IPA Dokumentation

IT Support

|  |  |  |  |
| --- | --- | --- | --- |
| Kunde | 4teamwork AG | | |
| Projektname | Automatische Installation von Demo-Webplattformen | | |
| Autor | Michel Weingart | | |
| Ausgabedatum | XXXXX | | |
| Version | V 0.8 | | |
| Status | In Arbeit | In Prüfung | Genehmigt, zur Nutzung |
|  |  |  |  |

**4teamwork**

**IPA**

**Michel Weingart**

| Beteiligter Personenkreis | |
| --- | --- |
| Benutzer, Anwender | Michel Weingart, IT Support |
| Prüfung | Pascal Habegger, Fachvorgesetzter Manuel J. Schaffner, Hauptexperte Verena Meinung, 2. Expertin |
| Genehmigung | Yvan Kolly, Validexperte |
| zur Information, Kenntnis | Josef Müller, Berufsbildner |

Dokumentinformationen

Änderungskontrolle, Prüfung, Genehmigung

In der Änderungskontrolle werden nur Punkte aufgelistet, welche vollständig in der Dokumentation enthalten ist, weshalb der Arbeitsplan nicht 100% mit der Änderungskontrolle übereinstimmen wird.

|  | | | |
| --- | --- | --- | --- |
| Version | Datum | Name | Beschreibung |
| Vorlage | 05.05.2014 |  | IPA\_Dokumentation\_2014-final |
| V0.1 | 05.05.2014 | Michel Weingart | Dokumenteröffnung |
| V0.2  V0.3  V0.4  V0.5  V0.6  V0.7  V0.8 | 06.05.2014  07.05.2014  08.05.2014  19.05.2014  20.05.2014  21.05.2014  22.05.2014 | Michel Weingart  Michel Weingart  Michel Weingart  Michel Weingart  Michel Weingart  Michel Weingart  Michel Weingart | Abschluss der Phase Initialisierung  Mehrere Konzepte erarbeitet  Abschluss der Phase Konzept  Erarbeitung des Skripts angefangen  Skript „demo-teamraum.py“ erstellt  Profil und Runskript erstellt |

Verwendete Abkürzungen

| Abkürzung | Bedeutung |
| --- | --- |
| IPA | Individuelle praktische Arbeit |
| xx | xx |
| xx | xx |
| xx  PEP | xx  Python Enhancement Proposal |

Inhaltsverzeichnis

Dokumentinformationen 3

Inhaltsverzeichnis 4

Abbildungsverzeichnis 7

Tabellenverzeichnis 7

Management Summary 8

Teil 1: Ablauf und Umfeld 9

1. Aufgabenstellung 10

1.1 Ausgangslage 10

1.2 Auftragsformulierung 10

1.3 Mittel und Methoden 11

1.4 Projektmanagementplan 12

1.5 Projektrollen 13

2. Vorkenntnisse 14

3. Vorarbeiten 14

4. Rahmenbedingungen 14

5. Firmenstandards 15

6. Organisation der IPA 15

6.1 Datensicherung der IPA 16

7. Meilensteine 17

8. Zeitplan 18

9. Arbeitsjournal 19

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 32

9.10 Zehnter Tag: Dienstag, 27.05.2014 33

9.11 Arbeitszeit total 34

10. Projektjournal 35

11. Abschlussbericht 40

11.1 Vergleich Ist/Soll 40

11.2 Mittelbedarf 40

11.3 Realisierungsbericht 40

11.4 Testbericht 40

11.5 Fazit zum Projekt 40

11.6 Persönliches Fazit 40

12. Unterschriften Teil 1 41

Teil 2: Projektdokumentation 42

13. Initialisierung 43

13.1 Situationsanalyse 43

13.1.1 Plone 43

13.1.2 Teamraum 43

13.1.3 Github / Git.4teamwork 43

13.2 Studie Ist Zustand / Soll-Zustand 44

13.3 Zeitrahmen / Budget 46

13.4 Übergeordnete Ziele 46

13.5 Chancen 46

13.6 Anforderungen 46

13.7 Anwendungsbereich 47

13.8 Wirtschaftlichkeit 48

13.9 Risikoanalyse 49

13.10 Varianten 49

13.10.1 Variante 1 49

13.10.2 Variante 2 49

13.11 Variantenentscheid 50

13.11.1 Begründung 50

13.12 Lösungen suchen 50

13.13 Freigabe Konzept 52

14. Konzept 53

14.1 Anforderungen 53

14.2 Systemstruktur 53

Physikalische Hardware / Software Server 54

14.3 Zugriffskonzept 55

14.3.1 Inhaltsstruktur 56

14.3.2 Berechtigungen 57

14.4 Betriebskonzept 58

14.4.1 eGov 58

14.4.2 Teamraum Buildout 58

14.4.3 Erstellen und Einpflegen des Skripts 58

14.4.4 Versionierung 58

14.4.5 Hotfixes 59

14.4.6 Betrieb auf dem Server 59

14.5 Testkonzept. 59

14.5.1 Ausgangslage 59

14.5.2 Testmethode 59

14.5.3 Testplanung 59

14.5.4 Testziele 60

Die Hauptziele betreffend den Tests sind nachfolgend aufgelistet: 60

14.5.4.1 Hilfsmittel 60

14.5.5 Testvorbereitung 60

Spezifische Testfälle betreffend Skript 60

Spezifische Testfälle betreffend Plattform 62

14.6 Einführungskonzept 63

14.7 Zusammenfassung 63

14.8 Freigabe Realisierung 64

15. Realisierung 65

15.1 Legende 65

15.2 Bestehende Umgebung verifizieren 65

15.3 Konfiguration festlegen 65

15.4 Vorarbeiten 66

15.5 Installationsskript erstellen 66

15.5.1 Bibliotheken Installationsskript 66

15.5.2 Hauptelemente Installationsskript 66

15.5.3 Wichtige Elemente des Quellcodes 67

15.5.4 Zusammenfassung 68

15.6 Profil definieren im Paket „eGov“ 68

15.6.1 Hauptelemente des Profils 68

15.6.1.1 teamraum-modules.json 69

15.6.1.2 01\_organisation.json 69

15.6.2 Abhängigkeit 70

15.7 Runskript erarbeiten in einem Branch des Pakets „buildout-base“ 70

15.8 Programmablaufplan 71

15.9 Benutzeranleitung / Betriebshandbuch 71

15.10 Testprotokoll 71

15.11 Freigabe Einführung 72

16. Einführung 73

16.1 System einführen 73

17. Quellenverzeichnis 74

18. Glossar 74

19. Unterschriften für Abnahme 76

20. Anhang 77

Abbildungsverzeichnis

[Abbildung 1: Logo 4teamwork AG 10](file:///H:\IPA_michel.weingart\dokumente\IPA_Dokumentation_V0.7.docx#_Toc388449899)

[Abbildung 2: Hermes 5 IPA 12](file:///H:\IPA_michel.weingart\dokumente\IPA_Dokumentation_V0.7.docx#_Toc388449900)

[Abbildung 3: Organigramm Projekt 13](file:///H:\IPA_michel.weingart\dokumente\IPA_Dokumentation_V0.7.docx#_Toc388449901)

[Abbildung 4: Mein Arbeitsplatz Schema 16](#_Toc388449902)

[Abbildung 5: Mein Arbeitsplatz Bild 16](#_Toc388449903)

[Abbildung 6: Kanban 17](#_Toc388449904)

[Abbildung 7: Fehler Zeitplan 28](#_Toc388449905)

[Abbildung 8: Fehlermeldung nach git push 39](#_Toc388449906)

[Abbildung 9: Erläuterung Git 44](#_Toc388449907)

[Abbildung 10: Phasenmodell 62](#_Toc388449908)

Tabellenverzeichnis

[Tabelle 1: Personenkreis 13](#_Toc388449909)

[Tabelle 2: Projektrollen 14](#_Toc388449910)

[Tabelle 3: Vorkenntnisse 15](#_Toc388449911)

[Tabelle 4: Datensicherung Git 18](#_Toc388449912)

[Tabelle 5: Terminalbefehle git 44](#_Toc388449913)

[Tabelle 6: Ist-Ablauf Plattform aufsetzten 1 45](#_Toc388449914)

[Tabelle 7: Ist-Ablauf Plattform aufsetzten 2 46](#_Toc388449915)

[Tabelle 8: Soll-Ablauf Plattform aufsetzten 46](#_Toc388449916)

[Tabelle 9: Vorgehensziele 47](#_Toc388449917)

[Tabelle 10: Systemziele 47](#_Toc388449918)

[**Tabelle 11: Variantenentscheid** 51](#_Toc388449919)

[Tabelle 12: Physikalische Hardware/Software Macbook pro 56](#_Toc388449920)

[Tabelle 13: Physikalische Hardware/Software Server 57](#_Toc388449921)

[Tabelle 14: Erläuterung der Inhaltstype 59](#_Toc388449922)

[Tabelle 15: Erläuterung der Inhaltstype 72](#_Toc388449923)

[Tabelle 16: Erläuterung Aufbau Inhaltsseite 73](#_Toc388449924)

[Tabelle 17: Testprotokoll 73](#_Toc388449925)

Management Summary

Zusammenfassung der IPA-Resultate; max. einer A4 Seite (Management Summary).

max. zwei A4 Seiten Abschlussbericht (zweitletzte Seite Teil 1).

Ausgangssituation

Umsetzung

Ergebnis

Teil 1: Ablauf und Umfeld

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

Autor: Michel Weingart



Abbildung : Logo 4teamwork AG

# Aufgabenstellung

Für mehrere webbasierte Standardlösungen soll regelmässig (e.g. wöchentlich) und automatisiert öffentliche Demoinstallationen mit Beispielinhalten im Internet verfügbar gemacht werden.

## Ausgangslage

4teamwork entwickelt mehrere webbasierte Standardlösungen: teamraum Collab, teamraum Intranet, teamraum Web. Diese Lösungen basieren auf dem Content Management System Plone und werden in einem geordneten Releaseprozess laufend erneuert und erweitert. Neue Funktionen werden verfügbar gemacht, von den Benutzern gemeldete Fehler und Unzulänglichkeiten werden korrigiert. Es existiert eine Demoplattform demo.teamraum.ch, über die sich interessierte Personen einloggen und die Lösung unverbindlich testen können. Die Demoplattform wird von 4teamwork nur unregelmässig aktualisiert und ist daher meistens nicht aktuell. Sie enthält kaum Beispielinhalte.  
  
Entwicklungs- und Testinstallationen der Standardlösungen werden von den Entwicklern von 4teamwork zwar immer wieder neu erstellt, allerdings sind diese nur lokal verfügbar, und nicht von aussen zugänglich. Diese Installationen dienen vorwiegend für automatisierte Softwaretests, ohne Benutzerinteraktionen. Aus diesen Gründen werden neue Releases von teamraum unter Umständen zu wenig von Menschen getestet (Human Testing), bevor sie freigegeben werden.

## Auftragsformulierung

Es werden die folgenden Resultate erwartet:

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

## Mittel und Methoden

Die Erläuterungen zu den wichtigsten Systemen befindet sich in der Phase Initialisierung des Projekts.

* 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 ein Word2010 verfasst.
* Anwendungswissen von Plone und teamraum ist erforderlich.
* Als Programmiersprachen kommen (in geringem Umfang) Python und Python-Skripte zum Einsatz.

Das Conten Management System „Plone“, welches für die Webbasierten Plattformen verwendet wird basiert auf Python.

* Als Programmierungsumgebung wird der Texteditor Textmate verwendet.

Die Entscheidung der Programmierungsumgebung fiel aus folgenden Gründen auf den erwähnten Texteditor:

* Syntax-Highlighting für Python (und andere Programmiersprachen) ist vorhanden
* Die Software ist Freeware
* Es besteht eine grosse Entwicklergemeinschaft zu dieser Software
* Erweiterungen sind einfach zu installieren
* Gute Kenntnisse von X-HTML und XML sind erforderlich.
* Die Demoinstallationen können lokal auf Mac Notebooks oder auf einem Linux-Server von 4teamwork/in einer Cloud-Umgebung gemacht werden.
* 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.

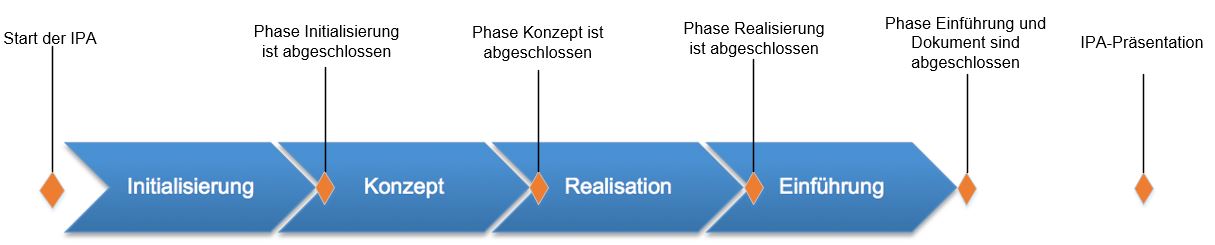


Abbildung : Hermes 5 IPA

## Projektmanagementplan

**Beteiligter Personenkreis:**

|  |  |
| --- | --- |
| Name | Kürzel |
| Michel Weingart | MW |
| Manuel Schaffner | MS |
| Verena Meinung | VM |
| Dr. Pascal Habegger | PH |
| Mathias Leimgruber | ML |
| Dr. Thomas Buchberger | TB |

Tabelle : Personenkreis

**Aufgraggeber**

4teamwork AG

Abbildung : Organigramm Projekt

**Projektausschuss**

Dr. Pascal Habegger

**Qualität- & Sicherheitsmanager**

Valid-Experte: Yvan Kolly   
Hauptexperte: Manuel J. Schaffner  
2. Expertin: Verena Meinung

**Projektleiter:**

Michel Weingart

**Entwickler**

Michel Weingart

**Product Manager Teamraum**

Mathias Leimgruber

**Tester**

Dr. Thomas Buchberger

**Fachvorgesetzter**

Dr. Pascal Habegger

## Projektrollen

|  |  |
| --- | --- |
| 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 des Teams. |
| Entwickler | Er ist zuständig für das technische Design und die Implementierung der Software. Er ist somit für die technische Umsetzung der Anforderung zuständig. |
| Product Manager Teamraum | Er dient zur Unterstützung des Auszubildenden in Sachen technischer Umsetzung und allgemeinen Fragen. Er ist ein Spezialist betreffend der Produkte von teamraum |
| Tester | Er testet alle Komponenten gemäss den definierten Testfällen und erfasst die Ereignisse in die dazugehörigen Testprotokolle. |

Tabelle : Projektrollen

# 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 | - | - |
| Plone Grundwissen | Supportarbeiten | 1 Jahr |
| Git | Kleinere Entwicklungen | 0.5 Jahr |
| Office | Tägliche Arbeiten | 7 Jahre |
| HERMES 5 | Bei kleineren Projekten in der Schule. | 0.5 Jahre |

Tabelle : Vorkenntnisse

# 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 vorbefüllt werden können.
* Besprechung mit einem Entwickler betreffend Programmiersprache Python.
* Programmierung von Python mit Hilfe von einem Entwickler und alleine.
* Einrichtung eines Arbeitsplatzes für die IPA.
* Erstellen eines Kanban-Plakats.
* Dokumente von durchgeführten IPAs durchgelesen.

# 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 pkorg hochgeladen werden
* Die Dokumentation ist einseitig auszudrucken.
* Eine ausgedruckte Version, mit Deckblatt Blau, ist dem Hauptexperten (A-Post) an die private Adresse zu versenden.
* Eine ausgedruckte Version, mit Deckblatt Blau, ist der Zweitexpertin (A-Post) an die geschäftliche Adresse zu versenden.
* Eine ausgedruckte Version, mit Deckblatt Gelb, ist dem Fachvorgesetzten abzugeben.

Die definierten Termine (Besuchstage und Präsentation) müssen eingehalten werden.

# Firmenstandards

Da bei der Firma 4teamwork AG keine Standards betreffend Vorlagen existieren, benutze ich die Dokumentvorlage welche von Pkorg zur Verfügung gestellt ist. Für die Programmierung lehne ich mich an die Geltungsrichtlinien von PEP 8, welche unter <http://legacy.python.org/dev/peps/pep-0008/> aufgeführt sind und bei 4teamwork auch eingesetzt werden.

# Organisation der IPA

Um die IPA ungestört durchführen zu können habe ich meinen Arbeitsplatz, mit dem nötigen Zubehör, vom Supportzimmer in ein meist ungenutztes Zimmer verlegt.



Abbildung : Mein Arbeitsplatz Schema

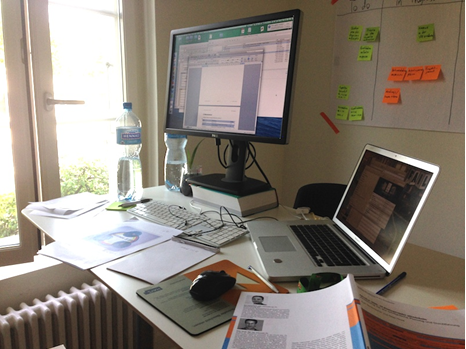


Abbildung : Mein Arbeitsplatz Bild

Zusätzlich zum Wechsel des Arbeitsplatzes habe ich mich entschlossen, das Vorgehensmodell „Kanban“ einzusetzen. Dies 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 mir zur vereinfachten Aufgabenteilung und schnelleren Erkennung von Problemen.



Abbildung : Kanban

## Datensicherung der IPA

Die Datensicherung und Dokumentablage erfolgt auf zwei Arten.

1. Mittels privatem Git Repository

Eine automatische Versionierung ist 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 verfasst.

|  |  |  |
| --- | --- | --- |
| Name Repository | Pfad | Beschreibung |
| IPA\_michel.weingart | https://github.com/4teamwork/IPA\_michel.weingart | Ablage von:  - Dokumenten  - Quellcode  - Bilder/Grafiken |

Tabelle : Datensicherung Git

1. Mittels externer Festplatte

Ein Backup wird jeden Abend auf eine externe Festplatte gespeichert.

# Meilensteine

In der IPA sind folgende fünf 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 | Phase Realisierung ist abgeschlossen und Phase Einführung freigegeben |  |
| 26.05.2014 | Phase Einführung ist abgeschlossen und Dokument abgeschlossen. |  |

Tabelle 4: Meilensteine

# Zeitplan

Planung und Umsetzung in Form von A3 Querseiten (entsprechend falten)

Individueller Zeitplan (nach Projektmethode).¨

# Arbeitsjournal

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

## Erster Tag: Montag, 05.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Kickoff-Meeting*  Das Kickoff-Meeting habe ich mit dem Product Manager Teamraum, Mathias Leimgruber durchgeführt. Das Projekt kann beginnen. | MW/ML | 0.5 | 0.5 |
| *Git Repository*  Das Git Repository habe ich unter „ [https://github.com/4teamwork“ unter dem Namen „IPA\_michel.weingart.git](%20https://github.com/4teamwork)“ erstellt um die Dokumentversionen und sonstige Daten hochladen zu können. | MW | 0.5 | 0.5 |
| *Dokumentation eröffnen*  Ich habe die Dokumentation eröffnet. | MW | 0.5 | 0.5 |
| *Standardeintragungen in der Dokumentation*  Ich habe die ersten kleinen Eintragungen in der Dokumentation vorgenommen wie Header / Footer... | MW | 0.5 | 0.5 |
| *Zeitplan erstellen*  Den Zeitplan habe ich erstellt und die Soll-Zeiten eingetragen. | MW | 2.0 | 2.5 |
| *Aufgabenstellung erfassen*  Die Aufgabenstellung zum Projekt habe ich erfasst. | MW | 1.0 | 1.5 |
| *Vorkenntnisse / Vorarbeiten / Firmenstandards / Organisaiton des Projekts erfassen*  Elemente der Dokumente habe ich eingepflegt. | MW | 1.0 | 1.0 |
| *Studie Ist / Soll*  Eine kurze Reflektion betreffend des Ist und Soll -Zustandes habe ich gemacht. | MW | 0.5 | 0.5 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 8.5 |
| Tages Ablauf |  |  |  |
| Das Projekt konnte ich nach dem Kickoff-Meeting erfolgreich starten. Nach kleineren Problemen betreffend dem Git Repository, habe ich die Dokumentation eröffnet.  Einige Probleme bereitete mir auch das Erstellen des Zeitplans. Dieser beanspruchte mehr Zeit als eingeplant, da ich an ein Excel auf Windows-Systemen gewöhnt bin und selten auf dem MacBook Pro Excel verwende, bei welchem die Software anders aufgebaut ist. Das schätzen des Zeitaufwands für die verschiedenen Phasen habe ich mir einfacher vorgestellt, hoffe aber dass die Schätzungen realistisch sind. Dafür gestaltete sich das Erfassen der ersten Einträge in der Dokumentation leichter als erwartet, was wiederum einen kleinen Zeitgewinn einbrachte. | | | |
| Hilfestellungen |  |  |  |
| Ich habe anfangs 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 Daten. | | | |
| 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 | | | |

## Zweiter Tag: Dienstag, 06.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Studie Ist / Soll*  Der aktuelle und zu erwartende Zustand habe ich aufgezeigt und den Prozess abgebildet. | MW | 1.5 | 1.0 |
| *Vorgehensziele / Systemziele / Anforderungen*  Ich habe mehrere Elemente der Dokumentation hinzugefügt. | MW | 1.5 | 1.5 |
| *Variantenentscheid*  Die bessere Variante konnte ich anhand des Variantenentscheids eruieren. | MW | 1.5 | 1.5 |
| *Informationssicherheit und Datenschutz / Lösungen suchen*  Ich habe weitere Elemente der Dokumentation hinzugefügt. | MW | 1.5 | 1.5 |
| *Freigabe Konzept*  Die Phase Konzept wurde freigegeben. | MW/PH | 0.5 | 0.5 |
| *Eintrag Arbeitsjournal*  Ich habe das Arbeitsjournal nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 7.0 |
| Tages Ablauf |  |  |  |
| Der Ist / Soll Vergleich konnte ich gut abschliessen und grafisch darstellen.  Die bessere Variante hat sich deutlicher herauskristallisiert als ich dies angedacht habe, 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 Einfüllen der Vorgehensziele / Systemziele und 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. Mit 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 Zeitgewinnes bei der Studie zum Ist / Soll Vergleich habe ich dort folgenden Punkt zur Verbesserung: Die Aufteilung der Arbeiten über mehrere Tage habe ich nicht als förderlich empfunden und werde dies in Zukunft vermeiden. Die Studie habe ich am Montag angefangen und am Dienstag beendet. Der Unterbruch verlangte, dass ich mich zuerst wieder in das Thema einarbeiten musste. | | | |
| Nächste Schritte |  |  |  |
| Ich werde mich auf den Erstbesuch des Hauptexperten vorbereiten und Fragen bezüglich der IPA aufschreiben um diese ihm stellen zu können.  Weitere Aufgaben welche ich für morgen geplant habe:   * Situationsanalyse * Allgemeine Anforderungen * Systemarchitektur ermitteln * Zugriffskonzept erstellen * Erarbeitung des Betriebskonzepts | | | |

## Dritter Tag: Mittwoch, 07.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Situationsanalyse*  Die Situationsanalyse betreffend des Konzepts habe ich eingepflegt. | MW | 1.5 | 1.5 |
| *Allgemeine Anforderungen*  Die Anforderungen, damit das Skript lauffähig ist, konnte ich definieren. | MW | 0.5 | 0.5 |
| *Systemarchitektur ermitteln*  Die Systemarchitektur vom Server Poseidon habe ich ermittelt und aufgezeigt. | MW | 1.0 | 1.0 |
| *Zugriffskonzept erstellen*  Die Erarbeitung habe ich begonnen, jedoch noch nicht abgeschlossen. | MW | 1.5 | 1.5 |
| *Erarbeitung des Betriebskonzepts*  Das Betriebskonzept habe ich fertiggestellt. | MW | 1.0 | 1.0 |
| *Expertenbesuch*  Der Expertenbesuch hat stattgefunden. Details dazu, können dem Projektjournal entnommen werden. | MW/MS/VM/ PH | 1.0 | 2.0 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 8.5 |
| Tages Ablauf |  |  |  |
| Die Erarbeitung des Zugriffskonzeptes konnte ich nicht fertigstellen, da ich noch einige Fragen betreffend der geplanten Datenstruktur (Beispielinhalt auf der Plattform) meinem Fachvorgesetzten Pascal Habegger stellen musste. Dieser war aber am Morgen in einer externen Besprechung. Um die Zeit sinnvoll nutzen zu können habe ich deshalb die Erarbeitung des Betriebskonzepts auf den Vormittag verschoben.  Für die Erarbeitung des Zugriffkonzepts habe ich 1.5 Stunde aufgewendet, konnte dies aber noch nicht ganz abschliessen und werde dies morgen erledigen und als Punkt im Arbeitsjournal vermerken.  Heute erfolgte der Besuch des Hauptexperten Manuel J. Schaffner und der Zweitexpertin Verena Meinung. Weitere Informationen betreffend dieser Sitzung sind im Projektjournal festgehalten. | | | |
| Hilfestellungen |  |  |  |
| Heute Morgen habe ich noch die Rückmeldung des Experten betreffend der gestrigen Fragestellung (Anpassung Zeitplan) erhalten. Er hat mich darauf hingewiesen, dass die Soll-Zeiten angepasst werden können, hingegen die Ist-Zeiten so belassen werden sollten. | | | |
| Ein interner Projektleiter, Murat Tokmak hat mir, betreffend des Aufbaus 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 Branches von Branches erstellt oder ist dies mit einzelnen Commits umzusetzen | | | |
| Reflexion |  |  |  |
| Das Zusammentragen der gewünschten Informationen hat gut funktioniert. Ich hätte mehr Zeit einplanen müssen für die Überarbeitung der Dokumentation und die Überarbeitung des Zeitplans. Ich schätze, dass dies noch ein 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 | | | |

## Vierter Tag: Donnerstag, 08.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Anpassung des Zeitplan*  Den Zeitplan habe teilweise überarbeiten. | MW | 0.0 | 1.0 |
| *Erarbeitung des Zugriffskonzept*  Die Erarbeitung des Zugriffkonzepts habe ich abgeschlossen | MW | 0.0 | 1.0 |
| *Erarbeitung des Testkonzepts*  Das Testkonzept mit den verschiedenen Testfällen dazu habe ich erstellt. | MW | 2.0 | 1.5 |
| *Erarbeitung des Einführungskonzepts*  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 |
| *Überarbeitung Dokumentation*  Einige Punkte der Dokumentation habe ich überarbeitet. | MW | 0.0 | 1.0 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.5 | 1.0 |
| Total: |  | 7.5 | 7.0 |
| Tages Ablauf |  |  |  |
| Das Testkonzept ist mir sehr gut gelungen und ich konnte eine halbe Stunden 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.  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. | | | |
| Hilfestellungen |  |  |  |
| Heute habe ich keine Hilfestellung benötigt. | | | |
| Reflexion |  |  |  |
| Heute konnte ich produktiv arbeiten, da ich mich unter Anderem über den Mittag kurz sportlich betätigt habe. Ich konnte am Nachmittag die Produktivität auf einem hohen Niveau halten. | | | |
| Nächste Schritte |  |  |  |
| Der nächste Tag, an welchem ich für die IPA arbeiten kann ist erst am Montag dem 19.05.2014, da ich während einer Woche die Computerschule Bern CSBE besuche. Ich hoffe, dass ich gut an das Projekt anschliessen kann nach diesem Unterbruch. Die nächsten Schritte welche ich geplant habe sind:   * Freigabe der Phase Realisierung * Bestehende Umgebung verifizieren * Konfiguration festlegen * Installationsskript erstellen (noch nicht fertig) | | | |

## Fünfter Tag: Montag, 19.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Anpassung des Zeitplans*  Die letzten Anpassung, welche am Expertenmeeting besprochen wurden, habe ich vorgenommen. | MW | 0.0 | 0.5 |
| *Freigabe der Phase Realisierung*  Beschreibung | MW | 3.0 | 0.5 |
| *Verifizierung der bestehenden Umgebung*  Beschreibung | MW | 1.0 | 1.0 |
| *Festlegen der Konfiguration für die Realisierung*  Beschreibung | MW | 1.0 | 1.0 |
| *Überarbeitung der Dokumentation*  Mehrere Punkte der Dokumentation habe ich überarbeitet. | MW | 0.0 | 2.0 |
| *Erarbeitung des Installationsskripts*  Ich habe die Erarbeitung begonnen. | MW | 1.5 | 1.5 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.0 | 1.0 |
| **Total** |  | **7.5** | **7.5** |
| Tages Ablauf |  |  |  |
| Ein kleiner aber wirkungsvoller Fehler in meinem Zeitplan hat dazu geführt, dass am Morgen nur die Freigabe der Realisierung durchgeführt werden musste.SSD:Users:mischu:Desktop:Bildschirmfoto 2014-05-19 um 10.55.37.png  Abbildung : 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 konsultieren von verschiedene Google-Quellen war schnell klar, dass auf der ersten Zeile den Zeichensatz deklariert werden muss. | | | |
| Hilfestellungen |  |  |  |
| Betreffend Optionen dem Skript hinzufügen habe ich beim Entwickler, Jonas Baumann, eine kleine Hilfestellung erhalten. Morgen werde ich versuchen, dass eine Option dem Skript mitgegeben werden kann. | | | |
| Reflexion |  |  |  |
| Bei einem nächsten Projekt, muss ich den Zeitplan besser kontrollieren. Der Fehler war recht offensichtlich und hätte vermieden werden können. Das Gute des Fehlers ist, dass, dass ich viele Punkte der Dokumentation überarbeiten konnte. Der Start in des Skripts ist mir geglückt und ich konnte heute viel erarbeiten. | | | |
| Nächste Schritte |  |  |  |
| Morgen werde ich mich ausschliesslich mit dem Skript befassen und versuchen dies wenn möglich fertigzustellen. | | | |

## Sechster Tag: Dienstag, 20.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Erarbeitung des Installationsskripts „demo-teamraum.py“*  Das Installationsskript konnte ich fertigstellen und die Hauptfunktionen sind gewährleistet. Falls noch Zeit bleibt, werde ich das Skript noch verbessern. | MW | 6.5 | 6.0 |
| *Überarbeiten der Systemstruktur und des Testkonzepts*  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 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 8.5 |
| Tages Ablauf |  |  |  |
| 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 Verständnis, noch einige Punkte hinzufügen. | | | |
| Hilfestellungen |  |  |  |
| Trotz gestriger kurzer Hilfestellung betreffend Optionen dem Skript hinzuzufügen, musste ich nochmals beim Entwickler, Jonas Baumann nachfragen. Ich wusste nicht, wie ich dem OptionParser einen Standardwert mitgeben kann. | | | |
| Reflexion |  |  |  |
| Es war sehr hilfreich dass ich mich, als Vorarbeit der IPA, mit Entwicklern und alleine mit der Programmiersprache Python befasst habe. Dies hat der Arbeitsaufwand am Skript deutlich reduziert. | | | |
| Nächste Schritte |  |  |  |
| Morgen werde ich ein Profil definieren im Branch des Pakets „eGov“. Runskript definieren, wobei einige Konfigurationen diesbezüglich vom Product Manager Teamraum, Mathias Leimgruber durchgeführt werden müssen.  Weitere Schritte welche ich geplant habe sind:   * Vorbereitung Expertenbesuch2 * Lösung antesten | | | |

## Siebter Tag: Mittwoch, 21.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Profil definieren im Branch des Pakets „eGov“*  Profil und Inhalte konnte ich definieren. | MW | 2.5 | 2.5 |
| *Runskript im Branch des Pakets buildout-base erstellen*  Das Runskript habe ich mit dem Fachspezialisten Technik, Mathias Leimgruber erarbeitet. | MW | 3.0 | 3.0 |
| *Lösung testen*  Die Lösung konnte ich aus Zeitmangel nicht testen. | MW | 0.5 | 0.0 |
| *Vorbereitung Expertenbesuch2*  Ich habe Frage aufgeschrieben, welche ich dem Experte morgen stellen werden. | MW | 0.5 | 0.5 |
| *Eintrag Arbeitsjournal*  Das Arbeitsjournal habe ich nachgeführt. | MW | 1.0 | 1.0 |
| Total: |  | 7.5 | 7.5 |
| Tages Ablauf |  |  |  |
| 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 eingepflegt werden können. Deshalb werde ich dies morgen noch mit dem Fachspezialisten Technik, Mathias Leimgruber erarbeiten. Das Runskript habe ich auch mit ihm erarbeitet und implementiert, da dies technisch zu anspruchsvoll war. Das erste Testen konnte ich aus Zeitmangel nicht durchführen und werde dies weglassen und mich bei der Einführungsphase auf das Testen konzentrieren. | | | |
| Hilfestellungen |  |  |  |
| Der Fachspezialist Technik, Mathias Leimgruber hat mich beim Erarbeiten des Runskripts geführt. | | | |
| Reflexion |  |  |  |
| Ich habe zu wenig Zeit für die Realisierung eingeplant, hoffe aber, dass ich diese Zeit wieder einholen kann. | | | |
| Nächste Schritte |  |  |  |
| Für morgen sind folgende Sachen geplant:   * Fertigstellen des Profils * Programmablaufplan erstellen * Anleitung erstellen * Freigabe der Phase Einführung * Expertenbesuch2 | | | |

## Achter Tag: Donnerstag, 22.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Neunter Tag: Montag, 26.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Zehnter Tag: Dienstag, 27.05.2014

|  |  |  |  |
| --- | --- | --- | --- |
| Tätigkeiten | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
| *Überschrift*  Beschreibung |  |  |  |
| *Überschrift*  Beschreibung |  |  |  |
| Total: |  |  |  |
| Tages Ablauf |  |  |  |
| Beschreibung von Erfolgen und Misserfolgen (Problemen) | | | |
| Hilfestellungen |  |  |  |
| Wer hat in welcher Form Hilfestellung erbracht | | | |
|  | | | |
| Reflexion |  |  |  |
| Gedanken über das Vorgehen  Was ist gut was würde ich besser machen | | | |
| Nächste Schritte |  |  |  |
| Wie weiter, nächste Schritte... | | | |

## Arbeitszeit total

|  |  |  |  |
| --- | --- | --- | --- |
| Totaler Zeitaufwand | Person | Aufwand  geplant (Std) | Aufwand  effektiv (Std) |
|  |  |  |  |
| Reflexion |  |  |  |
| Kurze Begründung falls unterschritten oder überzogen (Richtwert 80 Std +/-10%) | | | |

# Projektjournal

Im Projektjournal sind Informationen chronologisch gesammelt, welche im Verlauf

Des Projekts eine Rolle spielen. Besprechungsprotokolle mit Entscheiden und Abmachungen

sind besonders wichtig. Das Projektjournal wird laufend abgefüllt.

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Meilenstein |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Product Manager Teamraum, Mathias Leimgruber |
| Themen: | Start der IPA |
| Ergebnis / Entscheide: | IPA wurde nach dem Kickoff-Meeting gestartet |

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Telefonbesprechung |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Hauptexperte, Manuel J. Schaffner * Fachvorgesetzter, Pascal Habegger |
| Themen: | Telefonische Besprechung betreffend Meetings (Expertenbesuch1, Expertenbesuch2, Präsentation IPA) |
| Ergebnis / Entscheide: | Meetings festgelegt auf: 07.05.2014 / 22.05.2014 und 10.06.2014 |

|  |  |
| --- | --- |
| Datum: | 05.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Product Manager Teamraum, Mathias Leimgruber |
| Themen: | Analyse / Auswahl für Konzepterarbeitung |
| Ergebnis / Entscheide: | Analyse für die verschiedenen Konzeptthemen. Die Entscheidung ist auf folgende Konzeptthemen gefallen:  - Systemanforderung  - Systemarchitektur  - Zugriffskonzept  - Betriebskonzept  - Testkonzept  - Einführungskonzept |

|  |  |
| --- | --- |
| Datum: | 06.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Product Manager Teamraum, Mathias Leimgruber |
| Themen: | Analyse / Auswahl Kriterien für Variantenentscheid |
| Ergebnis / Entscheide: | Eine Auswahl von verschiedenen Kriterien für den Variantenentscheid wurde getroffen. Bei folgenden Kriterien macht es Sinn, diese für den Variantenentscheid einzusetzen:   * Usability für Entwickler * Usability für weitere Benutzer * Flexibilität * Unabhängigkeit * Aufwand für: * Entwicklung * Einführung * Testing |

|  |  |
| --- | --- |
| Datum: | 06.05.2014 / 07.05.2014 |
| Typ: | Mailing |
| Beteiligte Personen: | - Projektleiter, Michel Weingart  - Hauptexperte, Manuel J. Schaffner |
| Themen: | Fragen betreffend Anpassung Zeitplan |
| Ergebnis / Entscheide: | Rückmeldung erhalten inwiefern der Zeitplan angepasst werden darf.  Soll-Zeiten werden belassen, hingegen bei den Ist-Zeiten kann es Verschiebungen geben. |

|  |  |
| --- | --- |
| Datum: | 07.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Interner Projektleiter, Murat Tokmak |
| Themen: | Fragen betreffend Betriebskonzept |
| Ergebnis / Entscheide: | Der Aufbau des Betriebskonzepts wurde erläutert und ist klar. |

|  |  |
| --- | --- |
| Datum: | 07.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Fachvorgesetzter, Pascal Habegger |
| Themen: | Fragen betreffend Dateistruktur und Berechtigungen |
| Ergebnis / Entscheide: | Die vorgeschlagene Dateistruktur muss noch überarbeitet werden. Die Struktur der Berechtigung wurde angenommen. |

|  |  |
| --- | --- |
| Datum: | 07.05.2014 |
| Typ: | Meeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Hauptexperte, Manuel J. Schaffner * Fachvorgesetzter, Pascal Habegger |
| Thema: | * Vorstellungsrunde * Aufgabenstellung besprechen * Ablauf der IPA besprechen * Besprechen der Dokumentation * Fragen beantworten: |
| Ergebnis / Entscheide: | Redundanzen   * Redundanzen sollten in der Dokumentation vermieden werden stattdessen soll mit Verweisen gearbeitet werden.   Roter Faden   * Ein roter Faden soll sich durch die Dokumentation ziehen.   Zeitplan   * Die Programmierung zur Realisierung muss aufgeteilt werden. * Ist / Soll muss bei den Phasen klarer dargestellt sein. * Unterbruch im Zeitplan soll besser ersichtlich sein. * Meilensteine müssen im Zeitplan angegeben werden.   Allgemeine Anpassungen / Punkte   * Die Ich-Form wird ausschliesslich im Arbeitsjournal verwendet * Die Versionskontrolle soll kürzer ausgelegt sein. * Der Projektmanagementplan muss angepasst werden. * Bei den Projektrollen müssen die Rollen beschrieben werden. * Meilensteine sollen in die Umgebung Zeitplan verschoben werden * Beteiligte Personen sollen zu den Personen verschoben werden * HERMES5 Grafik gehört zu den Mitteln und Methoden * Beschriftung beim Ist / Soll –Vergleiche muss klarer abgegrenzt sein * Systemziele sollen nummeriert sein. * Für die Wirtschaftlichkeit ist eine Grafik wünschenswert(Stunden Kosten…) * Eine Situationsanalyse soll im Konzept nicht vorhanden sein * Für die Systemarchitektur wird eine Grafik gewünscht * Plone soll erklärt werden. * Bei Hilfestellungen soll die Person namentlich erwähnt werden. * Ein Stellvertretender Fachvorgesetzter ist nicht nötig. * Im Anhang wird Code aufgeführt, welcher ich selber erarbeitet habe. * Richtlinien betreffend Coding sind bie 4teamwork vorhanden (Pep8). * Implementationsschritte des Skripts auf den Server sind gewünscht. * Testprotokoll und Testplan erarbeiten für das Testing. * Bei der Realisierung kann auch mit Screenshots gearbeitet werden um etwas darzustellen. * Eventuell einen Link in der Dokumentation mitgeben auf eine existierende Umgebung, damit sich die Experten ein Bild davon machen können. |

|  |  |
| --- | --- |
| Datum: | 08.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Wein * Fachvorgesetzter, Pascal Habegger |
| Themen: | Diskussion betreffend Freigabekonzept |
| Ergebnis / Entscheide: | 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. |

|  |  |
| --- | --- |
| Datum: | 20.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Entwickler, Jonas Baumann |
| Themen: | OptionParser |
| Ergebnis / Entscheide: | OptionParser soll verwendet werden um dem Skript „demo-teamraum.py“ Argumente und Optionen mitgeben zu können. |

|  |  |
| --- | --- |
| Datum: | 21.05.2014 |
| Typ: | Kurzmeeting |
| Beteiligte Personen: | * Projektleiter, Michel Wein * Product Manager Teamraum, Mathias Leimgruber |
| Themen: | Nach einem git push auf dem Paket eGov wird folgende Fehlermeldung ausgegeben:  SSD:Users:mischu:Desktop:IPA_michel.weingart:Bilder:Fehlermeldung git push.png  Abbildung : Fehlermeldung nach git push  Da die Bedeutung der Ausgabe nicht ganz klar war, wurde dies nachgefragt. |
| Ergebnis / Entscheide: | 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 neuen Master definiert worden. |

|  |  |
| --- | --- |
| Datum: | 22.05.2014 |
| Typ: | Meeting |
| Beteiligte Personen: | * Projektleiter, Michel Weingart * Hauptexperte, Manuel J. Schaffner * Fachvorgesetzter, Pascal Habegger |
| Themen: | * Stand der Dokumentation * Allgemeine Verbesserungen in der Dokumentation * Verbesserungen Skirpt * Besprechung Präsentation * Besprechung der kommenden Termine |
| Ergebnis / Entscheide: | Zeitplan   * Phase Einführung findet nicht statt, Punkte werden in Phase Realisierung verschoben   Dokumentation   * Grafik für Plone -> erläutern wie dies mit den anderen Systemen zusammenhängt * Abbildungen alle beschriften * Eingefügte Bilder besser beschreiben (was ist das Ticketingsystem...) * Für Tabellen soll eine andere Kopffarbe genommen werden * Was ist ein Profil? -> Erklären * Einleitungssatz bei der Auflistung der Konfiguration * Link auf Plattform oder hilfreiche Dokumente (z.B. für weitere Informationen siehe...) * Teil 2 der Dokumentation: Ich muss mir bewusst sein, dass die Beschreibungen für aussenstehende Personen verständlich sein muss. Vom groben Detail ins feine.   Testprotokolle   * Effektives Ergebnis gehört insTestprotokoll * Testnummern ersetzen mit aussagekräftigem Stichwort * Mit Verweisen arbeiten   Skripts   * Warum json? * Wurden schon andere Profile definiert? * Runskript umbenennen und sinnvoller Name vergeben   Präsentation   * Teil 1 dauert 15 – 20 Minuten und ist in Schriftdeut abzuhalten. Schwerpunkte setzen auf die spannendsten Themen mit den Präsentationsmittel welche ich benutzen will. * Teil 2 dauert 10 Minuten und beinhaltet eine Demo * Teil 3 ist das Fachgespräch   Websummary   * Das Websummary ist 24 Stunden vor der Präsentation auf pkorg hochzuladen * Das Websummary muss ausgedruckt bei der Präsentation vorliegen. |

# Abschlussbericht

Wurden die geforderten Ziele erreicht?

## Vergleich Ist/Soll

Ist die Umsetzung wie geplant oder gab es Differenzen?

## Mittelbedarf

Welche Mittel wurden gebraucht oder mussten noch beschafft werden?

## Realisierungsbericht

Gab es Probleme während der Realisation? Ungeplante Sachen zum Vorschein oder Ergänzungen?

## Testbericht

Wie verliefen die Tests (Erfolgreich / weniger Erfolgreich)? Gibt es Fehler, die schwerwiegend sind oder solche die später korrigiert werden können/müssen?

## Fazit zum Projekt

Wie ist das Projekt verlaufen (Objektive) Meinung

Hatte das Projekt Stolpersteine, welche etc…

## Persönliches Fazit

Feedback und Reflexion des Lernenden rund um die IPA.

Was war gut, was weniger? Was habe ich gelernt und was würde ich ev. das nächste Mal anders machen? Ausführlich formulieren.

# 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 |
|  | Michel Weingart |  |
|  | Pascal Habegger |  |

Teil 2: Projektdokumentation

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

Autor: Michel Weingart

# Initialisierung

## Situationsanalyse

Um ein Basisverständnis zu diesem Projekt zu schaffen, wird zuerst aufgezeigt, welche wichtigsten Systeme/ Entwicklungstools bei 4teamwork zum Einsatz kommen.

### Plone

Plone ist im eigentlichen Sinne ein Content Management System welches vorwiegend für den Einsatz in Organisationen und Unternehmen ausgerichtet ist. Folgende Eigenschaften zeichnen dieses CMS unter Anderem aus:

* Basiert auf Zope, einem Web-Framework
* Ist quelloffen (Open-Source)
* Wurde in der objektorientierten Programmiersprache Python entwickelt
* Unterstützt rund 50 Sprachen
* Ist in einer grossen und professionellen Community integriert

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

### Teamraum

Die modulare [teamraum Produktelinie](http://www.4teamwork.ch/loesungen/teamraum" \t "_self) 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.

### Github / Git.4teamwork

Github und Git.4teamwork sind webbasierte Plattformen und werden bei 4teamwork verwendet um Entwicklungsprojekte ablegen und versionieren zu können. Auch wird ein Review-Prozess durchgeführt um Fehler einzugrenzen. Beide Systeme werden über ein Terminal angesprochen. Die wichtigsten Befehle in Bezug auf das Projekt, sind in der nachfolgenden Tabelle eingetragen.

|  |  |
| --- | --- |
| Terminalbefehl | Auswirkung |
| git clone git@git.4teamwork.ch:egov/buildout-base.git demo-teamraum | Das existierende Git Repository wird in den Ordner „demo-teamraum“ geklont / Das Paket ist somit ausgecheckt. |
| git checkout –b ipa-demo-teamraum | Der Branch ipa-demo eines Pakets  wird innerhalb eines heruntergeladenen Paktes erstellt. Automatisch wird nach diesem Befehl auf den Branch gewechselt. |
| git add . | Alle Änderungen, welche innerhalb des heruntergeladenen Pakets 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 : Terminalbefehle git

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



Abbildung : Erläuterung Git

## 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 einem geordneten Releaseprozess eingebettet.

Diese werden genutzt um Verbesserungen in die Produkte einfliessen zu lassen.

Um die Durchführung dieses Releaseprozesses einhalten zu können, sind mehrere Teilschritte definiert.

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

Pro Jahr werden viele kleinere Paketreleases und 4-6 Major-Releases für die Produkte von 4teamwork eingespielt.



Ein neuer Release wird vom zuständigen Entwickler zwar lokal getestet, doch ist dieser Release für externe Personen nicht zugänglich. Diese können aber Tests auf der Demoplattform 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 Beispielinhalte vordefiniert. Dies kann dazu führen, dass Releases freigegeben werden, ohne dass diese genügend getestet wurden.

**Soll:**

Das Ziel des Projekts ist, die automatische Bereitstellung von Demoplattformen, auf welchen neue Releases getestet werden können. Benutzer und Inhalte werden zudem automatisch eingepflegt. Zudem wäre die Möglichkeit zu einer einfachen Implementation der Lösung auf den Servern von 4teamwork wünschenswert.



## Zeitrahmen / Budget

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

## 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. Wird die Qualität der Produkte erhöht, festigt dies die Marktposition der Produkte von 4teamwork.

## 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.

## Anforderungen

In der untenstehenden Tabelle sind wichtigsten Anforderungen an das System erläutert.

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

Tabelle : Systemziele

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

|  |  |
| --- | --- |
| Nr. | Nicht funktionale Anforderungen |
| 1 | Einfache Bedienung ist gewährleistet |
| 2 | Die Lösung kann ohne grossen Aufwand erweitert werden |
| 3 | Die Lösung kann ohne grossen Aufwand portiert werden |

Tabelle : Nicht funktionale Anforderungen

## Anwendungsbereich

Die Lösung wird einerseits von Entwicklern von 4teamwork genutzt um einen Release auf einer Demoplattform aufzusetzen, anderseits von externen Benutzern welche sich auf der Demoplattform anmelden um die neusten Features und somit die neusten Releases zu testen.

## Wirtschaftlichkeit

Die Wirtschaftlichkeit des Projekts kann zu diesem Zeitpunkt noch nicht eindeutig erläutert werden. Die Klarheit zur Wirtschaftlichkeit soll per Zeitrechnung nach dem Projekt mit folgenden Vergleichstabellen errechnet werden.

**Vor Projektbeginn**

|  |  |  |
| --- | --- | --- |
|  | Anzahl | Aufwand h |
| Wie viele Tickets wurden zum Release xy nachträglich von externen Benutzern erstellt? |  | Pro erstelltem Ticket |
| Wie viele Tickets wurden zum Release xy nachträglich von Mitarbeitern 4teamwork erstellt? |  |  |
| Auf wie vielen Plattformen wurde ein Fehler, welcher nachträglich behoben werden musste, eingespielt? |  | Pro Plattform |
| Wie viele neue Plattformen wurden initialisiert? |  |  |
| Total |  |  |

**1 Jahr nach Abschluss des Projekts**

|  |  |  |
| --- | --- | --- |
|  | Anzahl | Aufwand h |
| Wie viele Tickets wurden zum Release xy nachträglich von externen Benutzern erstellt? |  | Pro erstelltem Ticket |
| Wie viele Tickets wurden zum Release xy nachträglich von Mitarbeitern 4teamwork erstellt? |  |  |
| Auf wie vielen Plattformen wurde ein Fehler, welcher nachträglich behoben werden musste, eingespielt? |  | Pro Plattform |
| Wie viele neue Plattformen wurden initialisiert? |  |  |
| Total |  |  |

Es ist nicht garantiert dass alle Fehler in den Releases mit dem Einsatz der Lösung gefunden werden aber sicher mehr als bis anhin.

## Risikoanalyse

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

## Varianten

Es bestehen mehrere Möglichkeiten die gewünschte Demoplattform bereitzustellen. Entweder wird diese auf einer Virtuellen Maschine installiert und das Image bei jedem Release neu konfigurieren, oder es wird ein Skript erarbeitet, welches ermöglicht, die Demoplattform ohne Aufwand aufzusetzen. Anhand folgender Beurteilungskriterien, wird eine von den zwei Varianten ausgewählt:

* **Usability:** Wie bedienerfreundliche ist das Aufsetzten der Demoplattform für die Entwickler? Kann die Demoplattform auch von weiteren Personen benutzt werden?
* **Flexibilität:** Welcher Aufwand wird benötigt, um die Variante zu erweitern oder anzupassen?
* **Unabhängigkeit:** Kann das Aufsetzten der Demoplattform unter verschiedenen Betriebssystemen durchgeführt werden?
* **Entwicklungsaufwand:** Welcher Aufwand wird benötigt um die Variante zu entwickeln?

### Variante 1

Auf einer virtuellen Maschine wird ein vorbereiteter Release einer Webanwendung, inklusiv Beispielbenutzer und Beispieldaten, 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 konnte ich zu dieser Variante herausfiltern:

**+ Aufwand:** Der Aufwand um eine Virtuelle Maschine mit Unix-basiertem Betriebssystem aufzusetzen und ein Skript zur Automatisierung zu erstellen ist relativ gering.

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

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

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

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

### Variante 2

Auf einer Linux-Umgebung, werden per Skript die Webanwendung automatisch installiert und Beispielbenutzer und Beispielinhalte erfasst. Folgende Vorteile und Nachteile konnten zu dieser Variante herausgefiltert werden:

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

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

**+ Unabhängigkeit:** Ein Python-Skript kann nur auf Unix-basierten Betriebssystemen ausgeführt werden.

**- Flexibilität:** Ein Skript kann schnell angepasst und erweitert werden. Eine neue Version einzutragen wird von einem Entwickler in kürzester Zeit durchgeführt.

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

## Variantenentscheid



**Tabelle 8: Variantenentscheid**

### Begründung

Aufgrund des Variantenentscheids, wurde die Umsetzung per Variante 2 entschieden. Ein sehr grosser Vorteil der Variante per Skript ist die Kombination der Flexibilität und Usability für die Entwickler. Das Skript könnte auch Eingabeparameter entgegennehmen, welche für die Variante per Virtueller Maschine nicht definiert werden könnten.

## Lösungen suchen

Um Lösungsansätze zu finden wie das Skript funktionieren / aussehen und aufgebaut sein soll, wurde die Kreativitätstechnik „Brainstorming“ gewählt. Für 5 Minuten wurden alle Ideen zu folgender Fragestellung gesammelt:

*Was muss das ideale Skript beinhalten und wie funktioniert dieses ?*

Die wichtigsten Kriterien wurden herausgefiltert und werden für den Lösungsentscheid verwendet.

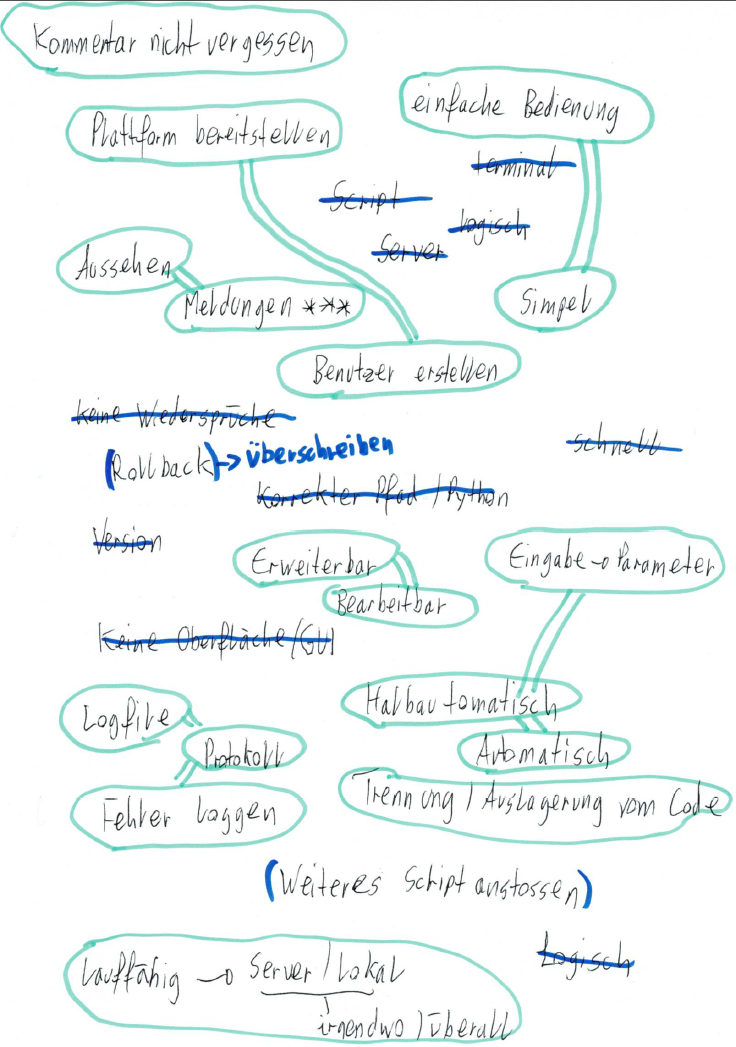


Abbildung : Brainstorming

**Entscheid über den Lösungsvorschlag**

Das Skript kann folgende Elemente beinhalten:

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

## Freigabe Konzept

Die Phase Konzept wird freigegeben:

|  |  |  |
| --- | --- | --- |
| Datum | Name / Auftraggeber | Unterschrift |
| 06.05.2014 | Pascal Habegger |  |

# Konzept

In dieser Phase steht die Erarbeitung der Grundlagen betreffend der Realisierung und Einführung des Projektes im Vordergrund.

## Anforderungen

Um die Lauffähigkeit des Skripts gewährleisten zu können, müssen gewisse Anforderungen erfüllt sein. Diese wurden teilweise dem Brainstorming vom vorgehenden Kapitel entnommen.

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

Tabelle : Anforderungen Skript

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

|  |  |
| --- | --- |
| Nr. | Nicht funktionale Anforderungen |
| 1 | Das Skript soll Eingabeparameter entgegennehmen können. |
| 2 | Das Skript soll Fehler dokumentieren. |
| 3 | Das Skript soll erweiterbar sein. |

Tabelle : Nicht funktionale Anforderungen Skript

## Systemstruktur

Das Skript wird auf einem MacBook pro entwickelt und soll bis zum Projektende auf dieser Umgebung implementiert bleiben. Trotzdem wird die Systemstruktur des Servers Poseidon aufgezeigt, da das Skript nach Abschluss des Projekts auf diesen migriert wird.



Abbildung : Systemstruktur Poseidon

### Physikalische Hardware / Software Server

|  |  |
| --- | --- |
| **Beschreibung** | **Daten** |
| Typ | Server |
| Hostname | poseidon.4teamwork.ch |
| Motherboard | IBM |
| Betriebssystem | CentOS 5.10 |
| CPU | 2.80GHz Intel(R) Xeon(R) |
| Festplattenspeicher disk1 | IBM 175GB |
| Festplattenspeicher disk2 | IBM 145GB |

Tabelle : Hardware / Software Poseidon

## Zugriffskonzept

Bei der Erarbeitung des Zugriffkonzepts geht es darum, die grundsätzlichen Berechtigungen der verschiedenen automatisch erzeugten Benutzer auf der Demoplattform aufzuzeigen. Um dies verständlich darstellen zu können, wird zuerst die Inhaltsstruktur der Demoplattform und anschliessend die Berechtigungen der Benutzer vereinfacht aufgezeigt.

### Inhaltsstruktur

****

Abbildung : Inhaltsstruktur

**Erläuterung der wichtigsten Inhaltstypen**

|  |  |
| --- | --- |
| Inhaltstyp | Funktion |
| Adressenblock | Im Adressenblock kann eine Adresse hinterlegt werden. Die Ausgabe enthält zusätzlich den Standort aufgezeigt in einer GoogleMaps. |
| Arbeitsraum | Innerhalb eines Arbeitsraumes werden Daten ausgetauscht. Der Arbeitsraum kommt hauptsächlich bei Projekten zum Einsatz. |
| Auflistungsblock | 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 Platform. Innerhabl der ContentPage werden Inhalte wie Textblocke, Auflistungsblocke...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. |
| Traktandum | Ein Traktandum kann einem Meeting hinzugefügt werden. |

Tabelle : Erläuterung der Inhaltstype

### Berechtigungen

Die Rechte auf der Demoplattform werden anhand eines definierten Workflows vergeben. Es wird bei den Berechtigungen zwischen globalen und lokalen Berechtigungen unterschieden. **Globale Berechtigungen** sind nicht Typenspezifisch (beziehen sich nicht auf Inhaltstype) und werden an untergeordnete Elemente vererbt. Da die globalen Berechtigungen auf der obersten Ebene vergeben werden, ist die Vererbung auf der ganzen Plattform vergeben. Eine globale Berechtigung kann nicht unterbrochen werden.

**Typenbezogene Berechtigungen** (beziehen sich auf Inhaltstype) und können an untergeordnete Elemente vererbt werden. Diese Vererbung kann unterbrochen werden.

Da für die Demoplattform nur ein Benutzer pro Rolle erstellt wird und somit keine individuellen Berechtigungen auf Elemente vergeben werden müssen, werden globale Berechtigungen wie folgt vergeben:

|  |  |  |  |
| --- | --- | --- | --- |
| Benutzer | Rolle | Berechtigung | Funktion |
| Administrator | Administrator | Global | Administriert die Inhalte der Plattform. Er kann die Struktur anpassen und Module hinzufügen. |
| Autor | User | Global | Kann Inhalte erfassen / bearbeiten und löschen |
| Anonym | Reader | Global | Hat Leserechte auf Inhalte |

Tabelle : Berechtigungstabelle

## Betriebskonzept

Im folgenden Abschnitt wird beschrieben, wie das Skript effektiv betrieben werden soll.

Die Pakete „eGov“ und „Teamraum buildout“ werden erläutert, da diese die Hauptpakete für das automatische Aufsetzen der Demo-Plattform darstellen.

### eGov

Das Paket eGov ist der Kern der teamraum-Produkte. Bei 4teamwork werden in diesem Paket auch die Profile für die verschiedenen webbasierten Standardlösungen (teamraum Collab, teamraum Intranet und teamraum Web) definiert. Im Profil sind die Daten, welche nach Aufsetzen der Demoplattform zur Verfügung stehen sollen, definiert. Für die Demoplattform wird ein neues Profil benötigt, welches in diesem Paket hinzugefügt werden muss.

### Teamraum Buildout

Buildout ist ein Buildsystem welches auf python basiert und dazu dient, Webanwendungen automatisch zu installieren. Buildout sammelt Pakete und installiert daraus die Webanwendung. Bei 4teamwork wird in diesem Paket auch das runscript definiert, welches in diesem Projekt dazu dient, das definierte Profil auszuwählen welches auf der Plattform installiert werden soll. Bei 4teamwork wird differenziert zwischen einem Buildout welches benötigt wird um Entwicklungen durchzuführen und einem Buildout, welches für den produktiven Einsatz verwendet wird. Für dieses Projekt wird mit dem Buildout „development.cfg“ gearbeitet, welches für Entwicklungen verwendet wird

### Erstellen und Einpflegen des Skripts

Das Skript wird in einem Branch des Pakets „Teamraum buildout“ erstellt. Es wird ein Ordner „Skripts“ hinzugefügt, welcher das Runscript „install\_demo\_site.py“ enthält.

Zudem wird ein Branch des Pakets „eGov“ mit dem Profil „teamraum\_demo“ erstellt und der Inhalt in diesem Branch definiert.

### Versionierung

Nachdem das Skript die geforderten Funktionen erfüllt, wird die Version 1.0 in den Master des betroffenen Pakets gemerged. Für eine neue Version wird wiederum ein Branch des Masters erstellt und dieser nach Fertigstellung gemerged.

### Hotfixes

Hotfixes werden ähnlich wie die Implementierung neuer Versionen gehandelt. Wird ein Fehler im Skript entdeckt, wird ein Branch des Masters erstellt, in welchem der Fehler behoben wird. Dieser wir nach Fertigstellung wieder in den Master gemerged.

### Betrieb auf dem Server

Die Ausführung des Skripts auf dem Server wird ähnlich wie die Ausführung auf einer lokalen Maschine funktionieren. Einige kleinere Anpassungen sind nötig um diese Funktionalität gewährleisten zu können. Die Implementation wird ein einem Release 2.0 des Skripts nach Abschluss des Projekts erstellt.

## Testkonzept.

### Ausgangslage

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

### Testmethode

Für dieses Projekt wird ein Testvorgehen nach folgendem Phasenmodell erarbeitet.



Abbildung : Phasenmodell

### Testplanung

Der Testumfang beschränkt sich auf die Funktionalität des Skripts 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.

### Testziele

#### Die Hauptziele betreffend den Tests sind nachfolgend aufgelistet:

|  |  |
| --- | --- |
| Nr. | Ziele |
| 1 | Fehler im Skript werden gefunden. |
| 2 | Fehlerhafte Inhalte auf der Plattform werden gefunden. |
| 3 | Fehlerhaftes Verhalten betreffend vordefinierten Benutzern werden gefunden. |
| 4 | Die Stabilität des Skripts ist ermittelt. |

Tabelle : Hauptziele Tests

#### Hilfsmittel

Als Hilfsmittel dienen der Testplan und das Testprotokoll, welche im Anhang dieses Dokuments beigefügt sind. Es besteht kein Ausbildungsbedarf für die testende Person.

### Testvorbereitung

Das Skript wird auf einem Testnotebook bereitgestellt. Das Testprotokoll und der Testplan liegen bei. Bei Bedarf / Fehler werden die Tests zusätzlich iterativ durchgeführt.

### Spezifische Testfälle betreffend Skript

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 1 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Starten des Skripts per Terminalbefehl: „python demo\_teamraum.py“ | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Das Skript wird nicht durchgeführt, da kein Instanzenordner mitgegeben wurde, eine Fehlermeldung wird geworfen |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 2 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Starten des Skripts per Terminalbefehl: „python demo\_teamraum.py teamraum-buildout“ | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Das Skript wird höchstwahrscheinlich durchgeführt, ein Fehler betreffend System-Python kann aber erwartet werden. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |
| Testfall Nr. | | | | 3 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Starten des Skripts per Terminalbefehl: „python demo\_teamraum.py –p ’Pfad zu einem python’ teamraum-buildout“ | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Das Skript wird durchgeführt, falls ein korrektes Python als Option angegeben wird. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 4 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| 1. Starten des Skripts per Terminalbefehl „python demo\_teamraum.py –p ’Pfad zu einem python’ 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’ teamraum-buildout“ | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Das Skript wird abgebrochen (beim zweiten Ausführen) , mit der Fehlermeldung, dass der Instanzenordner schon existiert. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 5 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Über mehrere Terminals wird das Skripts mehrmals gestartet per Terminalbefehl: „python demo\_teamraum.py –p ’Pfad zu einem python’ teamraum-buildout“ | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| 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. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

### Spezifische Testfälle betreffend Plattform

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 6 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Ist die Plattform nach durchlaufen des Skripts vorhanden? Verbinden über localhost:8080 | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Die Plattform ist vorhanden. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 7 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Ist die Struktur wie in Kapitel 13.5.1 beschrieben vorhanden? Verbinden über localhost:8080 und die Plone-Seite anzeigen lassen. | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Die Struktur ist vorhanden. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 8 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Sind folgende Benutzer unter <http://localhost:8080/gemeinde/user_management> vorhanden?   * Administrator * Autor * Anonym | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Die Benutzer sind vorhanden. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Testfall Nr. | | | | 7 |
| Tester | | Thomas Buchberger | **Datum** | 22.05.2014 |
| Beschreibung | | | | |
| Funktionieren die Berechtigungen laut Tabelle xy ?  Hat der Benutzer „Anonym“ Leserechte auf alle Inhalte?  Hat der Benutzer „Autor“ die Möglichkeit Inhalte anzupassen?  Kann der Benutzer „Administrator“ Inhalte hinzufügen? | | | | |
| erwartetes Ergebnis | **effektives** **Ergebnis** | | | **erfüllt** |
| Die Berechtigungen funktionieren gemäss Tabelle xy. |  | | |  |
| Korrekturen / Anpassungen: | | | | |
|  | | | | |

## Einführungskonzept

Das Skript wird nach Abschluss des Projekts auf den Servern von 4teamwork unter dem Pfad ./scripts abgelegt. Um python-Skripte ausführen zu können wird ein Python-Interpreter auf dem Server verlangt. Diese Voraussetzung ist auf allen Servern von 4teamwork gegeben.

Wird das Skript direkt über den Konsolenbefehl „python demo\_teamraum.py“ ausgeführt, benötigt dieses lediglich Leserechte.

Soll das Skript auch über den Konsolenbefehl „./demo\_teamraum.py“ ausgeführt werden, müssen folgende Anforderungen erfüllt sein:

* Im Header des Skripts muss auf der ersten Zeile der Eintrag „#! Pfad-zum-python

“ stehen:

* Das Skript muss excecutable –Rechte haben.

Die eigentliche Einführung auf dem Server ist mit dem kopieren des Skripts in den erwähnten Pfad erledigt.

## Zusammenfassung

Das Produkt konnte in dieser Phase konkretisiert werden. Die Ergebnisse wurden erarbeitet und das Produkt ist auf einer verlässlichen Grundlage geplant und kann dementsprechend auch realisiert werden. Die Konzeption hat ergeben, dass die Realisierung des Projekts als sinnvoll eingestuft werden kann. Die Wirtschaftlichkeit ist gewährleistet, die Ziele und die Anforderung an das Projekt werden eingehalten.

## Freigabe Realisierung

Die Phase Realisierung wird freigegeben:

|  |  |  |
| --- | --- | --- |
| Datum | Name / Auftraggeber | Unterschrift |
| 19.05.2014 | Pascal Habegger |  |

# Realisierung

## Legende

Terminalbefehle, Quellcode und Fehler werden wie folgt dargestellt:

Terminaleingaben

Quellcode

Fehler

## Bestehende Umgebung verifizieren

Die technischen Details der Umgebung sind in Kapitel 13.4 aufgeführt. Die Realisierung wird in der folgenden Struktur durchgeführt.

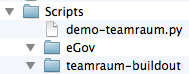


Abbildung : Bestehende Dateistruktur

Die Namensgebung wird von Kapitel xy des Konzepts übernommen.

Die Systemstruktur ist in Kapitel xy aufgezeigt.

Die benötigte Software ist wie in Kapitel xy beschrieben vorhanden.

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

## Konfiguration festlegen

Für das Paket eGov wird ein Branch erstellt

Der Branch erhält den Namen „mw-ipa\_profile\_demo-content“

Für das Paket teamraum-buildout wird ein Branch erstellt

Der Branch erhält den Namen „mw-ipa\_runscript“

Das Skript überschreibt eine existierende Plattform auf Anfrage

Es muss zwingend eine Meldung ausgegeben werden, falls die Plattform existiert.

Der Pfad zu einem gültigen Python muss nicht angegeben werden

Falls kein Pfad an den Option.Parser übergeben wird, soll ein Standardpfad verwendet werden.

Es muss sichergestellt sein, dass das Buildout korrekt ausgeführt worden ist

Falls Daten Fehlen, wird ein Exception error geworfen.

Zu Demonstrationszwecken werden Standortmeldungen bei Ordnerwechsel angezeigt

Diese Implementation wird bei der Einspielung auf einem Server wieder entfernt.

## Vorarbeiten

Das Paket eGov wird von Github geklont und in den im Kapitel 15.2 definierten Ordner abgelegt.

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

Der Branch des Pakets eGov wird erstellt.

git checkout –b mw-ipa\_profile\_demo-content

Das Paket buildout-base wird von git.4teamwork geklont und in den im Kapitel 15.2 definierten Ordner abgelegt.

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

Der Branch des Pakets buildout-base wird erstellt.

git checkout –b mw-ipa\_runscript

## Installationsskript erstellen

Nachfolgend werden die wichtigsten Implementationen des Installationsskripts demo-teamraum.py aufgezeigt. Das vollständige Skript ist als Anhang beigefügt.

### Bibliotheken Installationsskript

Verschiedene Bibliotheken müssen im Skript implementiert werden. Für die untenstehenden Bibliotheken werden nur die benötigten Funktionen erläutert.

|  |  |
| --- | --- |
| Modul | Funktion |
| os | Kommandozeilenargumente können eingegeben weden |
| git | Pakete klonen |
| sys | Ermöglicht das Ansprechen des Python-Interpreters |
| optparse |  |

Abbildung : Bibliothek im Skript

### Hauptelemente Installationsskript

Der Grundaufbau des Installationsskripts beinhaltet folgende Hauptelemente:

* Instanzenordner erstellen
* Paket buildout-base in den Instanzenordern herunterladen
* Symbolischer Link auf development-ipa.cfg erstellen
* Interpreter Bootstrap ausführen
* Buildout ausführen
* Instanz starten

### Wichtige Elemente des Quellcodes

**Zeichensatz**

Damit beim Ausführen des Skripts keine Fehlermeldung betreffend Umlauten geworfen werden, muss die erste Zeile des Skripts folgenden Eintrag enthalten:

# -\*- coding: utf-8 -\*-

**Hervorhebung**

Um interessante Stellen im Skript hervorzuheben, wurde die Funktion „printstars“ erstellt.

def printstars(toprint):

print "\*" \* 20

print toprint

print "\*" \* 20

**OptionParser**

Der OptionParser ist ein Module der Bibliothek Optparse. Er dient dazu, einem Python-Skript Optionen und Argumente mitgeben zu können. In diesem Projekt wurde der OptionParser verwendet um als Argument den Instanzenordners mitzugeben, um als Option den Python-Pfad 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 definiert worden sein.

parser.set\_defaults(pythonpath=sys.executable)

Als Python wird in diesem Fall das System-Python verwendet. Jedoch ist zu empfehlen das Python per Option mitzugeben, da nicht garantiert werden kann, dass das System-Python funktionieren wird.

Falls die Option nicht angegeben und das Skript ohne Option ausgeführt wird, wird folgende Fehlermeldung 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 give me 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 allready exists: '%s'" % DIRNAME

sys.exit()

Um zu prüfen ob die Hauptelemente der Instanz, nach durchführen des 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)

### Zusammenfassung

Das Skript setzt, mit den mitgegebenen Parametern, lokal eine Plattform auf. Um dieser Plattform Inhalte mitzugeben wird ein Profil im Paket „eGov“ erstellt.

## Profil definieren im Paket „eGov“

### Hauptelemente des Profils

Der Grundaufbau des Profils teamraum-demo sieht folgendermassen aus:

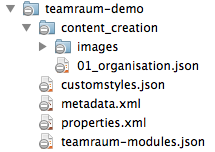


Abbildung : Grundaufbau des Profils

**Erläuterung der Stuktur**

|  |  |
| --- | --- |
| Datei | Funktion |
| teamraum-demo | Stellt den Namen des Profils dar |
| Content\_creation | Enthält alle visuellen Daten, welche als Inhalt des Profils dienen. |
| 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. |
| metadata.xml | ?? |
| properties.xml | ?? |
| teamraum-modules | Die gewünschten Module werden in dieser Datei aktiviert. |
|  |  |

Tabelle : Erläuterung der Inhaltstype

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

#### teamraum-modules.json

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



Abbildung : Module

Das contact-Modul ermöglicht das Hinzufügen eines Kontakteordners.

#### 01\_organisation.json

Die Struktur von Kapitel 14.5.1 wird in dieser Datei eingetragen. Als Beispiel wird der Aufbau einer Inhaltsseite und eines Kontakteordners aufgezeigt.



Abbildung : Aufbau Inhaltsseite 1



Abbildung : Aufbau Inhaltsseite 2

**Erläuterung des Aufbaus der Inhaltsseite und Kontakte**

|  |  |
| --- | --- |
| Metadaten | Funktion |
| \_path | Der Pfad, wo die Inhaltsseite / Der Kontaktordner 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 | Die Ansicht, welche für die Inhaltsseite / den Kontakteordner definiert ist |

Tabelle : Erläuterung Aufbau Inhaltsseite

### Abhängigkeit

Ohne Anpassung am Buildoutskript „development.cfg“ welches sich im Paket 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 ein Runskript in einem Branch des Pakets „eGov“ erstellt.

## Runskript erarbeiten in einem Branch des Pakets „buildout-base“

Das Runskript übernimmt folgende Hauptaufgaben:

* Theme (Aussehen der Plattform wird definiert)
* Root um Plattform korrekt aufsetzen zu können wird vergeben
* Rechte werden vergeben
* Ein definiertes Profil wird angewählt
* Der Name der Plattform wird verbgeben

## Programmablaufplan

Im Programmablaufplan wird aufgezeigt, wie die Durchführung zum Aufsetzen der Plattform funktioniert.



Abbildung : PAP

## Benutzeranleitung / Betriebshandbuch

Das Benutzerhandbuch beschreibt, wie das Installationsskript demo-teamraum.py erfolgreich gehandhabt wird.

## Testprotokoll

* Die Testprotokolle fassen die Ergebnisse zusammen:
* Testobjekt
* Angaben zur Durchführung (konnten alle Tests vom Konzept durchlaufen werden?
* Testresultate
* Was passiert mit einem „Failtesting“? „Re-Testing“ oder leichter Fehler?
* Testfazit und Empfehlung

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Nr. | Testobjekt | Testbeschreibung | Erwartung | Ergebnis | Visum |
| 1B |  |  |  |  |  |
| 2W |  |  |  |  |  |

Tabelle : Testprotokoll

## Freigabe Einführung

Die Phase Einführung wird freigegeben:

|  |  |  |
| --- | --- | --- |
| Datum | Name / Auftraggeber | Unterschrift |
|  | Pascal Habegger |  |

# Quellenverzeichnis

|  |  |  |  |
| --- | --- | --- | --- |
| **Quelle** | **Autor** | **Jahr** | **Thema** |
| Internet:  <http://de.wikipedia.org/wiki/Lenkungsausschuss> | Pseudonym: CatcherBlock | 2004 | Lenkungs-ausschuss |
| Buch:  Modul 306: IT-Kleinprojekt abwickeln TeilB, Auflage2 | Kurt Dadertscher / Peter Hirzel | 2007 | Informationen betreffend Projekten |
| Internet:  <http://books.google.ch/books?id=FBnZVpUMJzMC> | Hans W. Wieczorrek / Peter Mertens | 2008 | Situations-analyse |
| Internet:  <http://www.igrm.ch/app/download/2471810817/Zugriffskonzept+BK.pdf?t=1289218378> | Zosso Karin | unbekannt | Zugriffs-konzept |
| Internet:  <https://plone-3-benutzerhandbuch.readthedocs.org/en/latest/glossar.html> | Jan Ulrich Hasecke und Thomas Lotze | unbekannt | Benutzer-handbuch Plone |
| Internet:  <http://www.4teamwork.ch/loesungen> | Bernhard Bühlmann | 2013 | Produkte 4teamwork |
|  |  |  |  |

# Glossar

|  |  |
| --- | --- |
| Begriff | Bedeutung |
| Brainstorming | Eine Methode zur Ideenfindung. |
| Branch | Ein Zweig eines Paketes. |
| Buildout | Python-basiertes Buildsystem zur automatischen Installation von Webanwendungen. |
| Git | Webbasierte Verteilungs- und Versionierungssoftware |
| Github | Hosting Dienst für Software-Entwicklungsprojekte |
| Human Testing | Software wird durch Menschen getestet. |
| Kanban | Eine Methode zur Produtionsprozessteuerung. |
| Plone | Ein python-basiertes Content Management System. |
| Profil |  |
| Python | Eine interpretierte höhere Programmiersprache |
| Releaseprozess |  |
| Terminal | Eine Kommandozeile von Apple entwickelt für Mac. |
| Uability | Benutzerfreundlichkeit. |
| VMWare | Virtuelle Maschinen können auf Computern betrieben werden. |
|  |  |

# Unterschriften für Abnahme

|  |  |  |
| --- | --- | --- |
| Datum | Name / OE | Unterschrift |
|  |  |  |
|  |  |  |

# Anhang

Ausgedruckter Code, Sitzungsprotokolle, usw.: Vollständig aufführen und der Dokumentation beilegen.

Listings von Scripten und Programmen. Die Eigenleistung der Kandidatin oder des Kandidaten muss vollständig dokumentiert sein. Automatisch generierten Code weglassen, wenn für das Verständnis nicht zwingend nötig.

Falls Handbücher erstellt wurden, können diese hier als Anhang beigelegt werden.

Firmenstandards zwingend in den Anhang!