Dokumentation IPA yatplaner

IPA in Applikationsentwicklung

Auszubildender - Marius Küng

Auftraggeber - allink.creative GmbH

Projektleiter - Silvan Spross

Chefexperte - Hans Riesenmann

Experte - Antonio Di Luzio

Durchführungsort - allink.creative GmbH

Informatikmittelschule Basel

13.03. - 26.03.2012

Inhaltsverzeichnis

I. Umfeld und Ablauf **3**

1. Aufgabenstellung **4**

1.1. Titel der Facharbeit . . . 4
1.2. Thematik . . . 4
1.3. Klassierung . . . 4
1.4. Durchführungsblock . . . 4
1.5. Ausgangslage . . . 5
1.6. Detaillierte Aufgabenstellung . . . 5
1.7. Mittel und Methoden . . . 5
1.8. Vorkenntnisse . . . 6
1.9. Vorarbeiten . . . 6
1.10. Neue Lerninhalte . . . 6
1.11. Arbeiten in den letzten 6 Monaten . . . 6

2. Einleitung **7**

2.1. Projekorganisation . . . 7
2.2. Vorkenntnisse . . . 7
2.3. Vorarbeiten . . . 8
2.4. Firmenstandards . . . 8

II. Projekt **10**

3. Projektbeschreibung **11**

3.1. Umfeld . . . 11
3.2. Präzisierung der Aufgabenstellung . . . 11
3.3. Analyse Wochenplaner (IST-Zustand) . . . 11
3.3.1. Modellierung . . . 11
3.3.2. Darstellung . . . 12
3.3.3. Funktionsumfang . . . 12

3.3.4. Defizite 12
3.4. Definition Ziele 13
3.4.1. MUSS-Ziele 13
3.4.2. KANN-Ziele 14

4. Realisierung **15**
4.1. ERM 15
4.2. System-Beschreibung 15
4.3. Modellierung (Django, aufsetzen Projekt) 15
4.4. Templating Wochenansicht 15
4.5. JS Funktionalität (CRUD mit jQueryUI & AJAX) 15
4.6. Implementierung 15
4.7. Erreichte Ziele 15

5. Testing **16**
5.1. Testfälle erfassen 16
5.2. Testfälle ausführen 16
5.3. Testbericht (Ziele) 16

III. Arbeitsjournal **17**
5.4. 13.03.2012 18
5.5. 14.03.2012 18

6. Abbildungsverzeichnis **19**

7. Tabellenverzeichnis **20**

Teil I.

Umfeld und Ablauf

1. Aufgabenstellung

1.1. Titel der Facharbeit

Webapplikation zur Ressourcenplanung von allink.creative

1.2. Thematik

Es soll eine Webapplikation mit Django erstellt werden, mit welcher die Geschäftsleitung die Ressourcenplanung der Mitarbeiter vornehmen kann. Damit soll eine ältere Webapplikation abgelöst werden.

1.3. Klassierung

- Applikationsentwicklung OO
- UNIX / Linux
- andere_Programmiersprache

1.4. Durchführungsblock

Startblock 1: 12.03.2012 - 23.04.2012
IPA-Durchführung: 12.03.2012 - 23.04.2012
Einreichung bis: Montag, 30.01.2012

1.5. Ausgangslage

Bei allink besteht seit Mitte 2010 ein rudimentäres Ressourcenplanungstool. Zur Entwicklung wurden damals jedoch Technologien verwendet, die heute nicht mehr zur Kernkompetenz von allink zählen. Da dieses Tool jedoch jeden Freitag zur Planung der nächsten Woche verwendet wird, ist es seit längerem überfällig es in die bestehende Managementapplikation zu integrieren.

Dank der übermässig langen Testphase des Prototypen sind nun die Anforderungen an das definitive Tool gut bekannt. Daher soll eine Webapplikation mit Django erstellt werden, mit welcher die Geschäftsleitung von allink.creative die Ressourcenplanung der Mitarbeiter vornehmen kann. Damit soll eine ältere Webapplikation abgelöst werden.

1.6. Detaillierte Aufgabenstellung

Das bestehende Tool namens "Yatplaner" soll als Modul im bestehenden Management Tool namens "allink.planer" reimplementiert werden. Dabei soll die Bedienbarkeit verbessert werden. Das Ziel ist es das neue Tool so intuitiv bedienen zu können, dass für die Geschäftsleitung keine Schulung nötig ist. Der Praxistest wird voraussichtlich in der letzten IPA Woche an der Wochenplansitzung durchgeführt.

Nebst der begleitenden IPA Dokumentation, wo unter anderem der Funktionsumfang des bestehenden Tools analysiert wird, wird keine zusätzliche Dokumentation gefordert. Der Funktionsumfang des bestehenden Tools soll vom Lernenden in einer Analysephase aufgenommen werden. Dabei sollen die bestehenden Features als Muss- und mögliche neue Features als Kann-Ziele ausformuliert werden.

Der Quellcode des bestehenden Tools ist unter folgender Adresse einsehbar:
https://github.com/sspross/yatplaner/tree/rails

1.7. Mittel und Methoden

Folgende Technologien sind zwingend zu verwenden:

- Python 2.6
- Django 1.3
- Piston 2.3
- jQuery 1.8

- HTML5
- CSS3

Das Tool soll in folgenden Browsern fehlerfrei funktionieren:

- Firefox >= 8
- Chrome >= 10
- Safari >= 5

Der Internet Explorer muss explizit nicht unterstützt werden.

1.8. Vorkenntnisse

Dem Lernenden sind alle genannten Technologien bereits bekannt. Seit Beginn seines Praktikums im August 2011 setzt er sich damit auseinander. Gewisse Kombinationen wie z.B. mit jQuery einen AJAX Request zu erstellen, sind jedoch Neuland.

1.9. Vorarbeiten

Es findet keine explizite Vorarbeit statt.

1.10. Neue Lerninhalte

Wie bereits erwähnt sind dem Lernenden alle Technologien bereits bekannt. Jedoch sind gewisse Kombinationen noch nie vom Lernenden selbst angewandt worden. Das Know-how ist bei allink ausreichend vorhanden. Der Lernende kann zudem auf eine Vielzahl von bestehenden Projekten zurückgreifen, wo er unzählige Beispiele studieren kann.

1.11. Arbeiten in den letzten 6 Monaten

Der Lernende hat überwiegend Webseiten mit den oben genannten Technologien umgesetzt. Zu seinen umfassendsten Arbeiten zählen bis jetzt eine Webseite eines Immobilienunternehmens und einer Eventplattform eines Finanzkonzerns. Für ersteres arbeitete der Lernende rund 250 Stunden daran.

2. Einleitung

2.1. Projekorganisation

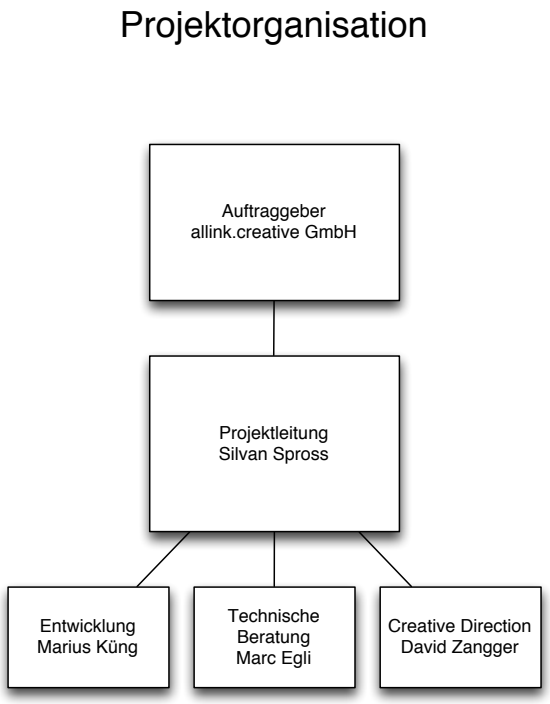

Abbildung 2.1.: Projekorganisation[1]

2.2. Vorkenntnisse

Technologien

- Python Grundkenntnisse
- Django Fortgeschrittene Kenntnisse

[1] Eigene Darstellung

- Piston Grundkenntnisse
- jQuery Fortgeschrittene Kenntnisse
- HTML5 Gute Kenntnisse
- CSS3 Gute Kenntnisse
- AJAX Fortgeschrittene Kenntnisse

Anwendungen

- Mehrere komplette Webauftritte realisiert
- Applikationen in Django erstellt(News, Blog, Produkteübersicht)
- Schnittstellen programmiert (XML, JSON)
- Per AJAX dynamische Inhalte laden und einfügen
- Dynamische Formulare abschicken und Request entgegenehmen
- DOM-Elemente manipulieren, per Events steuern

2.3. Vorarbeiten

Ich habe, um einen ersten Eindruck über die Funktionalität zu erhalten, den bestehenden yatplaner ausprobiert. Ausserdem den allink.planer studiert in den, der Wochenplaner implementiert wird. Ansonsten fand keine explizite Vorarbeit statt.

2.4. Firmenstandards

- Betriebsystem: Mac OS X
- Editor: Textmate
- Entwicklungsumgebung: Python, Django (Python-Framework für Webapplikationen)
- Virtuelle Testumgebung: Django Serversimulation (per Terminal steuerbar)
- Deployment: per fabric-script auf Apache-Server mit WSGI-Protokoll
- Versionierung: Git, auf Github gehostet

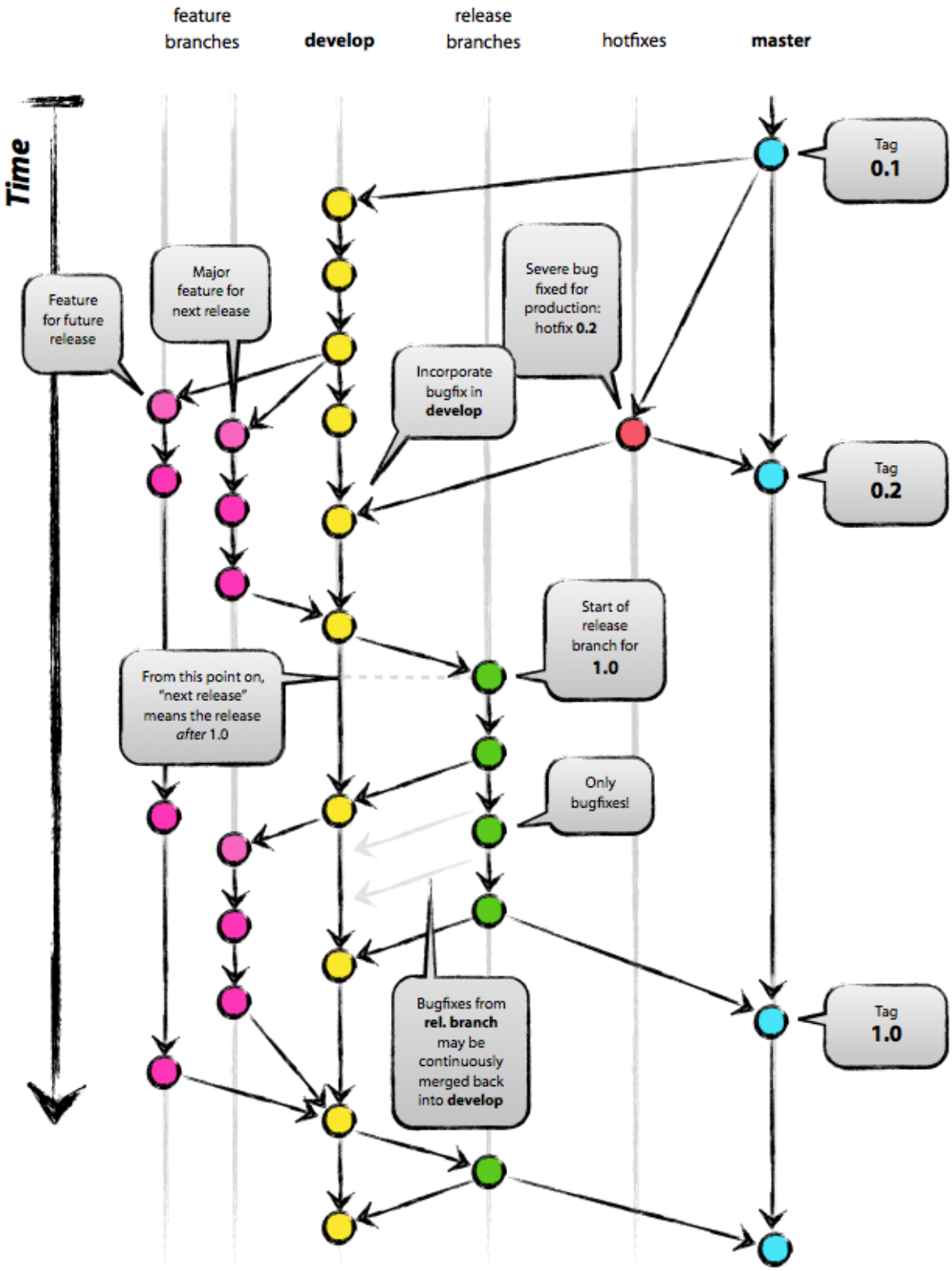

Abbildung 2.2.:
"A successful Git branching model" von Vincent Driessen

Teil II.

Projekt

3. Projektbeschreibung

3.1. Umfeld

Momentan läuft der Wochenplaner vom allink.planer getrennt und kann nicht in den allink Projektablauf einbezogen werden. Das Tool fungiert mehr als eine Tafel auf die Post-It's geklebt werden. D. h. die Aufgaben sind nicht direkt im Projektmanagenttool mit der Person verbindet. Somit bietet das Tool keinen Mehrwert bei der Auswertung des Projektablaufs.

Das Tool ist in Ruby on Rails programmiert, was nicht (mehr) dem Firmenstandard entspricht, und kann deshalb nur mühsam gewartet und erweitert werden.

3.2. Präzisierung der Aufgabenstellung

Da die Aufgabenstellung detailliert erfasst wurde und ein Prototyp existiert, ist relativ klar was die eigentliche Aufgabenstellung umfasst. Zur weiteren Präzisierung, werde ich eine detaillierte IST-Analyse vornehmen. Diese wird mit dem Auftraggeber besprochen, damit die MUSS- und KANN-Ziele erfasst werden können und ersichtlich wird, welche Funktionalitäten überhaupt noch Sinn machen und was weggelassen werden kann.

3.3. Analyse Wochenplaner (IST-Zustand)

3.3.1. Modellierung

Analysiert anhand der Rails-Modelle

Person

- Name
- Locked Days (gesperrte Tage, Auswahl von Montag bis Freitag)
- Beschreibungstext für jeden gesperrten Tag

Task

- Name
- Datum (Fälligkeitsdatum, DATE)
- Person (Fremdschlüssel)
- Dauer (Stundenanzahl)

3.3.2. Darstellung

Der Wochenplaner verfügt über mehrere Darstellungen:

- Kalenderansicht der aktuellen Woche (Screenshot)
- Übersicht aller Angestellten
- Monatsansicht einer Person
- Übersicht aller erfassten Tasks

3.3.3. Funktionsumfang

- Neue Task durch Doppelklick auf Personentag per Dialog erfassen (Screenshot)
- Klick auf Task öffnet die Detailansicht (um bspw. Änderungen vorzunehmen)
- Verschiebung einer Task per Drag & Drop
  - Task in anderen Tag verschieben
  - Task einer anderen Person zuweisen
- In der Kalenderwoche per Link vor- oder zurückspringen
- weitere 7 Tage in der Ansicht hinzufügen (kann auch rückgängig gemacht werden)
- Wochentage können für wiederkehrende Events (Schule, Frei) markiert werden

3.3.4. Defizite

- Die Sperrtage haben kein Startdatum und sind für jeden Wochentag erfasst

3.4. Definition Ziele

In einer Sitzung mit der Projektleitung wurde der IST-Zustand besprochen und die Ziele definiert.

3.4.1. MUSS-Ziele

Nr	Funktion	Beschreibung
1	Task erfassen	Der Benutzer kann eine neue Task erfassen.
2	Task Namen geben	Der Benutzer kann einem Task einen Namen geben.
3	Task Datum geben	Der Benutzer kann einem Task ein Datum geben.
4	Task Person zuweisen	Der Benutzer kann einem Task eine Person zuweisen.
5	Task Dauer geben	Der Benutzer kann einem Task eine Dauer (in Stunden) geben.
6	Task bearbeiten	Der Benutzer kann einen Task bearbeiten.
7	Task löschen	Der Benutzer kann einen Task löschen.
8	Person hinzufügen	Der Benutzer kann eine Person dem Wochenplaner hinzufügen.
9	Sperrtag Person zuweisen	Der Benutzer kann einer Person einen Sperrtag zuweisen.
10	Sperrtag Namen geben	Der Benutzer kann einem Sperrtag einen Namen geben.
11	Sperrtag bearbeiten	Der Benutzer kann einen Sperrtag bearbeiten.
12	Sperrtag löschen	Der Benutzer kann einen Sperrtag löschen.
13	Task einem Projekt zuweisen	Der Benutzer kann einen Task einem Projekt zuweisen.

Tabelle 3.1.: Zwingend umzusetzende Funktionen des Prototypen

3.4.2. KANN-Ziele

Nr	Funktion	Beschreibung	Priorität
14	Task duplizieren	Der Benutzer kann einen Task duplizieren.	1
15	Sperrtag Startdatum geben	Der Benutzer kann einem Sperrtag ein Startdatum geben.	3
16	Warnmeldung Max	Es erscheint eine Warnmeldung wenn die Summe aller Arbeitsstunden pro Tag 8h überschreiten.	4
17	Tasks sortieren	Tasks können innerhalb eines Tages sortiert werden.	2

Tabelle 3.2.: Nicht zwingend umzusetzende Funktionen des Prototypen

4. Realisierung

4.1. ERM

4.2. System-Beschreibung

4.3. Modellierung (Django, aufsetzen Projekt)

4.4. Templating Wochenansicht

4.5. JS Funktionalität (CRUD mit jQueryUI & AJAX)

4.6. Implementierung

4.7. Erreichte Ziele

5. Testing

5.1. Testfälle erfassen

5.2. Testfälle ausführen

5.3. Testbericht (Ziele)

Teil III.

Arbeitsjournal

5.4. 13.03.2012

Beginn IPA yatplaner

- Dokumentation auf github hosten und versionieren
- Kapitelstrukur aufbauen Teil 1

- Zeitplan erstellen (Planung)
- Aufgabenstellung erfassen
- Projektorganisation grafisch erfassen
- Designbesprechung mit Dave
- Vorkenntnisse erfassen
- Vorarbeiten erfassen
- Firmenstandards erfassen Teil 2

- Umfeld erfassen
- IST-Analyse vorerfassen (Ansichten, Funktionalität)

5.5. 14.03.2012

- IST-Analyse vorerfassen (Modellierung der Rails App)
- Sitzungen mit Silvan Spross und Marc Egli zur Definition der MUSS- und KANN-Ziele
- MUSS-Ziele erfasst
- KANN-Ziele erfasst

6. Abbildungsverzeichnis

2.1. Projekorganisation 7
2.2. “A successful Git branching model” von Vincent Driessen 9

7. Tabellenverzeichnis

3.1. Zwingend umzusetzende Funktionen des Prototypen 13
3.2. Nicht zwingend umzusetzende Funktionen des Prototypen 14