zu installieren, werden verschiedenste Repositories benötigt. Nachfolgend werden die wichtigsten Repositories kurz beschrieben, die für dieses Projekt benötigt werden.

##### eGov

Das Repository eGov ist der Kern der teamraum®- Produkte. Bei 4teamwork sind in diesem Repository u.a. die Profile für die verschiedenen webbasierten Standardlösungen (teamraum® Collab, teamraum® Intranet und teamraum® Web) definiert. Ein Profil beinhaltet die Daten, welche nach Aufsetzen der Demo-Plattform zur Verfügung stehen sollen. Für die Demo-Plattform wird ein neues Profil benötigt, welches diesem Repository hinzugefügt werden muss. Die Inhalte der Profile sind im Datenformat JSON erfasst. Als Vorlage für die Profildaten dient, wie in Kapitel 13.2 erwähnt, die Gemeinde Belp.

Das neue Profil lautet auf den Namen „**teamraum-demo**“.

Nachfolgend sind die Struktur und die Beispieldaten aufgezeigt, welche das Profil „teamraum-demo“ enthalten werden. In der Klammer steht der jeweilige Inhaltstyp geschrieben, welcher in einer Tabelle in Anhang 1, erläutert wird.



Wurzel	Ebene1	Ebene2	Ebene3
gemeinde (PloneSite)			
	Neues Aktuelles (NewsFolder)		
		Abwasserreglement (News)	Paragraph1 (TextBlock)
		Kulturtage (News)	
			Paragraph1 (TextBlock)
	Lebensqualität (ContentPage)	Jugend (ContentPage)	
			Prix Belp (TextBlock)
	Verwaltung (ContentPage)		
		Adresse (AddressBlock) Bilder (ListingBlock) Paragraph1 (TextBlock)	
			image1 (Image) image2 (Image) image3 (Image)
	Wirtschaft Gewerbe (ContentPage)	Zusammenarbeit (TextBlock)	
	Politik (ContentPage)	Kräfteverhältnis (TextBlock) Aktuell (NewsFolder)	
			600 Jahre Belp (News)
	Kontakte (Folder) Projekte (Folder)		
		Flughafenzufahrt Belp (Workspace)	
			Flughafen (Image) Evaluation.xlsx (File) IT.docx (File) Nachhaltigkeit.pdf (File)
	Belpberg(Subsite)	Porträt (ContentPage)	
			Bild (Image) Intro (Textblock)

Abbildung 14: Inhaltsstruktur

## Buildout / teamraum-buildout

Buildout ist ein Buildsystem, welches auf Python basiert und dazu dient, Webanwendungen automatisch zu installieren. Buildout sammelt verschiedene Pakete und installiert daraus die Webanwendung.

Bei 4teamwork sind im Repository teamraum-buildout das Buildsystem selbst und firmenspezifisch erweiterte Konfigurationseinstellungen enthalten.

Im Repository teamraum-buildout wird auch das Skript implementiert, welches in diesem Projekt dazu dient, das Profil teamraum-demo zur Installation auszuwählen.

Das erwähnte Skript lautet auf den Namen „**apply\_profile.py**“.

### 14.1.2 demo\_teamraum.py

Das Skript „demo\_teamraum.py“ kann als Hauptskript betrachtet werden. Es führt die benötigten Konsolenbefehle automatisiert durch, klonet die benötigten Repositories und stösst das Skript „apply\_profile.py“ an. Im Projekt wird auf das Skript „demo\_teamraum.py“ am

meisten eingegangen, da es, wie bereits erwähnt, die Hauptkomponente des automatisierten Prozesses darstellt.

### 14.1.3 Implementation in die bestehende Umgebung

Damit parallel unterschiedliche Versionen des Repositories „teamraum-buildout“ weiterentwickelt werden können, wird ein Branch mit Namen **mw-ipa\_runscript** erstellt, welchem ein neuer Ordner **scripts** hinzugefügt wird. Dieser enthält wiederum die beiden Skripts „demo\_teamraum.py“ und „apply\_profile.py“.

Im Repository „eGov“ wird ein Branch mit Namen **mw-ipa\_profile\_demo-content** erstellt. Auf diesem Branch werden die neuen Profildaten erstellt.

Um die Zusammenhänge der verschiedenen Skripts, Repositories und Branches verständlich zu machen, wird dies in der nachfolgenden Grafik nochmals zusammengefasst.

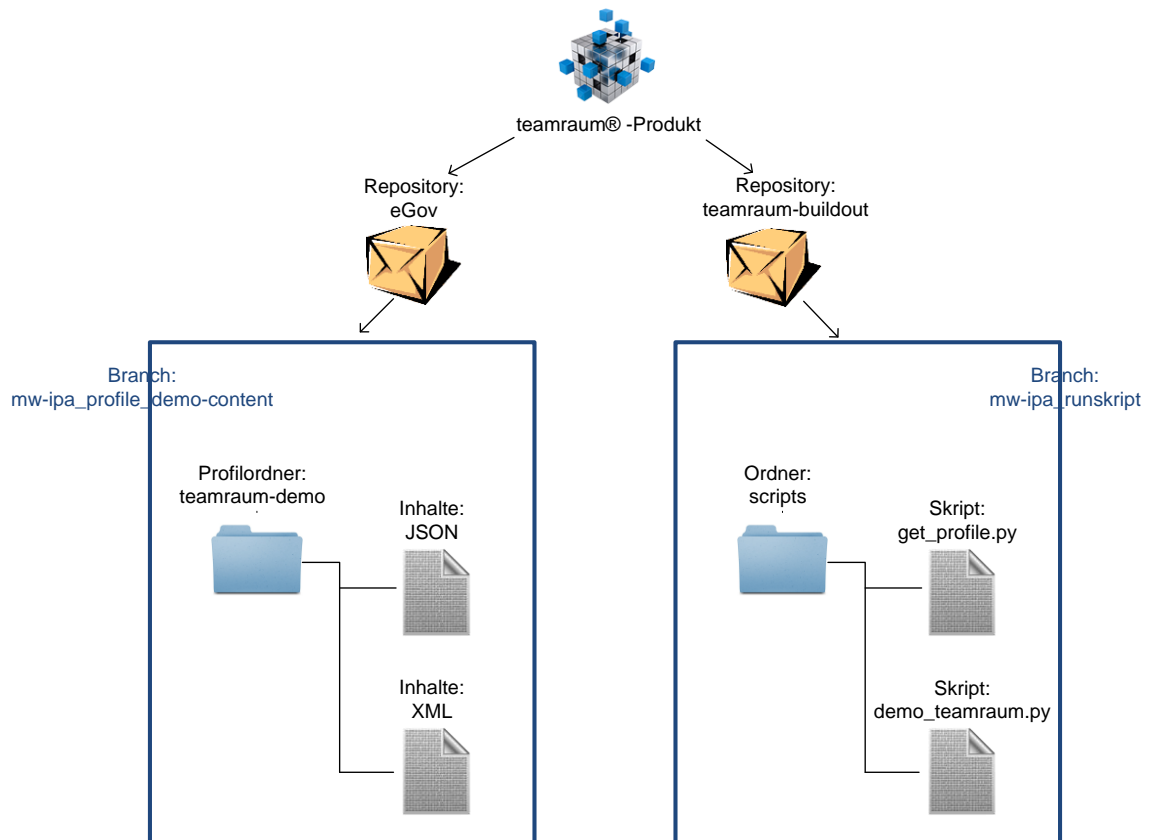


Abbildung 15: Zusammenhang Skripts, Repositories und Branches

### 14.1.4 Versionierung demo\_teamraum.py

Nach Projektabschluss wird die Version 1.0 des Skripts „demo\_teamraum.py“ in den Master des Repository „teamraum-buildout“ gemerged. Das bedeutet, dass bei einem erneuten Herunterladen des Repositories nicht auf den Branch „mw-ipa\_runscript“ gewechselt werden muss, um das Skript „demo\_teamraum.py“ zu erreichen.

Um eine neue Version des Skripts „demo\_teamraum.py“ zu erstellen, wird wiederum ein Branch des Repositories „teamraum-buildout“ erstellt und dieser nach Fertigstellung gemerged.

### 14.1.5 Betrieb auf dem Server

Die Ausführung des Skripts „demo\_teamraum.py“ auf dem Server wird ähnlich wie die Ausführung auf einem lokalen Rechner funktionieren. Einige kleinere Anpassungen sind nötig, um diese Funktionalität gewährleisten zu können. Die Implementation wird in einem Release 2.0 des Skripts „demo\_teamraum.py“ nach Abschluss des Projekts erstellt.

### 14.1.6 Anforderungen

Um die Lauffähigkeit des Skripts „demo\_teamraum.py“ gewährleisten zu können, müssen gewisse Anforderungen erfüllt sein. Diese wurden teilweise dem Brainstorming beim vorherigen Kapitel entnommen.

Nr.	Anforderungen	Erreicht bis
1	Python2.7.5 muss installiert sein, welches mit Plone umgehen kann.	19.05.2014
2	Dem Skript „demo_teamraum.py“ muss der Pfad zum korrekten Python-Interpreter bekannt sein.	20.05.2014
3	Das Skript „demo_teamraum.py“ muss die entsprechenden Berechtigungen besitzen, um eine Interaktion durchführen zu können.	22.05.2014
4	Das Skript „demo_teamraum.py“ muss gewisse Sicherheitsmechanismen besitzen, um z.B. keine Überschreibung bestehender Instanzen durchführen zu können.	22.05.2014
5	Das Skript „demo_teamraum.py“ muss sich auf das entsprechende Git Repository authentifizieren können.	22.05.2014

Tabelle 13: Anforderungen Skript „demo\_teamraum.py“

Zusätzlich zu den funktionalen Anforderungen können folgende nicht funktionalen Anforderungen an das Skript „demo\_teamraum.py“ gestellt werden.

Nr.	Nicht funktionale Anforderungen
1	Das Skript „demo_teamraum.py“ soll Eingabeparameter entgegennehmen können.
2	Das Skript „demo_teamraum.py“ soll Fehler dokumentieren.
3	Das Skript „demo_teamraum.py“ soll erweiterbar sein.

Tabelle 14: Nicht funktionale Anforderungen Skript „demo\_teamraum.py“

## 14.2 Zugriffskonzept

Bei der Erarbeitung des Zugriffskonzepts geht es darum, die grundsätzlichen Berechtigungen der verschiedenen automatisch erzeugten Benutzer auf der Demo-Plattform aufzuzeigen.

### 14.2.1 Berechtigungen

In der nachfolgenden Tabelle wird aufgeführt, welche Benutzer erstellt werden, welche Funktionen diese einnehmen und mit welchen Benutzerdaten sie sich einloggen werden.

Benutzer	Funktion	Loginname	Passwort
Test Administrator	Administriert die Inhalte der Plattform. Er kann die Struktur anpassen und Module hinzufügen.	test_administrator	Admin_4teamwork.ch
Test Author	Kann Inhalte erfassen / bearbeiten und löschen.	test_author	Author_4teamwork.ch
Test Gast	Hat Leserechte auf Inhalte.	test_gast	Gast_4teamwork.ch

Tabelle 15: Berechtigungstabelle

## 14.3 Testkonzept

Im Testkonzept werden die wichtigsten Schritte erläutert, welche für ein erfolgreiches Testen benötigt werden.

### 14.3.1 Ausgangslage

Der Projektabschluss ist für Ende Mai geplant. Bis zu diesem Zeitpunkt muss die Applikation für den Betrieb fit gemacht werden. Zu diesem Zweck werden die Testmethoden, die Testziele, ein Testrahmen, das Testvorgehen und ein Testkonzept erarbeitet. Es soll sichergestellt werden, dass nur eine stabile und den Qualitätsanforderungen entsprechende Applikation produktiv eingesetzt wird.

### 14.3.2 Testmethode

Für dieses Projekt wird ein Testvorgehen nach folgendem Phasenmodell erarbeitet.

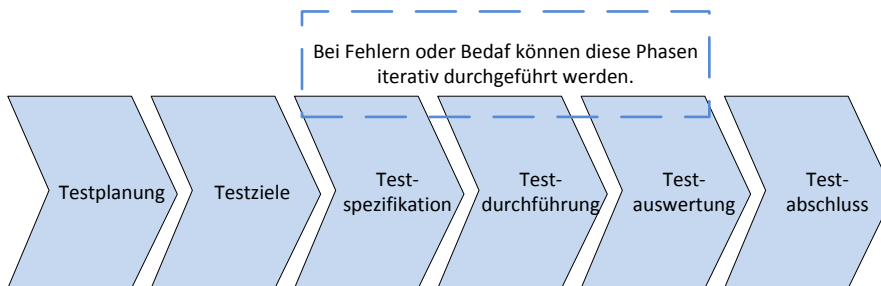


Abbildung 16: Phasenmodell

### 14.3.3 Testplanung

Der Testumfang beschränkt sich auf die Funktionalität auf einem MacBook Pro. Trotzdem sollen die Tests, auch für spätere Versionen und insbesondere für die Implementation auf einem Server, nicht an Gültigkeit verlieren. Werden die Tests nicht wie geplant abgearbeitet, besteht das Risiko, eine fehlerhafte Applikation in den Betriebsprozess zu implementieren, was mehr Schaden als Nutzen erbringen würde.

#### 14.3.4 Testziele

Die Hauptziele betreffend die Tests sind nachfolgend aufgelistet:

Nr.	Ziele
1	Fehler in den Skripts werden gefunden.
2	Fehlerhafte Inhalte auf der Plattform werden gefunden.
3	Fehlerhaftes Verhalten betreffend vordefinierten Benutzern werden gefunden.
4	Die Stabilität der Skripts ist ermittelt.

**Tabelle 16: Hauptziele Tests**

#### 14.3.5 Hilfsmittel

Als Hilfsmittel dienen der Testplan, das Testprotokoll und das Benutzeranleitung (Anhang 2). Es besteht kein Ausbildungsbedarf für die testende Person.

#### 14.3.6 Testvorbereitung

Das Skript „demo\_teamraum.py“ wird auf einem Testnotebook bereitgestellt. Die Benutzeranleitung, das Testprotokoll und der Testplan liegen bei. Bei Bedarf oder Fehler werden die Tests zusätzlich iterativ durchgeführt.

#### 14.3.7 Testdurchführung

Die Durchführung des Skripts „demo\_teamraum.py“ dauert ca. zwei Minuten und wird mit der Konsolenausgabe

Die Plattform wurde erfolgreich aufgesetzt.

abgeschlossen. Nach jedem Test muss zwingend der Instanzenordner gelöscht werden, damit das Resultat nicht verfälscht wird.

#### 14.3.8 Testfälle

Die nachfolgenden Testfälle werden in der Projektphase Realisierung abgearbeitet und das dazugehörige Testprotokoll ausgefüllt. Die Testnummern werden mit s für Skript und p für Plattform bezeichnet.

#### Spezifische Testfälle betreffend „demo\_teamraum.py“

<b>Testfall:</b> Nr. 1s
<b>Kurzbeschreibung:</b> Skript ausführen ohne Ordnername und ohne Pfad zu einem Python-Interpreter.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 22.05.2014
<b>Beschreibung:</b> Starten des Skripts per Terminalbefehl: „python demo_teamraum.py“
<b>Erwartetes Ergebnis:</b> Das Skript wird nicht durchgeführt, da kein Instanzenordner mitgegeben wurde. Eine Fehlermeldung wird geworfen

<b>Testfall:</b> Nr. 2s
<b>Kurzbeschreibung:</b> Skript ausführen mit Instanzenname und ohne Pfad zu einem Python-Interpreter.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 22.05.2014
<b>Beschreibung:</b> Starten des Skripts per Terminalbefehl: „python demo_teamraum.py teamraum-buildout“
<b>Erwartetes Ergebnis:</b> Das Skript wird höchstwahrscheinlich durchgeführt, ein Fehler betreffend Python kann aber erwartet werden.

<b>Testfall:</b> Nr. 3s
<b>Kurzbeschreibung:</b> Skript ausführen mit Instanzenname, mit Pfad zu einem Python-Interpreter und mit Instanzenordner.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 26.05.2014
<b>Beschreibung:</b> Starten des Skripts per Terminalbefehl: „python demo_teamraum.py -p 'Pfad zu einem Python-Interpreter' teamraum-buildout“
<b>Erwartetes Ergebnis:</b> Das Skript wird durchgeführt, falls ein korrekter Python-Interpreter als Option angegeben wird.

<b>Testfall:</b> Nr. 4s
<b>Kurzbeschreibung:</b> Zweimaliges Ausführen des Skripts mit demselben Ordernamen.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 26.05.2014
<b>Beschreibung:</b> 1. Starten des Skripts per Terminalbefehl „python demo_teamraum.py -p 'Pfad zu einem Python-Interpreter' teamraum-buildout“ 2. Unterbrechen des Skripts mit der Tastenkombination: „ctrl c“ Neustart des Skripts per Terminalbefehl: „python demo_teamraum.py -p 'Pfad zu einem Python-Interpreter' teamraum-buildout“
<b>Erwartetes Ergebnis:</b> Das Skript wird abgebrochen (beim zweiten Ausführen) mit der Fehlermeldung, dass der Instanzenordner schon existiert.

<b>Testfall:</b> Nr. 5s
<b>Kurzbeschreibung:</b> Skript mehrmals gleichzeitig ausführen.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 22.05.2014
<b>Beschreibung:</b> Über mehrere Terminals wird das Skripts mehrmals, per Terminalbefehl: „python demo_teamraum.py -p 'Pfad zu einem Python-Interpreter' teamraum-buildout“, gestartet
<b>erwartetes Ergebnis:</b> Das Skript, welches eher gestartet wird, wird erfolgreich ausgeführt. Das Skript, welches später gestartet wird, wird abgebrochen mit der Fehlermeldung, dass der Instanzenordner schon existiert.

### Spezifische Testfälle betreffend Plattform

<b>Testfall:</b> Nr. 6p
<b>Kurzbeschreibung:</b> Vorhandensein der Plattform.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 26.05.2014
<b>Beschreibung:</b> Ist die Plattform nach Durchlaufen des Skripts vorhanden? Verbinden über localhost:8080
<b>Erwartetes Ergebnis:</b> Die Plattform ist vorhanden.

<b>Testfall:</b> Nr. 7p
<b>Kurzbeschreibung:</b> Vorhandensein der Struktur.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 26.05.2014
<b>Beschreibung:</b> Ist die Struktur, wie in der Abbildung von Kapitel 14.1.1 beschrieben, vorhanden? Verbinden über localhost:8080 und die Plone-Seite anzeigen lassen.
<b>Erwartetes Ergebnis:</b> Die Struktur ist vorhanden.

<b>Testfall:</b> Nr. 8p
<b>Kurzbeschreibung:</b> Vorhandensein der Benutzer.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 22.05.2014
<b>Beschreibung:</b> Login mit Administrator. Sind folgende Benutzer unter <a href="http://localhost:8080/gemeinde/user_management">http://localhost:8080/gemeinde/user_management</a> vorhanden? <ul style="list-style-type: none"> <li>- Test Administrator</li> <li>- Test Author</li> <li>- Test Gast</li> </ul>
<b>Erwartetes Ergebnis:</b> Die Benutzer sind vorhanden.

<b>Testfall:</b> Nr. 9p
<b>Kurzbeschreibung:</b> Kontrolle der Berechtigungen.
<b>Tester:</b> Thomas Buchberger
<b>Datum:</b> 26.05.2014
<b>Beschreibung:</b> Funktionieren die Berechtigungen laut Zugriffskonzept 14.2.1 Hat der Benutzer „Anonym“ Leserechte auf alle Inhalte? Hat der Benutzer „Autor“ die Möglichkeit Inhalte anzupassen? Kann der Benutzer „Administrator“ Inhalte hinzufügen?
<b>Erwartetes Ergebnis:</b> Die Berechtigungen funktionieren gemäss Zugriffskonzept 14.2.

## 14.4 Einführungskonzept

Das Einführungskonzept beschreibt, welche Durchführungsschritte und Anforderungen erfüllt sein müssen, um das Skript „demo\_teamraum.py“ auf dem Server poseidon.4teamwork.ch zu installieren und somit die Demo-Plattform zur Verfügung zu stellen. Da das Skript nach Projektende für den Gebrauch auf dem Server noch angepasst werden muss, kann das Einführungskonzept in der Praxis noch nicht durchgeführt werden.

### 14.4.1 Anforderung

Um Python-Skripte ausführen zu können, wird ein Python-Interpreter auf dem Server verlangt. Diese Voraussetzung ist auf allen Servern von 4teamwork gegeben.



#### **14.4.2 Durchführung**

Um die Demo-Plattform aufsetzen zu können, werden lediglich das Repository „teamraum-buildout“ auf den Server geklont und das Skript „demo\_teamraum.py“ ausgeführt.

#### **14.4.3 Betrieb**

Um die Demo-Plattform öffentlich zugänglich zu machen, muss vom Systemtechniker noch eine RewriteRule auf dem Apache Server eingerichtet werden, damit Anfragen an den Webserver weitergeleitet werden.

## 14.5 Freigabe Realisierung

Die Phase Realisierung wird freigegeben:

Datum	Name / Auftraggeber	Unterschrift
19.05.2014	Pascal Habegger	

## 15. Realisierung

Die Phase Realisierung dient zur effektiven Erstellung des Systems. Das Erarbeiten der Skripts und des Profils werden nachfolgend aufgezeigt und erläutert. Anschliessen werden im Testprotokoll die erarbeiteten Testergebnisse festgehalten.

Am Ende der Phase Realisierung erfolgen die Problemanalyse und Problembehebung.

### 15.1 Legende

Terminalbefehle, Quellcode und Fehler werden in diesem Kapitel wie folgt dargestellt:

Terminaleingaben
Quellcode
Fehler

### 15.2 Bestehende Umgebung verifizieren

Die Realisierung wird lokal auf einem MacBook Pro in der folgenden Struktur durchgeführt. Die wichtigen Elemente sind markiert.

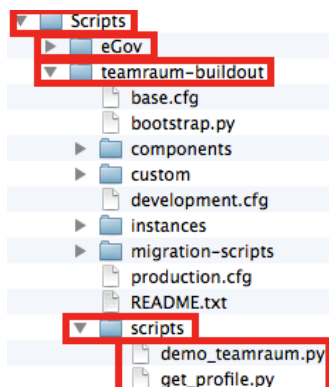


Abbildung 17: Struktur

Die Namensgebung wird von Kapitel 14.1 des Konzepts übernommen.

Die Anforderungen sind, wie in Kapitel 14.1.6 beschrieben, vorhanden.

Die Vollständigkeit der Repositories wird nach dem Download überprüft, um eine solide Basis für die Entwicklung der Skripts zu ermöglichen.

### 15.3 Konfiguration festlegen

Nachfolgend wird zusammengefasst, welche Konfigurationseinstellungen beim Erarbeiten der Skripts und des Profils eingehalten werden müssen.

- Für das Repository eGov wird ein Branch erstellt  
Der Branch erhält, wie in Kapitel 14.1.3 beschrieben, den Namen „mw-ipa\_profile\_demo-content“.
- Für das Repository teamraum-buildout wird ein Branch erstellt.  
Der Branch erhält, wie in Kapitel 14.1.3 beschrieben, den Namen „mw-ipa\_runscript“.

- Das Skript überschreibt eine existierende Plattform nie.  
Es muss zwingend eine Meldung ausgegeben werden, falls die Plattform existiert.
- Der Pfad zu einem gültigen Python-Interpreter sollte, muss aber nicht, angegeben werden.  
Falls kein Pfad übergeben wird, soll der der Python-Interpreter verwendet werden, mit welchem das Skript „demo\_teamraum.py“ gestartet wird.
- Es muss sichergestellt sein, dass das Buildout korrekt ausgeführt worden ist.  
Falls Daten fehlen, muss ein Exception error geworfen werden.
- Zu Demonstrationszwecken werden wichtige Aktionen bei der Ausführung des Skripts markiert.  
Diese Implementation wird bei der Einspielung auf einem Server wieder entfernt.

## 15.4 Vorarbeiten

Um die Skripts „demo\_teamraum.py“ und „apply\_profile.py“ sowie das Profil erstellen zu können, werden zuerst die benötigten Repositories geklont.

Das Repository eGov wird von Github geklont und der vorgegebenen Struktur hinzugefügt.

```
git clone git@github.com:4teamwork/eGov.git eGov
```

Der Branch des Repository eGov wird erstellt.

```
git checkout -b mw-ipa_profile_demo-content
```

Das Repository buildout-base wird von GitLab geklont und der vorgegebenen Struktur hinzugefügt.

```
git clone git@git.4teamwork.ch:egov/buildout-base.git
```

Der Branch des Repository buildout-base wird erstellt.

```
git checkout -b mw-ipa_runscript
```

## 15.5 Skript „demo\_teamraum.py“ erstellen

Nachfolgend werden die wichtigsten Implementationen des Skripts „demo\_teamraum.py“ aufgezeigt. Das vollständige Skript „demo\_teamraum.py“ ist als Anhang 3 beigelegt.

### 15.5.1 Hauptelemente Skript „demo\_teamraum.py“

Der Grundaufbau des Skript „demo\_teamraum.py“ beinhaltet folgende Hauptelemente:

- Instanzenordner erstellen
- Repository buildout-base in den Instanzenordnern herunterladen
- Symbolischer Link auf development.cfg erstellen
- Interpreter Bootstrap ausführen
- Buildout ausführen
- Instanz starten

### 15.5.2 Bibliotheken Skript „demo\_teamraum.py“

Verschiedene Bibliotheken müssen im Skript „demo\_teamraum.py“ implementiert werden. Für die untenstehenden Bibliotheken werden nur die benötigten Funktionen erläutert.

Modul	Funktion
os	Kommandozeilenargumente können eingegeben werden.
git	Repositories können geklont und angesprochen werden.
sys	Stellt Informationen in Konstanten, Funktionen und Methoden über den Python-Interpreter zur Verfügung.
optparse	Optionen können mitgegeben werden.

Abbildung 18: Bibliothek im Skript

### 15.5.3 Wichtige Elemente des Quellcodes

#### Zeichensatz

Damit beim Ausführen des Skripts „demo\_teamraum.py“ keine Fehlermeldung betreffend Umlauten geworfen werden, muss die erste Zeile folgenden Eintrag enthalten:

```
# -*- coding: utf-8 -*-
```

#### Hervorhebung

Um interessante Stellen im Skript „demo\_teamraum.py“ hervorzuheben, wurde die Funktion „printstars“ erstellt.

```
def printstars(toprint):
    print "*" * 20
    print toprint
    print "*" * 20
```

## OptionParser

Der OptionParser ist ein Modul der Bibliothek Optparse. Er dient dazu, einem Python-Skript Optionen und Argumente mitzugeben. In diesem Projekt wurde der OptionParser verwendet, um als Argument den Instanzenordner, als Option den Pfad zu einem Python-Interpreter mitgeben zu können und eine Hilfestellung zu generieren.

Mit folgender Eingabe wird dem OptionParser die Option „pythonpath“ hinzugefügt. Mit „-p“ wird die Option initialisiert.

```
parser.add_option("-p", dest="pythonpath")
```

Der OptionParser wird instanziiert.

```
options, args = parser.parse_args()
```

Die Option „pythonpath“ wird in die Konstante „PYTHONPATH“ gespeichert.

```
PYTHONPATH = options.pythonpath
```

Folgende Standard-Option wird dem OptionParser mitgegeben, sollte keine Option definiert worden sein.

```
parser.set_defaults(pythonpath=sys.executable)
```

In diesem Fall wird der Python-Interpreter verwendet, mit welchem das Skript „demo\_teamraum.py“ ausgeführt wurde. Jedoch ist zu empfehlen, den Python-Interpreter per Option mitzugeben da nicht garantiert werden kann, dass der Python-Interpreter mit welchem das Skript „demo\_teamraum.py“ ausgeführt wird, funktionieren wird.

Falls das Skript „demo\_teamraum.py“ ohne Option ausgeführt wird, wird möglicherweise eine Fehlermeldung ähnlich der Untenstehenden geworfen.

```
TypeError: unsupported operand type(s) for +: 'NoneType' and 'str'
```

Um eine kleine Hilfestellung hinzuzufügen, wird der OptionParser folgendermassen erweitert:

```
parser.add_option("-p", dest="pythonpath", help="If you dont set a path to the python, it will take the system python" )
```

Die Hilfestellung kann mit folgendem Aufruf eingesehen werden:

```
python demo-teamraum.py -h
```

Das Argument, um den Instanzennamen anzugeben, wird am Ende der Eingabe mitgegeben.

```
python demo-teamraum.py -o /Users/mischu/Plone/python/python-2.7/bin/python2.7  
demo-teamraum
```

## Sicherheitsprüfungen

Um zu prüfen, ob der OptionParser einen Instanzenordner erhalten hat, wurde folgende Prüfung eingebaut:

```
if len(args) != 1:
```

Sollte kein Instanzenordner angegeben worden sein, wird eine Fehlermeldung geworfen und das Skript beendet.

```
    parser.print_usage()
    print "ERROR: You must set a target directory"
    sys.exit()
```

Um keine existierenden Instanzen zu überschreiben, wurde in einer Schleife geprüft, ob der Instanzenordner schon vorhanden ist.

```
if not os.path.exists(DIRNAME):
    os.makedirs(DIRNAME)
    print "Folder with name '%s'" % DIRNAME
else:
    print "Folder already exists: '%s'" % DIRNAME
    sys.exit()
```

Um zu prüfen, ob die Hauptelemente der Instanz nach Durchführen von Buildout enthalten sind, wurde folgende Prüfung eingebaut:

```
foldercontents = os.listdir('.')
for foldername in ['bin', 'parts', 'buildout.cfg']:
    if foldername not in foldercontents:
        raise Exception('missing: ' + foldername)
```

### 15.5.4 Zusammenfassung

Das Skript „demo\_teamraum.py“ ist das Hauptskript und setzt lokal eine Plattform auf. Dies wird im nachfolgenden Programmablaufplan dargestellt.

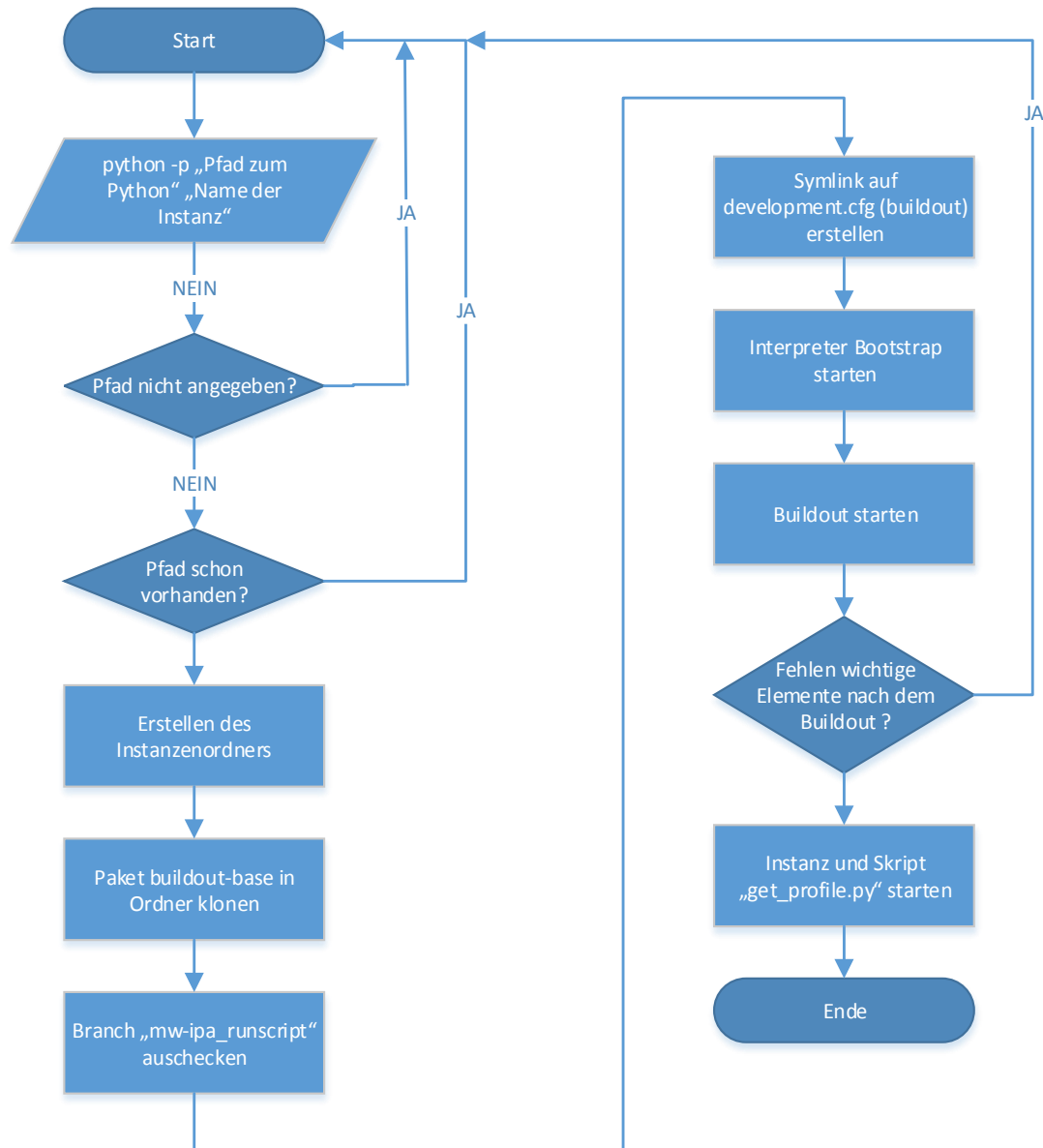


Abbildung 19: Programmablaufplan



## 15.6 Profil „teamraum-demo“ definieren im Repository „eGov“

Um der Plattform Inhalte mitzugeben, wird das Profil „teamraum-demo“ im Repository „eGov“ erstellt.

### 15.6.1 Hauptelemente des Profils

Der Grundaufbau des Profils (siehe Kapitel 14.1.1 für Informationen zum Profil) teamraum-demo sieht folgendermassen aus:

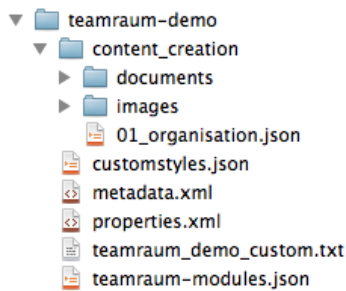


Abbildung 20: Grundaufbau des Profils

### Erläuterung der Struktur

Die Elemente des Profils, welche für dieses Projekt eine Rolle spielen, werden unten erläutert.

Datei	Funktion
teamraum-demo	Stellt den Namen des Profils dar.
Content_creation	Enthält alle visuellen Daten, welche als Inhalt des Profils dienen.
documents	Dokumente, welche für den Inhalt benötigt werden, sind innerhalb „documents“ definiert.
images	Bilder, welche für den Inhalt benötigt werden, sind innerhalb „images“ definiert.
01_organisation.json	Alle Inhaltstypen und deren Metadaten werden dort hinzugefügt. Auch wird die Struktur erstellt.
customstyles.json	Einige Styling-Elemente sind dort eingetragen.
teamraum-modules	Die gewünschten Module werden in dieser Datei aktiviert.

Tabelle 17: Erläuterung der Struktur

Auf die wichtigsten Elemente wird nachfolgend noch genauer eingegangen, um aufzuzeigen, wie Inhalt effektiv definiert und hinzugefügt wird.

#### 15.6.1.1 teamraum-modules.json

Um Module zu aktivieren, werden diese in der Datei teamraum-modules.json auf active: true gesetzt.

```

},
"contact": {
  "active": true,
  "creators": []
},

```

Abbildung 21: Module

### 15.6.1.2 01\_organisation.json

Die Struktur von Kapitel 14.1.1 wird in dieser Datei eingetragen. Als Beispiel wird der Aufbau einer Inhaltsseite und eines Kontakteordners aufgezeigt.

```
{
  "_path": "wirtschaft_gewerbe",
  "_type": "ContentPage",
  "title": "Wirtschaft Gewerbe",
  "_properties": {
    "layout": ["string", "authorities_view"]
  }
},
```

Abbildung 22: Aufbau Inhaltsseite 1

```
{
  "_path": "kontakte",
  "_type": "Folder",
  "title": "Kontakte",
  "_properties": {
    "layout": ["string", "contacts_view"]
  }
},
```

Abbildung 23: Aufbau Inhaltsseite 2

In der nachfolgenden Tabelle werden die wichtigsten Elemente der Datei „01\_organisation.json“ aufgezeigt.

Metadaten	Funktion
_path	Der Pfad, wo die Inhaltsseite / der Kontakteordner hinzugefügt werden soll, ist hier anzugeben.
_type	Die Bezeichnung des aktuellen Inhaltstyps.
title	Der Titel, welcher die Inhaltsseite / Kontakteordner erhält.
properties	Optionen, welche dem Inhaltstyp mitgegeben werden können.
layout	Eine Option, welche die Ansicht für die Inhaltsseite / den Kontakteordner definiert.

Tabelle 18: Wichtigste Elemente der JSON-Datei

### 15.6.2 Abhängigkeit

Ohne Anpassung am Buildoutsript „development.cfg“, welches sich im Repository „buildout-base“ befindet, wird das neu erstellte Profil nicht in die Installationsroutine eingebunden. Auf dem Branch mw-ipa\_runscript wird das Buildout mit der folgenden Zeile erweitert:

```
[branches]
eGov = add_profile_for_eGov_core
```

Nun kann die Plattform automatisch aufgesetzt und anschliessend das Profil „teamraum-demo“ ausgewählt werden, welches die Plattform mit Inhalten füllt. Damit die Vollautomation gewährleistet ist, wird das Skript „apply\_profile.py“ im Branch „mw-ipa\_runscript“ des Repository „eGov“ erstellt. Um die Benutzer automatisch hinzuzufügen, bedarf es noch weiterer Anpassungen, welche als Quellcode im Anhang 7 beigefügt sind.

## 15.7 Skript „apply\_profile.py“ erarbeiten in einem Branch des Repository „buildout-base“

Der nächste Schritt der Realisierung, beinhaltet das Erstellen des Skripts „apply\_profile.py“. Dieses übernimmt folgende Hauptaufgaben:

- Das Aussehen der Plattform wird mit einem Theme definiert
- Root, um Plattform korrekt aufsetzen zu können, wird vergeben
- Rechte werden vergeben
- Ein definiertes Profil wird angewählt
- Der Name der Plattform wird vergeben

Das erarbeitete Skript ist als Anhang 6 diesem Dokument beigelegt.

## 15.8 Benutzeranleitung

Mit dem vorgehenden Kapitel ist die Realisation abgeschlossen und die Demo-Plattform wird automatisiert aufgesetzt.

Die interne Benutzeranleitung beschreibt, wie das Skript „demo-teamraum.py“ erfolgreich gehandhabt wird. Die Benutzeranleitung ist als Anhang 2 diesem Dokument beigelegt.

## 15.9 Testprotokoll

Die während dem Abarbeiten der Testfälle festgestellten Mängel, werden in der folgenden Tabelle festgehalten.

Um einen groben Überblick zu verschaffen, dient die Spalte „Status“. Falls der Test erfolgreich war, ist die Zelle grün und falls der Test fehlschlägt, rot eingefärbt.

Nr.	Test-objekt	Kurz-beschrieb	Effektives Resultat	Visum	Ok?
1s	Skript „demo_teamraum.py“	Skript ausführen ohne Ordnernamen und ohne Pfad zu einem Python-Interpreter.	Das Skript wird nicht durchgeführt.	TB	OK
2s	Skript „demo_teamraum.py“	Skript ausführen mit Instanzennamen, und ohne Pfad zu einem Python-Interpreter.	Das Skript kann das Buildout nicht durchführen und wirft folgende Fehlermeldung: While Installing Loading extensions Error: There is a version conflict	TB	NOK
3s	Skript „demo_teamraum.py“	Skript ausführen mit Instanzennamen, mit Pfad zu einem Python-Interpreter und mit Instanzenordner.	Das Skript wird durchgeführt und wichtige Meldungen werden angezeigt.	TB	OK

4s	Skript „demo_team-raum.py“	Zweimaliges Ausführen des Skripts mit demselben Ordnername.	Das Skript wird unterbrochen mit der Meldung: Folder already exists: 'teamraum-buildout'	TB	OK
5s	Skript „demo_team-raum.py“	Skript mehrmals gleichzeitig ausführen.	Das Skript wird unterbrochen mit der Meldung: Folder already exists: 'teamraum-buildout'	TB	OK
6p	Plattform	Vorhandensein der Plattform.	Die Plattform ist nach Durchführung des Skripts vorhanden.	TB	OK
7p	Plattform	Vorhandensein der Struktur.	Die Struktur ist vorhanden, doch sind die Views noch nicht korrekt eingestellt.	TB	NOK
8p	Plattform	Vorhandensein der Benutzer.	Die Benutzer sind vorhanden.	TB	OK
9p	Plattform	Kontrolle der Berechtigungen.	Die Berechtigungen scheinen in Ordnung zu sein. Eingeloggt mit allen drei Benutzern und die Berechtigungstabelle kontrolliert.	TB	OK

Tabelle 19: Testprotokoll

## 15.10 Testauswertung / Testabschluss

Die Tests konnten allesamt durchgeführt werden. Der Test 7p wurde nach Behebung des Problems nochmals erfolgreich durchgeführt.

Die Testfälle konnten anschliessend mit Erfolg abgeschlossen werden.

## 15.11 Problemanalyse und -behebung

Wie bei der Testauswertung ersichtlich, mussten die Punkt 2s und 7p nochmals bewertet werden. Die entstandenen Fehler wurden folgendermassen gehandhabt:

### 2s

Es kann nicht garantiert werden, dass ein funktionierender Python-Interpreter vorhanden ist. Deshalb muss die ausführende Person die benötigten Grundlagen schaffen, um das Skript ohne Konflikte auszuführen. Das Skript kann nicht jede Eventualität abdecken (siehe Projektjournal vom 26.05.2014).

### 7p

Um die Views korrekt einzustellen, bedarf es jeweils einer kleinen Anpassung an der Profildatei „01\_organisation.json“.

Für die verschiedenen Inhaltstypen sind verschiedene Views definiert. Die korrekte View muss für den korrekten Inhaltstyp verwendet werden. Für einen Workspace muss z.B. die View „workspace\_view“ verwendet werden.

"layout": ["string", "workspaces\_view"]

## 16. Quellenverzeichnis

Quelle	Autor	Thema
Buch: Modul 306: IT-Kleinprojekt abwickeln TeilB, 43 bis 71 Auflage2 Jahr: 2007	Kurt Dadertscher / Peter Hirzel	Informationen betreffend Projekten
Internet: <a href="http://de.wikipedia.org/wiki/Lenkungsausschuss">http://de.wikipedia.org/wiki/Lenkungsausschuss</a> Jahr: 2007 Zugriff: 05.05.2014	Pseudonym: CatcherBlock	Lenkungs-ausschuss
Internet: <a href="http://books.google.ch/books?id=FBnZVpUMJzMC">http://books.google.ch/books?id=FBnZVpUMJzMC</a> Jahr: 2008 Zugriff: 06.05.2014	Hans W. Wiczorrek / Peter Mertens	Situations-analyse
Internet: <a href="http://www.igrm.ch/app/download/2471810817/Zugriffskonzept+BK.pdf?t=1289218378">http://www.igrm.ch/app/download/2471810817/Zugriffskonzept+BK.pdf?t=1289218378</a> Jahr: Unbekannt Zugriff: 07.05.2014	Zosso Karin	Zugriffskonzept
Internet: <a href="https://plone-3-benutzerhandbuch.readthedocs.org/en/latest/glossar.html">https://plone-3-benutzerhandbuch.readthedocs.org/en/latest/glossar.html</a> Jahr: Unbekannt Zugriff: 19.05.2014	Jan Ulrich Hasecke und Thomas Lotze	Benutzer-handbuch Plone
Internet: <a href="http://www.4teamwork.ch/loesungen">http://www.4teamwork.ch/loesungen</a> Jahr: 2013 Zugriff: 19.05.2014	Bernhard Bühlmann	Produkte 4teamwork
Internet: <a href="http://www.python-kurs.eu/sys_modul.php">http://www.python-kurs.eu/sys_modul.php</a> Jahr: Unbekannt Zugriff: 20.05.2014	Bernd Klein	sys-Modul für Python

## 17. Glossar

Begriff	Bedeutung
Brainstorming	Eine Methode zur Ideenfindung.
Branch	Eine Abspaltung von einer Version, damit unterschiedliche Versionen parallel weiterentwickelt werden können.
Buildout	Ein Python-basiertes Buildsystem zur automatischen Installation von Webanwendungen.

CMS	Ein <b>Content Management System</b> ist ein Programm, welches eine gemeinschaftliche Nutzung und Manipulation von Daten ermöglicht und organisiert.  Weitere Informationen: <a href="http://de.wikipedia.org/wiki/Content_Management_System">http://de.wikipedia.org/wiki/Content_Management_System</a>
Git	Webbasierte Verteilungs- und Versionierungssoftware
Github	Hosting Dienst für Software-Entwicklungsprojekte
Human Testing	Von Menschen und nicht von Maschinen getestet.
Inhaltstypen	Zum Beispiel „Datei“, „Ordner“, usw. Diese vereinfachen das Erstellen von Strukturen.
JSON	Ist ein kompaktes Datenformat, welches in einer einfach lesbaren Textform dargestellt ist. Es dient zum Datenaustausch zwischen Anwendungen.
Kanban	Ist eine Methode zur Produktionsprozesssteuerung.
Plone	Ist ein Content Management System, welches auf Zope basiert.  Weitere Informationen: <a href="http://de.wikipedia.org/wiki/Plone">http://de.wikipedia.org/wiki/Plone</a>
Python	Python ist eine objektorientierte Programmiersprache, welche auf fast allen Betriebssystemen installiert werden kann.  Weitere Informationen: <a href="https://www.python.org">https://www.python.org</a>
Releaseprozess	Stellt eine Sammlung von Konfigurationsänderungen zu einem Paket / Release dar und stellt die ordnungsgemäße Eingliederung in der Infrastruktur sicher.
Repository	Ist ein Verzeichnis zur Speicherung und Beschreibung von digitalen Objekten.
Review-Prozess	Ein Arbeitsschritt wird von mehreren Mitarbeitern angeschaut, um Fehler ausmerzen zu können.
Terminal	Eine Kommandozeile von Apple entwickelt für Mac.
Usability	Benutzerfreundlichkeit / Anwenderfreundlichkeit.
VMWare	Virtuelle Maschinen können auf Computern betrieben werden.
Zope	Zope ist ein Open-Source Webserver, welcher auf Python basiert.  Weitere Informationen: <a href="http://www.zope.org">http://www.zope.org</a>

Unterschriften für die Abnahme.

Datum	Name / OE	Unterschrift
27.05.2014	Michel Weingart	
27.05.2014	Pascal Habegger	



## 18. Anhang

---

### Anhang 1: Wichtigste Inhaltstypen

Inhaltstyp	Funktion
AdressBlock	Im Adressenblock kann eine Adresse hinterlegt werden. Die Ausgabe enthält zusätzlich den Standort, aufgezeigt in einer GoogleMap.
Workspace	Innerhalb eines Arbeitsraumes werden Daten ausgetauscht. Der Arbeitsraum kommt hauptsächlich bei Projekten zum Einsatz.
ListingBlock	Mehrere Dateien können im Auflistungsblock zum Download zur Verfügung gestellt werden. Zudem wird der Auflistungsblock oft auch als Bildergalerie verwendet.
ContentPage	Ist das Strukturierungselement der Plattform. innerhalb der ContentPage werden Inhalte, wie Textblöcke, Auflistungsblöcke, etc. erfasst.
Folder	Dient innerhalb von Arbeitsräumen zur Strukturierung.
NewsFolder	News werden innerhalb des NewsFolders erfasst und können in einem dynamischen Portlet dargestellt werden.
Subsite	Eine Seite, welche in hierarchischer Beziehung zu der Hauptseite steht.
TextBlock	Ein Element, um Texte zu erfassen und zu gliedern.

**Tabelle 20: Wichtigste Inhaltstypen**

## Anhang 2: Benutzeranleitung

# Aufsetzen einer Demo-Plattform

---

## Inhaltsverzeichnis

Inhaltsverzeichnis.....	82
Abbildungsverzeichnis.....	Fehler! Textmarke nicht definiert.
Tabellenverzeichnis.....	Fehler! Textmarke nicht definiert.
1. Einführung.....	83
2. Anforderungen .....	83
3. Vorarbeiten .....	83
4. Ausführen des Skripts „demo_teamraum.py“ .....	84
5. Authentifizierung auf der Plattform .....	84

# 1. Einführung

Die folgende Benutzeranleitung ist für Mitarbeitende von 4teamwork gedacht, welche zu Testzwecken eine Demo-Plattform mit dem neusten Release aufsetzen wollen. Die Anleitung befindet sich noch nicht in der definitiven Version, da das Skript noch nicht für den Produktivbetrieb freigegeben ist.

# 2. Anforderungen

Folgende funktionalen Anforderungen müssen erfüllt sein, damit die Demo-Plattform erfolgreich aufgesetzt wird:

Nr.	Anforderungen
1	Ein Python-Interpreter der Version 2.7.5 muss installiert sein, welcher mit Plone umgehen kann.
2	Das Skript muss mit einem Benutzer ausgeführt werden, welcher die entsprechenden Berechtigungen besitzt.
3	Der Benutzer muss Zugriffe auf die folgenden Repositories haben: <ul style="list-style-type: none"><li>- <a href="https://git.4teamwork.ch/egov/buildout-base">https://git.4teamwork.ch/egov/buildout-base</a></li><li>- <a href="http://github.com/4teamwork/eGov">http://github.com/4teamwork/eGov</a></li></ul>

# 3. Vorarbeiten

Das Skript befindet sich im Repository buildout-base, welches als erstes mit einem Terminalbefehl lokal heruntergeladen werden muss. Das Repository wird mit dem nachfolgenden Befehl in den Ordner „demo-teamraum“ an den Standort heruntergeladen, welcher aktuell im Terminal aktiv ist.

```
git clone git@git.4teamwork.ch:egov/buildout-base.git demo-teamraum
```

Anschliessend muss mit nachfolgendem Befehl auf den Branch gewechselt werden, auf welchem sich das Skript befindet.

```
git checkout mw-ipa_runscript
```

## 4. Ausführen des Skripts „demo\_teamraum.py“

1. Das Skript befindet sich im Verzeichnis „demo-teamraum“ / scripts.
2. In das Verzeichnis wechseln und in der Konsole folgenden Befehl eingeben, um das Skript zu starten:

```
python demo-teamraum.py -p Standort des Python-Interpreters Pfad
```

Ein beliebiger Pfad kann angegeben werden.

Falls konfiguriert, befindet sich dieses meist unter:  
„/Users/name/Plone/python/python-2.7/bin/“.  
Falls diese Option nicht angegeben wird, wird der  
Python-Interpreter verwendet, welcher zur Ausführung  
des Skripts verwendet wird, was aber möglicherweise  
nicht erfolgreich sein wird.

Option

Name des Skripts

Der aktuell verwendete Python-  
Interpreter

## 5. Authentifizierung auf der Plattform

1. Über localhost:8080 wird die neu aufgesetzte Plattform angezeigt.  
<http://localhost:8080/gemeinde>
2. Der Zugang erfolgt über „Ihre Plone-Webseite anzeigen“.



### Plone läuft.

 [Ihre Plone-Webseite anzeigen](#)

[Zope-Management-Interface](#) — systemnahe Konfiguration

Dokumentation, Erweiterungen, Support und die Community finden Sie auf  
[plone.org](http://plone.org).

3. Nachfolgend sind die drei Benutzer und derer Zugangsdaten, welche für die Plattform definiert sind, beschrieben.

Benutzer	Funktion	Loginname	Passwort
Test Administrator	Administriert die Inhalte der Plattform. Er kann die Struktur anpassen und Module hinzufügen.	test_administrator	Admin_4teamwork.ch
Test Author	Kann Inhalte erfassen / bearbeiten und löschen.	test_author	Author_4teamwork.ch
Test Gast	Hat Leserechte auf Inhalte.	test_gast	Gast_4teamwork.ch

Anschliessend kann mit den bereitgestellten Benutzern individuell getestet werden.

### Anhang 3: Skript „demo\_teamraum.py“

```
# -*- coding: utf-8 -*-

# Damit die Pakete von Git-Systemen geklont werden können.
import git
# Bibliothek um z.B Konsolenbefehle übergeben zu können
# os bedeutet Operating System.
import os
# Bibliothek um Interpreter Python ansprechen zu können z.B
# um eine maximale Rekursionstiefe zu erfragen.
import sys
# Der OptionParser dient dazu, einem Skript Argumente und
# Optionen mitgeben zu können.
from optparse import OptionParser

# Das Paket buildout-base erhält die Konstante REPOURL.
# Der Pfad zum Git Repository wird in dieser Konstante
# gespeichert.
REPOURL = 'git@git.4teamwork.ch:egov/buildout-base.git'

# Die nachfolgende Funktion dient dazu, wichtige Abschnitte zu
# Demonstrationzwecken, im Output zu markieren.
def print_stars(toprint):
    print "*" * 20
    print toprint
    print "*" * 20

# Der OptionParser wird instanziiert.
parser = OptionParser()

# sys.executable -> Der Python-Interpreter, mit welchem das
# Skript gestartet wird, falls keine Option mitgegeben wurde.
parser.set_defaults(pythonpath=sys.executable)

# Option, um den Python-Pfad angeben zu können, wird hinzugefügt.
# dest -> Unter welcher Option "pythonpath" verfügbar gemacht werden soll.
parser.add_option("-p", dest="pythonpath",
                  help="Falls kein Pfad zum Python-Interpreter angegeben
wird,"
                  " wird der Interpreter verwendet, mit welchem das Skript"
                  " angestossen wurde")

# Optionen und Argumente werden hier geparkt.
options, args = parser.parse_args()

# Es muss ein Argument angegeben werden, sonst wird mit
# einem sys.exit die Ausführung des Skripts abgebrochen.
if len(args) != 1:
    parser.print_usage()
    print "ERROR: You must set a target directory"
    sys.exit()

# Argument wird in der Konstante DIRNAME gespeichert.
DIRNAME = args[0]

# Der Pfad des Python-Interpreter wird dem OptionParser entnommen und
# in der Konstante PYTHONPATH gespeichert.
```

```
PYTHONPATH = options.pythonpath

# Überprüfung ob die Instanz schon existiert.
# Sollte dies der Fall sein, wird die Ausführung des
# Skripts abgebrochen.
if not os.path.exists(DIRNAME):
    os.makedirs(DIRNAME)
    print "Folder with name '%s'" % DIRNAME
else:
    print "Folder already exists: '%s'" % DIRNAME
    sys.exit()

# Es wird in das Verzeichnis DIRNAME gewechselt.
os.chdir(DIRNAME)

# Das Repository wird in das aktuelle Verzeichnis geklont.
repo = git.Git()
repo.clone(REPOURL, '.')

# Der Branch vom buildout-base wird ausgecheckt und verwendet.
print "Cloned %s" % REPOURL
repo.checkout("mw-ipa_runscript")

# Ein Symlink auf das development.cfg wird erstellt.
print_stars("Create Symlink")
os.system('ln -s development.cfg buildout.cfg')
print_stars("Created buildout.cfg Symlink to development.cfg")

# Der Interpreter Bootstrap wird ausgeführt.
print_stars("Start bootstrap")
os.system(PYTHONPATH + " bootstrap.py")
print_stars("End bootstrap")

# Das Buildout wird gestartet.
print_stars("Start buildout")
os.system("bin/buildout ")
print_stars("End buildout")

# Es wird geprüft, ob wichtige Elemente im erstellten
# Instanzenpfad existieren. Falls diese nicht existieren,
# wird eine Fehlermeldung geworfen.
foldercontents = os.listdir('.')
for foldername in ['bin', 'parts', 'buildout.cfg']:
    if foldername not in foldercontents:
        raise Exception('missing:' + foldername)

# Die Instanz wird zusammen mit dem Skript get_profile.py
# gestartet.
print_stars("Starting instance")
os.system('bin/instance1 run scripts/apply_profile.py')
os.system('bin/instance1 start')
```

## Anhang 4: Skript „01\_organisation.json“

```
[  
  {  
    "_path": "neues_aktuelles",  
    "_type": "NewsFolder",  
    "title": "Neues Aktuelles"  
  },  
  {  
    "_path": "neues_aktuelles/abwasserreglement_oeffentliche_mitwirkung",  
    "_type": "News",  
    "title": "Revision Abwasserreglement - öffentliche Mitwirkung",  
    "effectiveDate": "2014/05/22 09:00:00"  
  },  
  {  
    "_path":  
"neues_aktuelles/abwasserreglement_oeffentliche_mitwirkung/paragraph1",  
    "_type": "TextBlock",  
    "text": "<p>Das Abwasserreglement der Gemeinde Belp mit  
entsprechendem Abwassertarif ist seit 1. Januar 1990 in Kraft. Seither  
haben die gesetzlichen Grundlagen auf Bundes- und Kantonsebene verschiedene  
Änderungen erfahren. Auch die bautechnischen Normen wurden laufend  
überarbeitet und dem Stand der Technik angepasst. In einer Gesamtrevision  
soll das bisherige Abwasserreglement den übergeordneten Vorschriften  
angepasst werden. Das Musterreglement des Kantons Bern dient dabei als  
Grundlage, wobei einzelne gemeindespezifische Vorschriften aus dem  
bisherigen Reglement übernommen werden. Gleichzeitig mit der Revision des  
Abwasserreglements wird auch das Gebührenmodell den neuen Gegebenheiten  
angepasst. Mit der Einführung einer Regenabwassergebühr, der Anpassung der  
Grundgebühr pro Wohneinheit und Gewerbebetrieb und einer Erhöhung der  
Verbrauchsgebühr sollen die in den letzten Jahren um ca. 1 Mio. CHF  
gestiegenen Kosten im Abwasserbereich gedeckt werden können. Die  
Gemeindeversammlung wird am 4. Dezember 2014 über die Revision des  
Abwasserreglements beschliessen. Vor der Detailberatung im Gemeinderat  
führt die Gemeinde Belp vom 15. Mai bis zum 30. Juni 2014 eine  
Vernehmlassung unter den politischen Parteien und der Bevölkerung durch. Am  
Dienstag, 20. Mai 2014 orientieren Gemeindevertreter um 19.30 Uhr im  
Restaurant Kreuz, Gürbesaal, an einer öffentlichen Orientierungsversammlung  
über die geplanten Änderungen.</p>"  
  },  
  {  
    "_path": "neues_aktuelles/kulturtage_pedro_lenz",  
    "_type": "News",  
    "title": "Kulturtage: Pedro Lenz Ausverkauft!",  
    "effectiveDate": "2014/05/22 09:00:00"  
  },  
  {  
    "_path": "neues_aktuelles/kulturtage_pedro_lenz/paragraph1",
```



```

    "_type": "TextBlock",
    "text": "<p>Ab sofort können keine Reservationen mehr für die Lesung
mit Pedro Lenz aufgenommen werden.Sie haben die Möglichkeit, sich auf eine
Warteliste setzen zu lassen. Falls auf der Gemeinde Absagen eintreffen,
rücken Personen der Warteliste (nach Aufnahmedatum ihrer Reservation) nach.
Wir nehmen alsdann telefonisch mit diesen Personen Kontakt auf. </p>"
  },

  {
    "_path": "lebensqualitaet",
    "_type": "ContentPage",
    "title": "Lebensqualität",
    "_properties": {
      "layout": ["string", "simplelayout"]
    }
  },

  {
    "_path": "lebensqualitaet/jugend",
    "_type": "ContentPage",
    "title": "Jugend",
    "_properties": {
      "layout": ["string", "simplelayout"]
    }
  },

  {
    "_path": "lebensqualitaet/jugend/prix_belp",
    "_type": "TextBlock",
    "text": "<p>Seit der Gemeindeversammlung 2010 wird die Verleihung des
¬´Prix Belp¬`a und die Ehrung erfolgreicher Belper nach den vom Gemeinderat
Belp am 24. Juni 2010 genehmigten neuen Richtlinien vorgenommen.Mit einer
Mischvariante zwischen der 2007 ins Leben gerufenen Verleihung des ¬´Prix
Belp¬`a und des bis Ende 2006 durchgeführten ¬´Empfangs erfolgreicher
Belper¬`a gelingt es, auch sportliche bzw. messbare Leistungen einzubeziehen
und die Würdigung für mehrere Nominierte zu einem besonderen Moment zu
gestalten. </p>"
  },

  {
    "_path": "verwaltung",
    "_type": "ContentPage",
    "title": "Verwaltung",
    "_properties": {
      "layout": ["string", "simplelayout"]
    }
  },

  {
    "_path": "verwaltung/adresse",
    "_type": "AddressBlock",
    "title": "Adresse",
    "showContactPortlet": true,
    "address": "Gartenstrasse 2",

```

```
"zip": "3012",
"city": "Belp",
"phone": "031 818 22 22",
"fax": "031 818 22 12",
"email": "info@belp.ch",
"www": "http://www.belp.ch",
"showOpeningHours": true,
"openingHours": "Montag bis Freitag\n08.00 - 11.30 und 13.30 - 17.00
Uhr\n(Freitag bis 16.30)"
},

{
  "_path": "verwaltung/listingblock",
  "_type": "ListingBlock"
},

{
  "_path": "verwaltung/listingblock/bild1",
  "_type": "Image",
  "image:file": "images/01.jpg"
},

{
  "_path": "verwaltung/listingblock/bild2",
  "_type": "Image",
  "image:file": "images/02.jpg"
},

{
  "_path": "verwaltung/listingblock/bild3",
  "_type": "Image",
  "image:file": "images/03.jpg"
},

{
  "_path": "verwaltung/text",
  "_type": "TextBlock",
  "text": "<p>Belp von seiner schönsten Seite.</p>"
},

{
  "_path": "wirtschaft_gewerbe",
  "_type": "ContentPage",
  "title": "Wirtschaft Gewerbe",
  "_properties": {
    "layout": ["string", "simplelayout"]
  }
},

{
  "_path": "wirtschaft_gewerbe/zusammenarbeit",
  "_type": "TextBlock",
  "text": "<p>Eine gute Zusammenarbeit zwischen Wirtschaft und Politik
```

ist entscheidend, da ähnliche Anliegen verfolgt werden. Es geht in der Wirtschaft wie in der Politik um die Erhaltung und Schaffung von Arbeits- und Ausbildungsplätzen. Deshalb möchte die Gemeinde Belp Sie als Vertreter der Wirtschaft, als Unternehmerin oder Unternehmer, direkt ansprechen. Sprechen Sie mit uns ,Äi wir sprechen gerne mit Ihnen und haben ein offenes Ohr.Sie sehen in diesem Bereich der Belper Website Infos/Kontakte zu bestehenden Belper Unternehmen wie auch verschiedene Infos zur Wirtschaftsförderung.</p>"

```

    },

    {
      "_path": "politik",
      "_type": "ContentPage",
      "title": "Politik",
      "_properties": {
        "layout": ["string", "simplelayout"]
      }
    },

    {
      "_path": "politik/kraefteverhaeltnis",
      "_type": "TextBlock",
      "text": "<p>Die Gemeinde Belp wird von einem siebenköpfigen Gemeinderat, in dem die fünf Parteien EDU, EVP, SP, SVP und BDP vertreten sind, geführt. Die Entscheide werden von den Gemeindebürgern an der Gemeindeversammlung ,- in der Regel vier pro Jahr ,- getroffen. Sachgeschäfte über 3 Millionen Franken werden an der Urne entschieden.</p>"
    },

    {
      "_path": "politik/aktuell",
      "_type": "NewsFolder",
      "title": "Aktuell"
    },

    {
      "_path": "politik/aktuell/erste_news_meldung",
      "_type": "News",
      "title": "600 Jahre Belp",
      "effectiveDate": "2014/05/22 09:00:00"
    },

    {
      "_path": "politik/aktuell/erste_news_meldung/paragraph1",
      "_type": "TextBlock",
      "text": "<p>Finden Sie aktuelle Information über onegov unter <a href=\"http://www.onegov.ch\" target=\"_blank\">www.onegov.ch</a>, oder folgend Sie uns auf <a href=\"http://twitter.com/OneGovBox\" target=\"_blank\">Twitter</a></p>"
    },

    {
      "_path": "kontakte",
      "_type": "Folder",

```

```
    "title": "Kontakte",
    "_properties": {
      "layout": ["string", "contacts_view"]
    },
    "_constrain_types": {
      "locally": ["Contact"],
      "immediately": ["Contact"]
    }
  },

  {
    "_path": "projekte",
    "_type": "Folder",
    "title": "Projekte",
    "_placefulworkflow": ["workspace_policy", "workspace_policy"],
    "_properties": {
      "layout": ["string", "workspaces_view"]
    },
    "_constrain_types": {
      "locally": ["Workspace"],
      "immediately": ["Workspace"]
    }
  },

  {
    "_path": "projekte/flughafenzufahrt_bern_belp",
    "_type": "Workspace",
    "title": "Flughafenzufahrt Bern-Belp",
    "_properties": {
      "layout": ["string", "workspaces_view"]
    }
  },

  {
    "_path": "projekte/flughafenzufahrt_bern_belp/flughafen",
    "_type": "Image",
    "image:file": "images/02.jpg"
  },

  {
    "_path": "projekte/flughafenzufahrt_bern_belp/evaluation",
    "_type": "Image",
    "image:file": "documents/evaluation.xlsx"
  },

  {
    "_path": "projekte/flughafenzufahrt_bern_belp/it",
    "_type": "Image",
    "image:file": "documents/IT.docx"
  },

  {
    "_path": "projekte/flughafenzufahrt_bern_belp/nachhaltigkeit",
    "_type": "Image",
```

```
    "image:file": "documents/Nachhaltigkeit.pdf"
  },

  {
    "_path": "belpberg",
    "_type": "Subsite",
    "title": "Belpberg",
    "_properties": {
      "layout": ["string", "subsite_view"]
    }
  },

  {
    "_path": "belpberg/portraet",
    "_type": "ContentPage",
    "title": "Porträt",
    "_properties": {
      "layout": ["string", "simplelayout"]
    }
  },

  {
    "_path": "belpberg/portraet/bild",
    "_type": "Image",
    "image:file": "images/06.jpg"
  },

  {
    "_path": "belpberg/portraet/intro",
    "_type": "TextBlock",
    "text": "<p>Belpberg war bis zum 31. Dezember 2011 eine politische
Gemeinde im Verwaltungskreis Bern-Mittelland des Kantons Bern in der
Schweiz. Die Gemeindeverwaltung befand sich im Ortsteil Linden.</p>"
  }
]
```

## Anhang 5: Skript „teamraum-modules.json“

```
{
  "banner": {
    "active": true,
    "creators": [
      "create_default_banner_image"
    ]
  },
  "classification": {
    "active": true,
    "creators": []
  },
  "cms": {
    "active": true,
    "creators": []
  },
  "contact": {
    "active": true,
    "creators": []
  },
  "form": {
    "active": true,
    "creators": []
  },
  "workspace": {
    "active": true,
    "creators": []
  }
}
```

## Anhang 6: Skript „apply\_profile“

Dieses Skript wird in der Realisation nicht sehr detailliert beschrieben, da die Erarbeitung hauptsächlich durch einen weiteren Entwickler stattgefunden hat.

```
# -*- coding: utf-8 -*-

from AccessControl.SecurityManagement import newSecurityManager
import transaction
from Products.CMFPlone.factory import _DEFAULT_PROFILE
from Products.CMFPlone.factory import addPloneSite

# Die nachfolgenden Profile werden automatisch installiert.
EXTENSION_PROFILES = (
    'plonetheme.classic:default',
    'plonetheme.sunburst:default',
)

def setup():
    # Wurzelverzeichnis wird in app gespeichert.
    app = globals()['app']

    # Übernehmen des admin-Benutzers Zopemaster.
    user = app.acl_users.getUser('zopemaster')
    user = user.__of__(app.acl_users)
    # Der SecurityManager vergibt Rechte.
    newSecurityManager(app, user)

    app = makerequest(app)
    globals()['app'] = app

def install_teamraum():
    """Installiert einen Teamraum"""
    # Standard Profil von Plone wird mit dem Profil
    # teamraum-demo erweitert.
    extensions = list(EXTENSION_PROFILES)
    extensions.append('egov.core:teamraum-demo')

    site = addPloneSite(
        # Plone-Seite wird hinzugefügt und bekommt den Titel
        # und die Sprache Deutsch.

        globals()['app'],
        'gemeinde',
        title='Gemeinde Belp',
        profile_id=_DEFAULT_PROFILE,
        extension_ids=extensions,
        setup_content=False,
        default_language='de')

if __name__ == '__main__':
    setup()
    install_teamraum()
    print 'Committing'
    transaction.commit()
    print 'Die Plattform wurde erfolgreich aufgesetzt'
```

## Anhang 7: Skript „setuphandlers.py“

Dieses Skript wird in der Realisation nicht beschrieben, da die Erarbeitung durch einen weiteren Entwickler stattgefunden hat. Es sind nur die Wichtigsten Abschnitte des Skripts aufgeführt.

```

DEFAULT_USERS = [
    {'username': 'administrator',
      'password': 'Admin_4teamwork.ch',
      'firstname': 'Test',
      'lastname': 'Administrator',
      'email': 'test.admin@email.com',
      'roles': ('Administrator',
                 'Site Administrator',
                 'Reader',
                 'Contributor',
                 'Editor',
                 'Member')
    },
    {'username': 'author',
      'password': 'Author_4teamwork.ch',
      'firstname': 'Test',
      'lastname': 'Author',
      'email': 'test.author@email.com',
      'roles': ('Editor', 'Contributor', 'Reader', 'Member')
    },
    {'username': 'gast',
      'password': 'Gast_4teamwork.ch',
      'firstname': 'Test',
      'lastname': 'Gast',
      'email': 'test.gast@email.com',
      'roles': ('Reader', 'Member', 'Member')
    },
]

def create_teamraum_demo_user(context):

    if context.readDataFile('teamraum_demo_custom.txt') is None:
        return

    portal = context.getSite()
    regtool = getToolByName(portal, 'portal_registration')

    for user in DEFAULT_USERS:
        properties = {
            'username': user['username'],
            'fullname': '{0} {1}'.format(user['firstname'],
user['lastname']),
            'firstname': user['firstname'],
            'lastname': user['lastname'],
            'email': user['email']}

        regtool.addMember(
            user['username'],
            user['password'],
            roles=user['roles'],
            properties=properties)

```