|  |
| --- |
| Elmer Lukas, Heidt Christina, Steiner Diego, Treichler Delia, Waltenspül Remo  4. März 2011 |

|  |
| --- |
| SE2 Projekt MRT |
| Projektplan |
|  |

****

# Dokumentinformationen

## Änderungsgeschichte

|  |  |  |  |
| --- | --- | --- | --- |
| Datum | Version | Änderung | Autor |
| 03.03.2011 | 1.0 | Erste Version des Dokuments | WR |
| 04.03.2011 | 1.1 | Anpassung und Korrektur | TD |

## Inhaltsverzeichnis

[1 Dokumentinformationen 1](#_Toc287018752)

[1.1 Änderungsgeschichte 1](#_Toc287018753)

[1.2 Inhaltsverzeichnis 1](#_Toc287018754)

[2 Einführung 3](#_Toc287018755)

[2.1 Zweck 3](#_Toc287018756)

[2.2 Gültigkeitsbereich 3](#_Toc287018757)

[2.3 Definitionen und Abkürzungen 3](#_Toc287018758)

[2.4 Referenzen 3](#_Toc287018759)

[2.5 Übersicht 3](#_Toc287018760)

[3 Projekt Übersicht 4](#_Toc287018761)

[3.1 MRT (Mobile Reporting Tool) 4](#_Toc287018762)

[3.1.1 Ablauf 4](#_Toc287018763)

[3.1.2 Optionale Features 4](#_Toc287018764)

[3.2 Zweck und Ziel 4](#_Toc287018765)

[3.3 Annahmen und Einschränkungen 5](#_Toc287018766)

[4 Projektorganisation 6](#_Toc287018767)

[4.1 Team 6](#_Toc287018768)

[4.1.1 Lukas Elmer (Abk. EL) 6](#_Toc287018769)

[4.1.2 Christina Heidt (Abk. HC) 6](#_Toc287018770)

[4.1.3 Diego Steiner (Abk. SD) 6](#_Toc287018771)

[4.1.4 Delia Treichler (Abk. TD) 6](#_Toc287018772)

[4.1.5 Remo Waltenspül (Abk. WR) 7](#_Toc287018773)

[4.2 Anmerkung 7](#_Toc287018774)

[4.3 Organisationsstruktur 7](#_Toc287018775)

[4.4 Externe Schnittstellen 7](#_Toc287018776)

[5 Management Abläufe 7](#_Toc287018777)

[5.1 Projekt Kostenvoranschlag 7](#_Toc287018778)

[5.2 Projektplan 9](#_Toc287018779)

[5.2.1 Zeitplan 9](#_Toc287018780)

[5.2.2 Iterationsplanung / Milestones 10](#_Toc287018781)

[5.2.2.1 Inception 10](#_Toc287018782)

[5.2.3 Besprechungen 11](#_Toc287018783)

[5.2.4 Abgabe 11](#_Toc287018784)

[6 Risiko Management 11](#_Toc287018785)

[7 Arbeitspakete 11](#_Toc287018786)

[8 Infrastruktur 13](#_Toc287018787)

[8.1 Software 13](#_Toc287018788)

[8.2 Kommunikation 13](#_Toc287018789)

[9 Qualitätsmassnahmen 14](#_Toc287018790)

[9.1 Allgemein 14](#_Toc287018791)

[9.1.1 Regelmässige Teamsitzungen &teamfördernde Massnahmen 14](#_Toc287018792)

[9.1.2 Sourcecode Management 14](#_Toc287018793)

[9.1.3 Issuetracking 14](#_Toc287018794)

[9.1.4 Austausch 14](#_Toc287018795)

[9.1.5 Documetation Guidelines & Review 14](#_Toc287018796)

[9.2 Codequalität 15](#_Toc287018797)

[9.2.1 Codereview 15](#_Toc287018798)

[9.2.2 Styleguide für Code 15](#_Toc287018799)

[9.3 Tests 15](#_Toc287018800)

# Einführung

## Zweck

Dieses Dokument beschreibt den Projektplan für das Projekt MRT (Mobile Reporting Tool).

## Gültigkeitsbereich

Dieses Dokument gilt als Grundlage des Projektes und ist daher über die gesamte Projektdauer gültig.

## Definitionen und Abkürzungen

Siehe Glossar

## Referenzen

doc/00\_Projektantrag/projektantrag\_mrt.docx  
doc/01\_Projektplan/risikomanagement.xlsx  
doc/01\_Projektplan/glossar.docx  
doc/templates/template.dotx  
doc/media/logo.png

## Übersicht

Im Abschnitt „3. Projekt Übersicht“ wird das Projekt beschrieben und dessen Ziel und Zweck erläutert.

Darauf folgt der Abschnitt “4. Projektorganisation”, in dem näher auf die Organisationsstruktur und externe Schnittstellen eingegangen wird.

Die Planungsübersicht des Projektes wird im Abschnitt “5. Management Abläufe” detailliert dargestellt.

Der Punkt 6. Risiko Management wird in einem separaten Dokument aufgezeigt.

Sämtliche Arbeitspakete des Projektes werden im Abschnitt “7. Arbeitspakete” aufgelistet und beschrieben.

Im Abschnitt “8. Infrastruktur” werden die genutzten Räume, Geräte und Softwareprodukte aufgezeigt.

Abschliessend wird die Qualitätssicherung im Abschnitt “9. Qualitätsmassnahmen” genauer erläutert.

# Projekt Übersicht

## MRT (Mobile Reporting Tool)

Wie der Name „Mobile Reporting Tool“ bereits erahnen lässt, handelt es sich bei dem Produkt um ein Werkzeug, womit Aussendienstmitarbeiter auf unkomplizierte Art und Weiseihre Stundenrapportieren können und zwar unabhängig von ihrem Standort.

### Ablauf

#### Arbeitseinsatz

Der Arbeiter erhält von der Zentrale oder direkt von einem Kunden einen Auftrag. Sobald er bei dieser Adresse angelangt ist, drückt er die „Start“-Schaltfläche. Nun wird ihm eine Reihe von potentiellen, bereits registrierten Kunden vorgeschlagen. Der Arbeiter kann den passenden Kunden auswählen oder diesen Schritt überspringen. Durch das Drücken der „Start“-Schaltfläche beginnt die Zeitmessung.  
Sobald die Arbeit vollbracht ist, teilt der Arbeiter dies dem System durch das Drücken der „Stopp“-Schaltfläche mit. Um die spätere Verwaltung zu erleichtern, kann der Mitarbeiter zudem eine Beschreibung erfassen, dies ist optional.  
Zudem werden bei jedem Drücken der Schaltfläche die GPS-Daten an den Server übermittelt. So kann festgestellt werden, wo sich der Arbeiter aufgehalten hat.

#### Verwaltung & Auswertungen

Die Verwaltung der Kundendaten bzw. Mitarbeiter wird über einen dedizierten Server durchgeführt, auf den alle Aussendienstmitarbeiter sowie Büroangestellten über ein Web-Login Zugriff haben. Die für den Zugriff autorisierten Personen haben unterschiedliche Rechte, abhängig von deren Tätigkeiten.  
Es können neue Kunden bzw. Mitarbeiter erfasst, sowie diverse Auswertungen wie z.B. Rapporte, Mitarbeiterlisten etc. generiert werden

### Optionale Features

* Kunden direkt auf dem Android erfassen
* Kundenvorschläge anhand von GPS-Koordinaten
* Zusätzliche Auswertungen (z.B. um automatisch Rechnungen zu erzeugen)
* Assoziieren von Kundendaten mit GPS-Koordinaten

## Zweck und Ziel

Im Modul Software Engineering 1 in der Durchführung HS 2010/2011haben sich alle Teammitglieder fundamentales theoretisches Wissen im Bereich Software Engineering angeeignet. Nun soll dieses Wissen durch das Projekt MRT vertieft werden.

Ein weiteres Ziel ist, mittels geeigneten Werkzeugen wie Redmine, Subversion und Skype gut in der Gruppe zu kooperieren. Denn eine klare und effiziente Kommunikation ist die Grundlage eines jeden erfolgreichen Projektes.

Für das Projekt steht vergleichsweise wenig Zeit zur Verfügung. Aus diesem Grund wird das Produkt nach Abschluss des Projektes noch zu wenig ausgereift sein, um im produktiven Arbeitsumfeld ein­gesetzt werden zu können. Das Augenmerk wird darauf gerichtet, eine saubere und erweiterbare Basis zu entwickeln, welche zu einem späteren Zeitpunkt erweitert werden könnte. Es soll die Möglichkeit bestehen, das Projekt flexibel auszubauen, ohne auf entwicklungsbedingte Grenzen zu stossen.

## Annahmen und Einschränkungen

Für dieses Projekt werden keine Annahmen oder Einschränkungen getroffen.

# Projektorganisation

Das Projektteam besteht aus fünf sich gleichgestellten Mitgliedern. In der „Construction“-Phase werden zwei Unterteams gebildet, welche sich jeweils auf einen Tier konzentrieren und das Code-Review für das jeweils andere Team übernehmen.

## Team

### Lukas Elmer (Abk. EL)

Kenntnisse in: Ruby on Rails, PHP, Python / Django, Typo3, Wordpress, Java, XHTML,  
JavaScript, C++, Ubuntu Server

Motivation: Weitere Erfahrungen im Software-Engineering sammeln.  
SE1 Kenntnisse auf reales Projekt anwenden.

Spezialisierung: Android-Client

Verantwortlichkeiten: Coaching bei komplexen Problemen bei Ruby on Rails. Serverunterhalt  
von Redmine. Mithilfe bei Deployment.

Mailadresse : lelmer@hsr.ch

### Christina Heidt (Abk. HC)

Kenntnisse in: Java, HTML/CSS, C++, Photoshop

Motivation: Grundlagen von SE1 auf Projekt anwenden und Wissen vertiefen

Spezialisierung: Webplattform

Verantwortlichkeiten: Grafisches Design, Sitzungsprotokollierung, Teamleitung

Mailadresse: cheidt@hsr.ch

### Diego Steiner (Abk. SD)

Kenntnisse in: PHP, XHTML, ASP.net, Linux

Motivation: vorhanden

Spezialisierung: Webplattform

Verantwortlichkeit: Anforderungs- und Featurespezifikation

Mailadresse: dsteiner@hsr.ch

### Delia Treichler (Abk. TD)

Kenntnisse in: Java, HTML/CSS, C++

Motivation: Anwendung des theoretischen Wissens aus dem Modul SE1

Spezialisierung: Android Client

Verantwortlichkeit: Funktionale und nicht funktionale Anforderungen

Mailadresse: dtreichler@hsr.ch

### Remo Waltenspül (Abk. WR)

Kenntnisse in: Java, HTML/CSS, VB, C++

Motivation Erfahrungen in einem Software-Engineering Projekt sammeln

Spezialisierung: Webplattform

Verantwortlichkeit: technisches Design

Mailadresse: rwaltens@hsr.ch

## Anmerkung

Bei der Spezialisierung handelt es sich um die Unterteilung des Teams während der „Construction“-Phase.

## Organisationsstruktur



## Externe Schnittstellen

Für die Beratung und Benotung ist Prof. Hans Rudin zuständig. Als zusätzlicher Berater steht Daniel Keller zur Verfügung.

# Management Abläufe

Das folgende Kapitel erläutert kurz die essentiellen Management Abläufe. Dies beinhaltet den Projekt Kostenvoranschlag mit den Rahmenbedingungen sowie den Projektplan, welcher die zeitliche Aufteilung des Projekts in diverse Arbeitsschritte enthält.

## Projekt Kostenvoranschlag

Der Projektstart ist am 24. Februar 2011. Während 14 Wochen (exklusive einer Woche Ferien) wird am Projekt gearbeitet. Die Abgabe findet in der 22. Kalenderwoche statt, spätestens am 3. Juni 2011.

Von jedem Mitarbeiter wird eine Mindestarbeitszeit von acht Stunden erwartet, wobei diese acht Stunden bereits die vier offiziellen Lektionen enthalten. Falls zusätzliche nicht geplante Schwierigkeiten auftauchen, kann das Arbeitspensum pro Mitarbeiter auf höchstens 10-12 Stunden pro Woche erhöht werden.

Der Projektumfang erlaubt es uns, die vorgesehene Zeit vollumfänglich auszunutzen. Sofern die Kernfunktionalitäten schneller als geplant fertig gestellt werden können, wird das Projekt um zusätzlich vorerst als optional definierte Module ergänzt.

Es sind gegenwärtig keine geplanten Absenzen von Projektmitarbeitern bekannt, gegebenenfalls werden Abwesenheiten im Projektplan einkalkuliert.

## Projektplan

### Zeitplan

Die Zeiterfassung sowie der komplette Zeitplan werden vollumfänglich über einen dedizierten Redmine-Server durchgeführt. Dazu werden in einem ersten Schritt so genannte Tickets erfasst, welche vordefinierten Milestones zugewiesen werden. Diese Tickets entsprechen den Arbeitspaketen, die in einem nachfolgenden Abschnitt detaillierter spezifiziert werden.

Anhand der erstellten Tickets wird automatisch ein Zeitplan (Gantt-Diagramm) generiert, in dem alle Arbeitspakete übersichtlich dargestellt sind.

Für die einzelnen Reviews an den festgelegten Milestones werden Auszüge (Dumps) aus dem Redmine erstellt und anschliessend direkt in Excel formatiert. Mithilfe einer Excel-Vorlage können aus diesen generierten Daten automatisch Diagramme erzeugt werden, welche den Arbeitsaufwand der einzelnen Mitarbeiter optisch ansprechend anzeigen.

Zudem kann man zu jeder Zeit direkt im Redmine den Verlauf der verschiedenen Tickets bzw. Arbeitspakete verfolgen. Um dies zu ermöglichen, wird die Arbeitszeit von den Projektmitarbeiter direkt auf ein bestimmtes Ticket gebucht.

Ferner ist das Versionsverwaltungssystem SVN direkt mit Redmine verbunden und zeigt Änderungen zu einem Ticket in einer History an.

Der gesamte Zeitplan mit allen Arbeitspaketen wird über die Projektmanagementsoftware Redmine verwaltet. Sie ist für alle Projektmitglieder unter <https://redmine.elmermx.ch> erreichbar.  
Jeden Abend werden die am Tag gearbeiteten Stunden am Projekt auf das zugehörige Ticket gebucht.

### Iterationsplanung / Milestones

Siehe zeitplan\_gantt.pdf

#### Inception 21.02.-06.03.11 / SW01-SW02

|  |  |
| --- | --- |
| Inception I1 | Milestones |
| * Fertigstellung und Abgabe Projektantrag * Fertigstellung Projektplan * Erarbeitung Codestyleguide und Glossar | 25.02.11: Abgabe Projektantrag  04.03.11: MS1 - Abgabe Projektplan |

#### Elaboration 07.03.-17.04.11 / SW03-SW08

|  |  |
| --- | --- |
| Elaboration E1 / 07.03.-27.03.11 | Milestones |
| * Fertigstellung Use Cases * Fertigstellung DM, SSD, OCs * Usability Essentials ausarbeiten | 25.02.11: Abgabe Projektantrag  27.03.11: MS2 - Abgabe Analyse |

|  |  |
| --- | --- |
| Elaboration E2 / 28.03.-17.04.11 | Milestones |
| * Überarbeitung Use Cases * Überarbeitung DM, SSD, OCs * Usability Konzept getestet | 17.04.11: MS3 - Review: Elaboration |

#### Construction 18.04.-15.05.11 / SW09-SW13

|  |  |
| --- | --- |
| Construction C1 / 18.04.-15.05.11 | Milestones |
| * Version 0.3 releasen * Tests der Version 0.3 durchführen | 15.05.11: MS4 - Review: Construction |

#### Transition 30.05.-03.06.11 / SW14

|  |  |
| --- | --- |
| Transition T1 | Milestones |
| * Version 1.0 ausliefern * Benutzerhandbuch fertigstellen * Präsentation fertigstellen | 02.06.11 MS5 - Schlusspräsentation/Abgabe |

### Besprechungen

Teambesprechungen werden zweimal wöchentlich durchgeführt:

* Montags, 12:30 bis 13:00, kurze Besprechung
* Donnerstags, umfassende Besprechung am Nachmittag (2 Stunden)

Ort: in einem zur Verfügung stehenden Zimmer an der HSR  
Teilnehmer: EL, HC, SD, TD, WR  
Die Meetings dienen zum Austausch in der Gruppe, zur Besprechung anstehender Arbeiten, deren Koordinierung im Team und zum Erarbeiten von Lösungsansätzen für angefallene Probleme.  
Die Besprechungen am Donnerstag beinhalten auch die Meilenstein Reviews und weitere Besprechungen mit dem Betreuer. Zu jeder Besprechung wird ein Sitzungsprotokoll geführt (siehe Sitzungsprotokolle).

### Abgabe

Folgende Versionen sind vorgesehen:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Typ | Beschreibung | Datum |
| 0.1 | Prototyp | Basissystem, Datenstrukturen | 17.04.11 |
| 0.2 | Testversion | Businesslogic, UI, Schnittstelle | 29.04.11 |
| 0.3 | Releasecandidate | Bugs gefixt | 15.05.11 |
| 1.0 | Finalversion | Fertige Version | 02.06.11 |

# Risiko Management

Da das Risiko Management komplex ist, wurde es in ein separates Dokument ausgelagert. Sämtliche Angaben zu möglichen Risiken, die während dem Projekt auftauchen könnten, werden in diesem Dokument detailliert analysiert und erläutert.

Um dynamisch neue Risiken zu erfassen bzw. als beseitigt zu definieren, existiert ein spezielles Arbeitsblatt. Allfällige technische implementationsabhängige Risiken werden als Ticket imRedmine erfasst und sobald wie möglich geklärt bzw. beseitigt.

# Arbeitspakete

Die Beziehung zwischen den einzelnen Arbeitspaketen und den zugehörigen Iterationen ist im Redmine ersichtlich. Die unten dargestellte Tabelle dient nur zum Überblick, folge dessen werden die Pakete komplett über den Redmine-Server verwaltet.

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeitspaket | Inhalt | Abhängigkeiten | Risiken/Probleme |
| Projekt Management |  |  |  |
| Redmine Projekt | Projekt im Redmine aufsetzen |  |  |
| Projektantrag | Konzept in Antrag formulieren |  |  |
| Projektplan | Detaillierter Projektplan erstellen | Iterationsplanung |  |
| Logo | Logo Erstellung für Dokumentvorlagen |  |  |
| Dokumentvorlage | Einheitliche Dokumentvorlagen | Logo |  |
| Codestyleguide | Styleguide für Code definieren |  |  |
| Iterationsplanung | Planung der einzelnen Phasen |  |  |
| Administratives | Zeit rapportieren, Arbeitsumgebung einrichten |  |  |
| Business Modeling |  |  |  |
| DM | Domain Model | Use Cases |  |
| SSD | Systemsequenzdiagramm | Use Cases |  |
| Operation Contracts | Operation Contracts | Domain Model, SSD |  |
| Requirements |  |  |  |
| UC Mitarbeiter Rapport | Usecase des Aussendienstmitarbeiters |  |  |
| UC Rapport | Usecase des Sekretärs |  |  |
| UC Verwaltung | Usercase des Managers |  |  |
| Vision | Vision dokumentieren |  |  |
| Glossar | Glossar |  |  |
| Analyse und Design |  |  |  |
| Paperprototyping | GUI Papierprototyp erarbeiten |  |  |
| Interaktionsdiagramme | Interaktionsdiagramme |  |  |
| Klassendiagramme | Klassendiagramme | Domainmodel |  |
| Implementation |  |  |  |
| Android Client | Implementation des Android Clients | KDs, IDs |  |
| Reporting | Implementation des Reportings aus den gesammelten Daten | KDs, IDs, |  |
| Backoffice Server | Implementation des Serverteils der die Schnittstelle für Android bietet | KDs, IDs |  |
| Tests |  |  |  |
| Systemtests | Testcases für das gesamte System | Use Cases |  |
| Unit Tests | Testcases für einzelne Klassen | Use Cases |  |
| Bugfixing | Fehler die aus dem Test hervorgingen beheben | Systemtests, Unit Tests |  |
| Usability Tests | Testcases mit Benutzern | Paperprototyping |  |
| Deployment |  |  |  |
| Deployment auf Testumgebung | Release auf Testumgebung deployen |  |  |
| Deployment auf Liveumgebung | Release auf Liveumgebung deployen |  |  |
| Dokumentation |  |  |  |
| Sitzungsprotokolle | Protokolle zu den Sitzungen | Sitzungen |  |
| Bedienungsanleitung | Bedienungsanleitung |  |  |
| Schlusspräsentation | Schlusspräsentation |  |  |
| Sitzungen |  |  |  |
| Sitzungen | Sitzungen |  |  |
| Qualitätssicherung |  |  |  |
| Qualitätsmassnahmen | Qualitätsgrundlagen definieren |  |  |
| Code Review | Code reviewen | Implementation |  |
| Review Projektplan | Projektplan reviewen | Projektplan |  |
| Review Dokumente | Dokumente reviewen | Dokumente |  |
| Studium Technologien |  |  |  |
| Fachliteratur | Aneignung neuer Programmiersprachen |  |  |

# Infrastruktur

Für Besprechungen und gemeinsame Arbeiten werden die verfügbaren Räume der HSR genutzt. Die Projektmitglieder arbeiten bevorzugt mit den persönlichen Notebooks (ausgestattet mit Windows 7 und Ubuntu Linux). Bei Ausfall eines dieser Geräte kann gegebenenfalls einer der HSR-Arbeitsrechner verwendet werden. Zur Projektverwaltung wird der persönliche SVN-Server von Lukas Elmer genutzt. Zudem stehen für das Testen der mobilen Applikation zwei Android Mobiltelefone zur Verfügung.

## Software

Folgende Software kommt zum Einsatz:

|  |  |  |
| --- | --- | --- |
| Entwicklung | Projektmanagement | Dokumentation |
| * Eclipse IDE, Version 3.6.2 * NetBeans IDE, Version 6.9.1 * Subversion, Version 1.6.12 * Android SDK, Version 3.0 * Ruby on Rails, Version 3.0 | * Redmine, Version 1.1.1 | * MS Office (ab 2007) * Enterprise Architect, Version 7.5 * Google Docs |

## Kommunikation

Für die Kommunikation zwischen den Projektmitarbeiter werden folgende Hilfsmittel verwendet:

* Wiki (<https://redmine.elmermx.ch/projects/mrt/wiki>)
* E-Mail
* Skype
* Meetings

# Qualitätsmassnahmen

## Allgemein

### Regelmässige Teamsitzungen &teamfördernde Massnahmen

Neben den regelmässigen Teamsitzungen (siehe Abschnitt 5.2.3 Besprechungen) nimmt das gesamte Team einmal pro Monat an einem gemeinsamen Nachtessen Teil. Dabei wird das Projekt auf einer lockeren Basis diskutiert und der Teamgeist gefördert.

### Sourcecode Management

Der gesamte Sourcecode wird von Subversion verwaltet. Diese Software hat sich in einem Projekt während des letztens Semesters bewährt, somit wird auf eine solide Basis gebaut. Das Repository ist mit dem Redmine verlinkt, sodass direkt auf Issues committed werden kann.

### Issuetracking

Als Issuetracker wird Redmine auf <https://redmine.elmermx.ch>verwendet. Redmine bietet ein ausgereiftes Bug- und Issuemanagement sowie eine SVN Integration und ausgereiftes Reporting.

### Austausch

Als Plattform für Fragen und zum Austausch nutzen wir das bereits vorhandene Wiki von Redmine (<https://redmine.elmermx.ch/projects/mrt/wiki>). Die Ergebnisse der Sitzungen werden jeweils in den Sitzungsprotokollen festgehalten.

### Documetation Guidelines & Review

Alle im Rahmen des Projektes erzeugten Dokumente werden auf der Basis eines Templates erstellt. Dadurch werden den über alle dem Projekt zugehörigen Dokumenten eine gewisse Struktur und damit eine Konsistenz gegeben.  
Bevor ein Dokument als Final deklariert werden kann, muss es von mindestens zwei weiteren Instanzen gegengelesen und freigegeben werden.

## Codequalität

### Codereview

Mindestens einmal pro Iteration findet ein Codereview durch das jeweils andere Team statt. Damit behalten alle den Überblick über das gesamte Projekt und der Code wurde von mindestens zwei anderen Instanzen überprüft.

### Styleguide für Code

Alle Entwickler haben sich beim Schreiben von Code an die in dem Styleguide für Sourcecode definierten Richtlinien zu halten.

* Projektautomation
* Im Android-Teil sorgt ein Ant-Skript für regelmässiges kompilieren und automatisiertes Testen des Projektes.
* Für den Rails-Teil sind keine automatischen Builds vorgesehen, da Ruby eine Skriptsprache ist und dementsprechend nicht kompiliert werden muss. Jedoch werden bei jedem Commit automatisierte Tests ausgelöst.

Der Einsatz eines Hudson-Servers hat sich nach einer Teambesprechung als nicht sinnvoll erwiesen.

## Tests

#### Unit Tests

Für alle wichtigen Klassen und vor allem für die Kernarchitektur werden J- &Rails- Unit Tests geschrieben. Dies gewährleistet eine hohe Qualität der Kernelemente. Die Tests werden in die Projektautomation miteinbezogen.

#### Usability Tests

Die Bedienbarkeit wird von verschiedenen Personen getestet und vom Team dokumentiert. Dadurch wird eine Steigerung der Usability und eine grössere Benutzerakzeptanz erwartet. Im Vorfeld werden auch Paper-Prototyping Methoden eingesetzt.

#### Systemtests

Jeder Release wird einem vollständigen Systemtest unterzogen. Dazu werden Testspezifikationen geschrieben, um den Test nachvollziehbar zu machen. Testprotokolle werden erstellt und überprüft.