



E-Health Service Project Plan

Version 2.0

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

Revision History

Date	Version	Description	Author
2002-10-24	0.1	Initial Draft	E-Health Service Team
2012-11-02	1.0	First version of the document	E-Health Service Team
2012-12-10	1.1	Revision	Stefania Pezzetti
2013-01-20	2.0	Revised the entire document	Stefania Pezzetti

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

Table of Contents

1.	Introduction	4
1.1	Purpose of this document	4
1.2	Intended Audience	4
1.3	Scope	4
1.4	Definitions and acronyms	4
1.4.1	Definitions	4
1.4.2	Acronyms and abbreviations	4
1.5	References	4
2.	Background and Objectives	4
3.	Organization	6
3.1	Project group	6
3.2	Customer	7
3.3	Supervisor	7
3.4	Others	7
4.	Development process	8
5.	Deliverables	10
5.1.1	Remarks	10
6.	Inputs	11
6.1.1	Remarks	11
7.	Project risks	11
8.	Communication	12
8.1	Language	12
8.2	Tools	12
8.3	Organization	12
9.	Configuration management	12
10.	Project plan	13
10.1	Time schedule	13
10.1.1	Remarks	13
10.2	Activity plan	13
10.2.1	Remarks	14

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

1. Introduction

E-Health Service is a project for the Distributed Software Development course held by Politecnico di Milano (POLIMI) in Italy, Malardalen University (MDH) in Sweden and University of Zagreb (FER) in Croatia.

1.1 Purpose of this document

This is a formal, approved document used to guide both project execution and project control. The primary uses of the project plan are to document planning assumptions and decisions, facilitate communication among stakeholders and document schedule baselines. It's the first one of the project and should be continuously revised during the development.

1.2 Intended Audience

All project stakeholders are intended audience of this project: project customers, project supervisor, all team members and all people responsible for monitoring the project.

1.3 Scope

This document gives a brief overview of the project which includes details about organization, roles, deliverables, project risks and time plans. It does not include requirements analysis and system design description, which would appear in future dedicated documents.

1.4 Definitions and acronyms

1.4.1 Definitions

Keyword	Definitions
<i>Global Team</i>	The whole workgroup
<i>Italian Team</i>	The Italian part of the team
<i>Croatian Team</i>	The Croatian part of the team

1.4.2 Acronyms and abbreviations

Acronym or abbreviation	Definitions
<i>POLIMI</i>	Politecnico di Milano
<i>FER</i>	University of Zagreb
<i>MDH</i>	Malardalen University

1.5 References

Main references to this project can be found on the DSD course official webpage:
http://www.fer.unizg.hr/rasip/dsd/projects/e-health_service

2. Background and Objectives

E-Health system supports doctors in everyday activities, such as the retrieval of mixed media information of their patients that combines text with or without files referring to their personal data, their medical histories and patient-related diseases.

The patients are allowed to retrieve information about the doctor expertise and they update some vital parameters (e.g. heart rate) thus their health is continuously monitored. On the basis of such data, doctors may send an alarm in case of warning conditions.

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

The system manages alarm notifications sent to first aid both by doctors and directly by patients.

The final product will be delivered as web application and as mobile application. Milestones and deliverables are reported below:

Every Monday (from 2012-10-29 to 2013-01-21)	<i>Summary Week Report</i>
2012-10-30	<i>Project Plan Presentation</i>
2012-11-02	<i>Project Plan Document</i>
2012-11-02	<i>Requirements Definition Document</i>
2012-11-06	<i>Requirements Definition and System Architecture Presentation</i>
2012-11-09	<i>Design Description Document</i>
2012-11-27	<i>Alpha Prototype Presentation</i>
2012-12-18	<i>Beta Prototype Presentation</i>
2012-12-31	<i>Acceptance Test Plan</i>
2013-01-14	<i>Test Report</i>
2013-01-15	<i>Final Project Presentation</i>
2013-01-20	<i>Final Project Reports and Documents</i>
2013-01-20	<i>Final Product</i>

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

3. Organization

The team is organized as presented above:

- ✓ Project leader (POLIMI): Stefania Pezzetti
- ✓ Team leader Italy (POLIMI): Stefania Pezzetti
- ✓ Team leader Croatia (FER): Marko Kelava

Local teams are daily management by the two team leaders: Stefania is responsible for the Italian team and in general for coordinating the entire work, while Marko is responsible for the Croatian team. For facilitating the communication between members, tools like Google Groups, Gmail, Skype, Subversion and Google Drive will be used.

3.1 Project group

Name	Initials	Responsibility	Roles
Stefania Pezzetti	SP	Project Leader Team Leader Requirements Manager	Web Developer
Marko Kelava	MK	Team Leader Database Manager	Business Logic Developer
Gregorio Perego	GP	Documentation Manager Web Manager	Web Developer
Bojan Kosanović	BK	System Architect	Business Logic Developer
Tomislav Tolj	TT	SVN Manager Server Manager	Business Logic Developer
Petar Kekez	PK	Test Manager	Mobile Developer
Vedran Šikić	VS	Mobile Manager	Mobile Developer

Each responsibility is performed by one person. This helps in having a clear understanding about the different fields in the project. The responsible person is involved in coordinating the necessary tasks and keeping track of the progress in his own domain.

Project Leader

The project leader keeps control of the overall development process and guides the team to the required targets. His goal is to keep the team on schedule and hold communication channels active.

Team Leader

The team leader is responsible for the daily management of his assigned team. There is a team leader for each geographical located team and his goal is to stay informed about his team's status and progress, and discuss future progress with other team leaders.

Documentation Manager

Documentation manager reviews all the written contents, and makes sure that documents are consistent and conform to the required format. A second duty is that this member coordinates the documentation of the implementation and testing activities.

Requirements Manager

Analyzing, tracing, prioritizing requirements, controlling their change and communicating them to relevant stakeholders. His goal is to broadcast requirements of the project to other members, so that they are aware of the context of the application.

Server Manager

Server Manager is in charge of the installed software on the team's server. He is also in charge of deployment of

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

any version of the application on the machine.

SVN Manager

SVN manager applies the SVN policy and makes sure that the SVN repository is kept clean and consistent. This Manager keeps also track of the backup processes, and verifies that these backups are usable. He is also in charge of any recovery processes. His goal is to keep the team able to work on their code, documents and files.

System Architect

System Architect designs the general architecture of the system. His goal is to create the architecture that can fulfill the demands of the requirements.

Database Manager

Database manager designs the supporting database model, and keeps control of the consistency of the database and any changes made to the database.

Test Manager

Test Manager coordinates all test cases, and is in charge of performing the tests, and reports the results of these tests to other team members.

Mobile Manager

Mobile Manager establishes design and interface of the mobile application.

Web Manager

Web Manager establishes design and interface of the web application.

3.2 Customer

The project customer is Raffaella Mirandola, who is also the supervisor. Customers for the final product could be doctors and medical divisions.

3.3 Supervisor

The project supervisor is Raffaella Mirandola, professor at POLIMI.

3.4 Others

Other people interested in the evolution of the project are all professors involved in DSD course, from POLIMI (Italy), MDH (Sweden) and FER (Croatia).

Also Andrea Ciancone from POLIMI, an expert in mobile development, is involved in the project as consultant for the implementation.

Finally Simone Pellegrino and Fabrizio Quadrio are involved in the project as medical science domain experts.

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

4. Development process

While the course schedule pushes us towards a classic waterfall model, we feel that this approach is outdated and not viable in any non-trivial system development, so we opted for something similar, but with more flexibility concerning unavoidable changes to implementation and design - RUP (Rational Unified Process). RUP is an iterative software development process framework created by a division of IBM. It employs a lifecycle of four phases:

1. Inception Phase

The primary objective is to scope the system adequately as a basis for validating initial costing and budgets. In this phase the business case which includes business context, success factors (expected revenue, market recognition, etc.), and financial forecast is established. To complement the business case, a basic use case model, project plan, initial risk assessment and project description (the core project requirements, constraints and key features) are generated. After these are completed, the project is checked against the many criteria.

If the project does not pass this milestone, called the Lifecycle Objective Milestone, it either can be cancelled or repeated after being redesigned to better meet the criteria.

2. Elaboration phase

The primary objective is to mitigate the key risk items identified by analysis up to the end of this phase. The elaboration phase is where the project starts to take shape. In this phase the problem domain analysis is made and the architecture of the project gets its basic form.

The outcome of the elaboration phase is:

- A use-case model in which the use-cases and the actors have been identified and most of the use-case descriptions are developed. The use-case model should be 80% complete.
- A description of the software architecture in a software system development process.
- An executable architecture that realizes architecturally significant use cases.
- Business case and risk list which are revised.
- A development plan for the overall project.
- Prototypes that demonstrably mitigate each identified technical risk.

This phase must pass the Lifecycle Architecture Milestone criteria. If the project cannot pass this milestone, there is still time for it to be cancelled or redesigned. However, after leaving this phase, the project transitions into a high-risk operation where changes are much more difficult and detrimental when made.

3. Construction phase

The primary objective is to build the software system. In this phase, the main focus is on the development of components and other features of the system. This is the phase when the bulk of the coding takes place. In larger projects, several construction iterations may be developed in an effort to divide the use cases into manageable segments that produce demonstrable prototypes.

This phase produces the first external release of the software. Its conclusion is marked by the Initial Operational Capability Milestone.

4. Transition phase

The primary objective is to 'transit' the system from development into production, making it available to and understood by the end user. The activities of this phase include training the end users and maintainers and beta testing the system to validate it against the end users' expectations. The product is also checked against the quality level set in the Inception phase.

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

If all objectives are met, the Product Release Milestone is reached and the development cycle is finished.

Six Best Practices

Six Best Practices as described in the Rational Unified Process is a paradigm in software engineering, that lists six ideas to follow when designing any software project to minimize faults and increase productivity. These practices are:

Develop iteratively

It is best to know all requirements in advance; however, often this is not the case. Several software development processes exist that deal with providing solution on how to minimize cost in terms of development phases.

Manage requirements

Always keep in mind the requirements set by users.

Use components

Breaking down an advanced project is not only suggested but in fact unavoidable. This promotes ability to test individual components before they are integrated into a larger system. Also, code reuse is a big plus and can be accomplished more easily through the use of object-oriented programming.

Model visually

Use diagrams to represent all major components, users, and their interaction. "UML", short for Unified Modeling Language, is one tool that can be used to make this task more feasible.

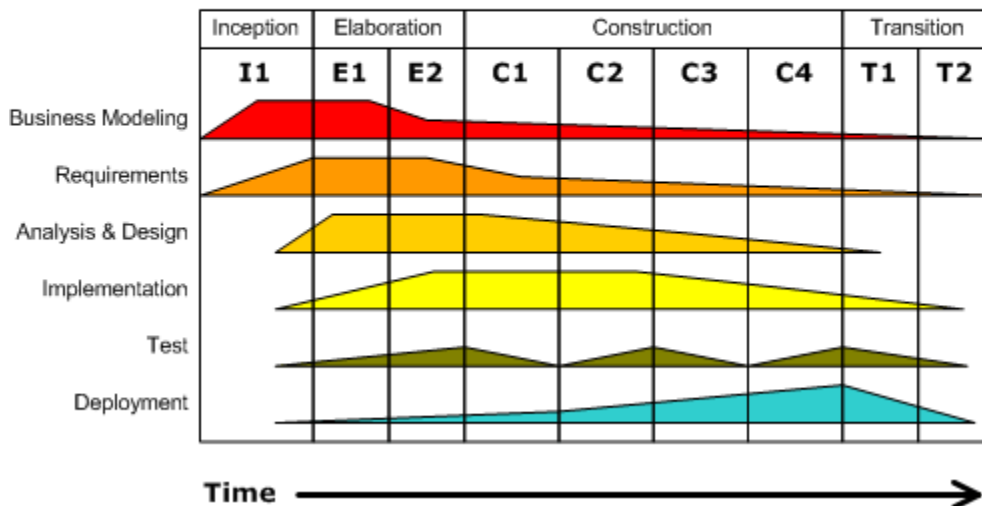
Verify quality

Always make testing a major part of the project at any point of time. Testing becomes heavier as the project progresses but should be a constant factor in any software product creation.

Control changes

Many projects are created by many teams, sometimes in various locations, different platforms may be used, etc. As a result it is essential to make sure that changes made to a system are synchronized and verified constantly.

Iterative Development
Business value is delivered incrementally in time-boxed cross-discipline iterations.



While we may not follow RUP to the point and we may adapt it to our needs, it is the general development model we will employ.

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

5. Deliverables

To	Output	Planned week	Promised week	Late +/-	Delivered week	Rem
Project Stakeholders Project Team members	<i>Project Plan</i>	44	44	0	44	
Project Stakeholders Project Team members	<i>Requirements Definition Document</i>	44	44	0	44	
Project Stakeholders Project Team members	<i>Design Description Document</i>	45	45	0	45	
Project Stakeholders Project Team members	<i>Acceptance test plan</i>	1	1	0	1	
Project Stakeholders Project Team members	<i>Test report</i>	3	3	0	3	
Project Stakeholders Project Team members	<i>Final Project Report</i>	3	3	0	3	
Project Stakeholders Project Team members	<i>Final versions of all documents</i>	3	3	0	3	
Project Stakeholders Project Team members	<i>Technical Documents</i>	3	3	0	3	
Project Stakeholders Project Team members	<i>User Manual</i>	3	3	0	3	
Project Stakeholders Project Team members	<i>Final Product</i>	3	3	0	3	

5.1.1 Remarks

Remark Id	Description

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

6. Inputs

From	Required item	Planned week	Promised week	Late +/-	Delivered week	Rem
Customer	Project Proposal	43	43	0	43	
DSD professors	SVN information	41	41	0	41	
DSD professors	Product evaluation	4	4			

6.1.1 Remarks

Remark Id	Description

7. Project risks

Possibility	Risk	Preventive action
<i>High</i>	Personal lack of time due to other courses or obligations	Weekly report to check status, arrange time in advance
<i>High</i>	Lack of technical expertise in the chosen implementation technologies	Cooperative learning within the team, and share good tutorials
<i>Medium</i>	Wrong interpretation of given requirements and boundaries	Have detailed use cases. Discuss to other team members if not clear about requirements.
<i>Medium</i>	Different understanding between team members	First discuss, then decide. Record decision in documents, for later reference.
<i>Medium</i>	Team Members are not reachable	Make sure that everyone knows what the other is doing so if someone doesn't complete his work another team member can complete that work.
<i>Low</i>	Member leaves	Have back up for each role. Have good documentation for work.
<i>Low</i>	Database Server Crash	Regular backup of database.
<i>Low</i>	SVN Server Crash	Regular backup at safe location

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

8. Communication

8.1 Language

English language will be official language among group members and when communicating with stakeholders.

8.2 Tools

The main tools that will be used for communication are the following:

- Asynchronous Communication
 - Google Groups
 - Gmail
 - Google Drive
 - Subversion
 - When is Good
- Synchronous Communication
 - Skype

8.3 Organization

The idea is to publish questions and proposals on Google Groups and then organize short meetings on Skype for getting all together important decisions about the project, using tools such as “When is Good” to establish the moment in which the meeting should take place. After every meeting one of the members has to write the “Minutes of Meeting” document and publish it on the official web site.

As regards Project Leader and Team Leaders, they have a role of mediation between other team leaders and team members of their country, so they have to be daily reachable.

9. Configuration management

Configuration management focuses on establishing and maintaining information assurance. SVN Subversion will be used for purpose of maintaining and sharing documentation. Of course we will need to do manually backups, in order to daily save a copy in local. Project code files will also be controlled and versioned with the official DSD SVN repository. One of SVN manager’s main responsibilities is creating SVN usage policy and management of SVN repository. Every member of the team has to follow the SVN project conventions.

FER provided us the following SVN repository:
<svn://lapis.rasip.fer.hr/svn/dsd12/HealthService>

Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

10. Project plan

10.1 Time schedule

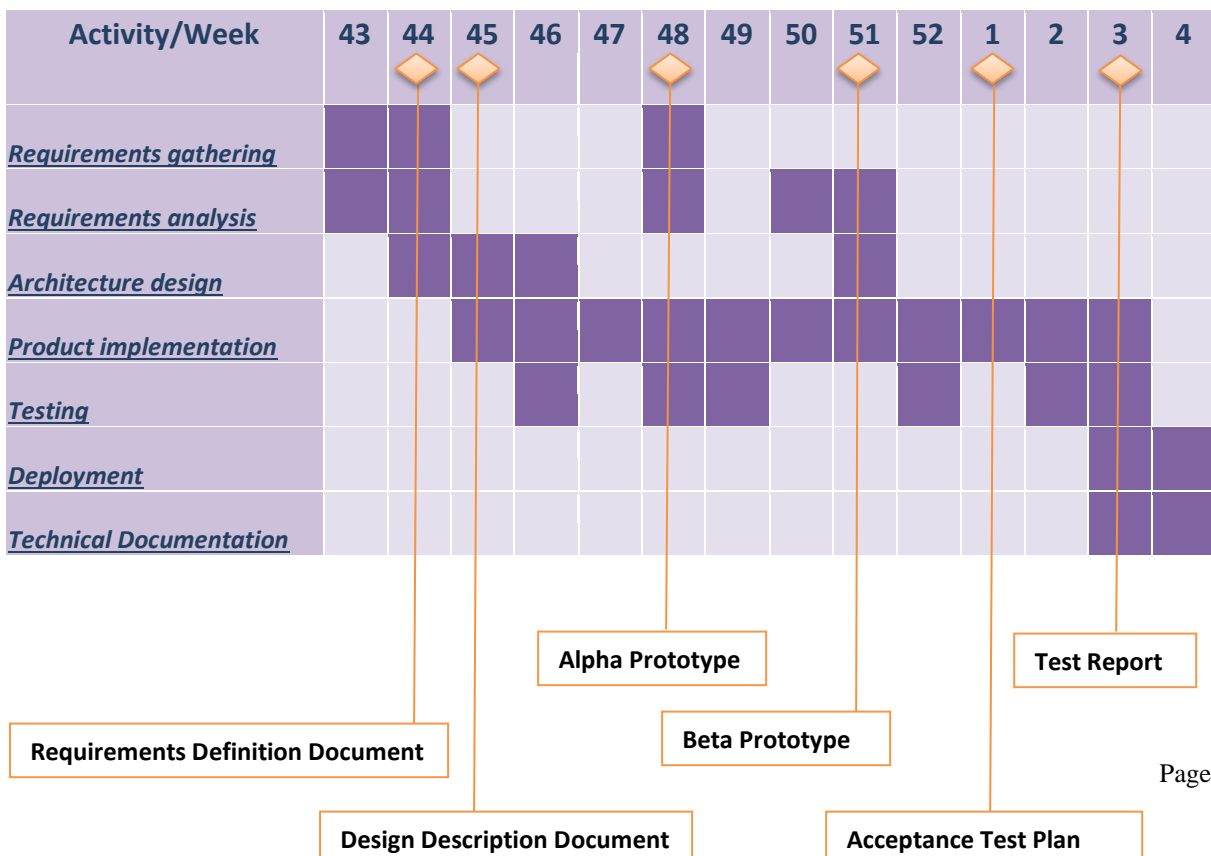
Id	Milestone Description	Responsible Dept./Initials	Finished week			Metr.	Rem.
			Plan	Forecast Week +/-	Actual		
M001	Project Plan	SP - GP	44		43		
	Requirement Engineering	MK - GP	44		44		
M002	Project Design	SP - MK	45		45		
M003	Policies	VS	46		46		
M004	Alpha prototype	SP - MK	48		48		
M005	Beta prototype	SP - MK	51		51		
M006	Acceptance Test Plan	GP	1		1		
M007	Test Report	MK	3		3		
M008	Installation Manual	SP - MK	3		3		
	User Manual	SP - MK	3		3		
	Final Product	SP - MK	3		3		
	Final Project Report	SP - MK	3		3		

10.1.1 Remarks

Remark Id	Description

10.2 Activity plan

Gantt Diagram:



Project Name	Version: 1.1
Project Plan	Date: 2013-01-20

ID	Predecessor	Activity	Days	Mdays	Rem.
1		Requirements gathering	2	6	rem.1
2	1	Requirements analysis	5	15	rem.1
3	1 - 2 - 4	Architecture design	5	15	rem.1
4	3	Product implementation	42	126	rem.1
5	4	Testing	1	3	rem.1
6	5	Deployment	2	6	rem.1
7	6	Technical Documentation	2	2	rem.1

Total planned calendar days for the project to be finished and total planned man-days required to finish the project are the following:

Planned effort (days)	Planned effort (man-days)
59	173

10.2.1 Remarks

Remark Id	Description
rem.1	We assume that each person works 3 hours a day on average.