|  |
| --- |
| 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 |

## Inhaltsverzeichnis

[1 Dokumentinformationen 1](#_Toc286936081)

[1.1 Änderungsgeschichte 1](#_Toc286936082)

[2 Einführung 2](#_Toc286936084)

[2.1 Zweck 2](#_Toc286936085)

[2.2 Gültigkeitsbereich 2](#_Toc286936086)

[2.3 Definitionen und Abkürzungen 2](#_Toc286936087)

[2.4 Referenzen 2](#_Toc286936088)

[2.5 Übersicht 2](#_Toc286936089)

[3 Projekt Übersicht 4](#_Toc286936090)

[3.1 MRT (Mobile Reporting Tool) 4](#_Toc286936091)

[3.2 Zweck und Ziel 4](#_Toc286936092)

[3.3 Annahmen und Einschränkungen 4](#_Toc286936093)

[4 Projektorganisation 5](#_Toc286936094)

[4.1 Team 5](#_Toc286936095)

[4.1.1 Lukas Elmer (abk. LE) 5](#_Toc286936096)

[4.1.2 Christina Heidt (abk. CH) 5](#_Toc286936097)

[4.1.3 Delia Treichler (abk. DT) 5](#_Toc286936098)

[4.1.4 Remo Waltenspül (abk. RW) 5](#_Toc286936099)

[4.1.5 Diego Steiner (abk. DS) 5](#_Toc286936100)

[4.2 Anmerkung 5](#_Toc286936101)

[4.3 Organisationsstruktur 6](#_Toc286936102)

[4.4 Externe Schnittstellen 6](#_Toc286936103)

[5 Management Abläufe 6](#_Toc286936104)

[5.1 Projekt Kostenvoranschlag 6](#_Toc286936105)

[5.2 Projektplan 6](#_Toc286936106)

[5.2.1 Zeitplan 6](#_Toc286936107)

[5.2.2 Iterationsplanung / Meilensteine 7](#_Toc286936108)

[5.2.2.1 Inception 7](#_Toc286936109)

[5.2.3 Besprechungen 8](#_Toc286936110)

[5.2.4 Abgabe 8](#_Toc286936111)

[6 Risiko Management 8](#_Toc286936112)

[7 Arbeitspakete 8](#_Toc286936113)

[8 Infrastruktur 10](#_Toc286936114)

[9 Qualitätsmassnahmen 10](#_Toc286936115)

[9.1 Allgemein 10](#_Toc286936116)

[9.1.1 Regelmässige Teamsitzungen & Teamfördernde Massnahmen 10](#_Toc286936117)

[9.1.2 Sourcecode Management 10](#_Toc286936118)

[9.1.3 Issuetracking 10](#_Toc286936119)

[9.1.4 Austausch 10](#_Toc286936120)

[9.1.5 Documetation Guidelines & Review 12](#_Toc286936121)

[9.2 Codequalität 12](#_Toc286936122)

[9.2.1 Codereview 12](#_Toc286936123)

[9.2.2 Styleguide für Code 12](#_Toc286936124)

[9.3 Tests 12](#_Toc286936125)

# 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

usw.

## 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 Weise ihre Stundenrapportieren können und zwar unabhängig von ihrem Standort.

**<Detailliertere Beschreibung zum Ablauf!!!>**

## 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, eine saubere und erweiterbare Basis zu entwickeln, auf der man später aufbauen könnte. Also soll die Möglichkeit bestehen, das Projekt flexibel auszubauen, ohne auf entwicklungsbedingte Grenzen zu stossen.

## Annahmen und Einschränkungen

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 Projektmitarbeiter bekannt, gegebenenfalls werden Abwesenheiten im Projektplan einkalkuliert.

Der späteste Zeitpunkt für die Abgabe wurde auf den 03. Juni 2011 um 17:00 festgelegt, diese Deadline kann nicht verzögert werden.

# Projektorganisation

Das Projektteam besteht aus 5 sich gleichgestellten Mitgliedern. In der Entwicklungsphase werden 2 Unterteams gebildet, das sich jeweils auf ein Tier konzentriert und das Code-Review für das jeweils andere Team übernimmt.

## Team

### Lukas Elmer (abk. EL)

Kentnisse in: Ruby on Rails, PHP, Python / Django, Typo3, Wordpress,     Java,

XHTML,Javascript, C++, Ubuntu Server  
Motivation:

### Christina Heidt (abk. HC)

Kentnisse in:

Motivation   
Verantwortlichkeit:   
Mailadresse: cheidt@hsr.ch

### Delia Treichler (abk. TD)

Kentnisse in:

Motivation:   
Verantwortlichkeit:  
Mailadresse: dtreichler@hsr.ch

### Remo Waltenspül (abk. WR)

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

Motivation Erfahrungen in einem Software-Engineering Projekt sammeln  
Verantwortlichkeit:   
Mailadresse: rwaltens@hsr.ch

### Diego Steiner (abk. SD)

Kentnisse in: PHP, XHTML, ASP.net, Linux  
Motivation:

Verantwortlichkeit:   
Mailadresse: dsteiner@hsr.ch

## Anmerkung

Die detaillierten Kompetenzen der einzelnen Projektmitarbeiter sind im Abschnitt Arbeitspakete ersichtlich. Die oben genannten Verantwortlichkeiten sind nur die Kerngebiete während der Construction Phase.

## https://lh4.googleusercontent.com/_wmtCwD88ZKXxVFtXHM-Uqfaxg4SzGiWfZ0mF1qVwnfYmlGGStMkXeVEtoZvcqxEHh_aap2blCQY8Gk1uCGyrzj6end243o5VdbFAkSB7WH1ydFEYEwOrganisationsstruktur

## Externe Schnittstellen

Für die Beratung und Benotung ist Prof. Hans Rudin zuständig, als zusätzlicher Berater steht ausserdem 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.

Pro Woche werden pro Teammitglied durchschnittlich 8 Arbeitsstunden budgetiert. Insgesamt ergibt dies 560 Stunden für die Realisierung des Projektes. Reicht die Zeit aufgrund von unerwarteten Problemen nicht aus, so können  die Arbeitsstunden pro Woche und Kopf um maximal zwei Stunden auf 10 Stunden angehoben werden.

## 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 (Gant-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.

Unser gesamte Zeitplan mit allen Arbeitspacketen verwalten wir über die Projektmanagementsoftware Redmine, die für alle Projektmitglieder unter <https://redmine.elmermx.ch> erreichbar ist

### Iterationsplanung / Milestones

Siehe zeitplan\_gantt.pdf

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

|  |  |
| --- | --- |
| Inception I1 | Milestones |
| * Fertigstellung und AbgabeProjektantrag * Erarbeitung Projektplan * Erarbeitung Codestyleguide und Glossar | 25.02.11: Abgabe Projektantrag  03.03.11: Abgabe Projektplan |

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

|  |  |
| --- | --- |
| Elaboration E1 / 07.03.-27.03.11 | Milestones |
| * Inhalt ergänzen | 24.03.11 MS2 Review: Anforderungen und Analyse, 14:05, 45 min |
| Elaboration E2 / 28.03.-17.04.11 | Milestones |
| * Inhalt ergänzen | 14.04.11 MS3 Review:  Ende Elaboration, 14:05, 45 min |

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

|  |  |
| --- | --- |
| Construction C1 / 18.04.-15.05.11 | Milestones |
| * Inhalt ergänzen | 12.05.11 MS4 Review: Architektur/Design, 14:05, 45 min |
| Construction C2 / 16.05.-29.05.11 | Milestones |
| * Inhalt ergänzen |  |

#### Transition 30.05.-03.06.11 / SW14

|  |  |
| --- | --- |
| Transition T1 | Milestones |
| * Inhalt ergänzen | 02.06.11 MS5 Review: Schlusspräsentation/Abgabe, Termin noch nicht definitiv |

### 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 Ordner Sitzungsprotokolle).

### Abgabe

<Beschreibung verschiedener Releases. Was ist alles realisiert? Wann sind die Realeases fertig? Typ (Prototyp, Beta, Finalversion) des Releases?>

# Risiko Management

Da das Risiko Management komplex ist, wurde es in ein separates Dokument risikomanagement.xlsx 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 im Redmine erfasst und sobald wie möglich geklärt bzw. beseitigt.

# Arbeitspakete

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeitspaket | Inhalt | Abhängigkeiten | Risiken/Probleme |
| Projekt Management |  |  |  |
| Redmine Projekt aufsetzen |  |  |  |
| Projektantrag | Konzept in Antrag formulieren |  |  |
| Projektplan | Detaillierter Projektplan erstellen |  |  |
| Logo | Logo Erstellung für Dokumentvorlagen |  |  |
| Dokumentvorlage | Einheitliche Dokumentvorlagen | Logo |  |
| Codestyleguide | Styleguide für Code definieren |  |  |
| Iterationsplanung | Planung der einzelnen Phasen |  |  |
| Managementabläufe | Abschnitt Managementabläufe erarbeiten |  |  |
| Business Modeling |  |  |  |
| Domain Model |  | Use Cases |  |
| SSD | Systemsequenzdiagramm | Use Cases |  |
| Operation Contracts |  | Domain Model, SSD |  |
| Requirements |  |  |  |
| UC Mitarbeiter Rapport |  |  |  |
| UC Rapport |  |  |  |
| UC Verwaltung |  |  |  |
| Vision | Vision dokumentieren |  |  |
| Glossar |  |  |  |
| Analyse und Design |  |  |  |
| Paperprototyping | GUI Papierprototyp erarbeiten |  |  |
| Interaktionsdiagramme |  |  |  |
| Klassendiagramme |  |  |  |
| Implementation |  |  |  |
| Android Client |  |  |  |
|  |  |  |  |
|  |  |  |  |
| Tests |  |  |  |
| Systemtests |  |  |  |
| Unit Tests |  |  |  |
| Bugfixing |  | Systemtests, Unit Tests |  |
| Usability Tests |  |  |  |
| Dokumentation |  |  |  |
| Sitzungsprotokolle |  | Sitzungen |  |
| Bedienungsanleitung |  |  |  |
| Schlusspräsentation |  |  |  |
| Sitzungen |  |  |  |
| Sitzungen |  |  |  |
| Qualitätssicherung |  |  |  |
| Qualitätsmassnahmen | Qualitätsgrundlagen definieren |  |  |
| Code Review |  |  |  |
| Review Projektplan |  |  |  |
| Review Dokumente |  |  |  |
| Studium Technologien |  |  |  |
| Fachliteratur | Aneignung neuer Programmiersprachen |  |  |

|  |  |  |  |
| --- | --- | --- | --- |
| Arbeitspaket | Inhalt | Abhängigkeiten | Risiken /Probleme |
| Projekt Managment |  |  |  |
| Redmine Projekt aufsetzen |  |  |  |
| Projektantrag | Konzept in Antrag formulieren |  |  |
| Projektplan | Detailierter Projektplan erstellen |  |  |
| Dokumentvorlage | Einheitliche Dokumentvorlage |  |  |
| Glossar | Glossar erstellen | Dokumentvorlage |  |
| Fachliteratur | Fachliteratur zur Gameprogrammierung lesen |  |  |
| Risikomanagement | Risiken analysieren und Reserven abschätzen |  |  |
| Qualitätsmassnahmen | QoS Grundlagen definieren |  |  |
| Managementabläufe |  |  |  |
| Codestyleguide | Styleguide für Code definieren |  |  |
| Iterationspläne | Phasen definieren |  |  |
| Paperprototyping | Einfaches GUI Design erstellen |  |  |
| Persona | Personas erstellen |  |  |
| Vision | Vision aufschreiben |  |  |
| Checkliste prüfen | Projektfortschritt mit Checkliste abgleichen |  |  |
| Businessmodel |  |  |  |
| Domain Model |  | Use Cases |  |
| SSD | Systemsequenzdiagram | Use Cases |  |
| Operation Contracts |  | Domain Model, SSD |  |
| UC Mitarbeiter Rapport |  |  |  |
| UC Rapport Review |  |  |  |
| UC Verwaltung |  |  |  |
| Implementation |  |  |  |
|  |  |  |  |
| Tests |  |  |  |
| Systemtests | Test vordefinieren und durchführen |  |  |
| Unittests |  |  |  |
| Code Review |  |  |  |
| Bugfixing |  | Systemtests, Unit Tests |  |
| Usailitytests |  |  |  |
| Dokumentation und Deployment |  |  |  |
| Schlusspräsentation |  |  |  |
| Sitzungsprotokolle |  |  |  |
| Bedienungsanleitung |  |  |  |
| Reporting |  |  |  |

# Infrastruktur

## Räumlichkeiten

Für Besprechungen und gemeinsame Arbeiten werden die verfügbaren Räume der HSR genutzt.

## Hardware

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) nimmt das gesamte Team 1 Mal 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, da es sich in unserem Team schon etabliert hat und wir auf eine solide Basis bauen wollen. Das Repository ist direkt im Redmine verlinkt, sodass direkt auf Issuescommitted werden kann.

### Issuetracking

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

### Austausch

Als Plattform für Fragen und zum Austausch (mit dem Betreuer) 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 wird eine gewisse Struktur und damit eine Konsistenz über alle dem Projekt zugehörigen Dokumente gegeben.  
Bevor ein Dokument als Final deklariert werden kann, muss es von mindestens 2 Weiteren Instanzen gegengelesen und freigegeben werden.

## Codequalität

### Codereview

Mindestens 1 Mal 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 2 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, womit auch bei diesem Part

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 in die Projektautomation miteinbezogen.

#### Usability Tests

Die Bedienbarkeit wird von verschiedenen Personen getestet und von uns dokumentiert. Dadurch erhoffen wir uns eine Steigerung der Usability und eine grössere Benutzerakzeptanz. 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.