Politecnico di Milano

Computer Science and Engineering's master degree course

Department of Electronics and Information

PowerEnjoy
Project Plan

Version 1.0

Release Date: 22/01/2017

Customer: Eng. Elisabetta DI NITTO

Authors:

Eng. Marco FERNI	Id. 877712
Eng. Angelo Claudio RE	Id. 877808
Eng. Gabriele TERMIGNONE	Id. 877645

Contents

Chapter 1

Introduction

1.1 Revision History

Version 1.0 - First Release Version

1.2 Purpose and Scope

The aim of this document is to plan the work of the PowerEnjoy project.

Planning is an hard task that is critical for the project life. An unrealistic estimation can lead the project to failure. A Project plan contains an estimation of the costs and efforts required to reach the project goals. Besides it studies how to organize the work in tasks and which resources should be allocated to them.

Last but not least there is a risk analysis, with a proactive approach to risks that can alter the work.

1.3 List of Definitions and Abbreviations

- *SE: Software Engineering*
- *PP:* Project Plan
- *PPD:* Project Plan Document
- *FP:* Function Points
- *PP:* Project Plan Document
- *ILF:* Internal Logic File
- *ELF:* External Logic File
- *EI:* External Input
- *EQ:* External Inquiries
- *EO:* External Output
- *SC:* Scale Factors
- *PREC:* Precedentedness
- *FLEX:* Development Flexibility
- *RESL:* Architecture / Risk Resolution
- *TEAM:* Team Cohesion
- *PMAT:* Process Maturity
- *RELY:* Required Software Reliability
- *DATA:* Data base size
- *CPLX:* Product Complexity

- *RUSE:* Required Reusability
- *DOCU:* Documentation match to life-cycle needs
- *TIME:* Execution Time Constraint
- *STOR:* Main Storage Constraint
- *PVOL:* Platform Volatility
- *ACAP:* Analyst Capability
- *PCAP:* Programmer Capability
- *APEX:* Application Experience
- *PLEX:* Platform Experience
- *LTEX:* Language and Tool Experience
- *PCON:* Personnel Continuity
- *TOOL:* Usage of Software Tools
- *SITE:* Multisite Development
- *SCED:* Required Development Schedule
- *HOW:* Hours of work mean per person

1.4 List of Reference Documents

- Assignment 4 of Assignments AA 2016-2017.pdf
- some PPD of last year
- SEII course's slides
- The COCOMO II Model Definition Manual (version 2.1, 1995 - 2000 Center for Software Engineering, USC).
- The Function Points complexity evaluation tables.
- Software Engineering - Principles and Practice Book (3rd Edition, Hans van Vliet, (c) Wiley, 2007)
- Software Engineering Book (9th edition, Ian Sommerville, Pearson, 2011)

Chapter 2

Estimate size, effort and cost

2.1 Function points

2.1.1 Internal Logic Files

An ILF is a homogeneous set of data used and managed by the application.

In our system, we can consider the database tables where we store persistent information as ILFs; the tables are the following:

- User
- Reservation
- Car
- Supervisor
- Safe area
- Safe park slot

ILF	**Complexity**	**FP**
User	AVG	10
Reservation	AVG	10
Car	AVG	10
Supervisor	LOW	7
Safe Area	AVG	10
Safe park slot	AVG	10

2.1.2 External Logic Files

An EIF is a homogeneous set of data used by the application but generated and maintained by other applications.
The only external application our system interact with is:

- Google Maps

ELF	Complexity	FP
Maps	HIGH	10

2.1.3 External Input

An EI is an elementary operation to elaborate data coming from the external environment.
In our system, we can consider as EIs the following:

- User Registration
- User Login
- Request for car's research
- Request for car's reservation
- Request for car's unlock
- Car's data (from sensors of physical cars)

EI	Complexity	FP
User Registration	AVG	4
User Login	LOW	3
Request for car's research	AVG	4
Request for car's reservation	LOW	3
Request for car's unlock	LOW	3
Car's data	HIGH	6

2.1.4 External Outputs

An EO is an elementary operation that generates data for the external environment and it usually includes the elaboration of data from logic files.
In our system, the EOs are:

- Confirmation email for registration
- Results of car research
- Lock/unlock commands to physical car
- Payment requests
- Confirmation email for payment

EO	Complexity	FP
User Registration	LOW	4
User Login	AVG	5
Request for car's research	LOW	4
Request for car's reservation	AVG	5
Request for car's unlock	LOW	4

2.1.5 External Inquiries

An EQ is an elementary operation that involves input and output, without significant elaboration of data from logic files.
In our system the EQs are:

- Notifications about reservation (to user and to physical car)
- Data requests from supervisors

EQ	Complexity	FP
Notifications	AVG	4
Data Request	LOW	3

2.1.6 Final Estimation

We can now calculate the total amount of FP of the system:

Function Type	**FP**
User Registration	4
User Login	5
Request for car's research	4
Request for car's reservation	5
Request for car's unlock	4
TOTAL	**119**

2.2 COCOMO

2.2.1 Introduction to COCOMO

According to the COCOMO II documentation, the multiplicator to convert the Function Points to Source Lines of Code for Java is 53, so we have:

$$\mathbf{SLOC = 53 \cdot 119 = 6307}$$

2.2.2 Scale driver

The exponent E is an aggregation of five SF that account for the relative economies or diseconomies of scale:

- **PREC:** it is **Low** because some of us have little experience of software design but most of the notions are new to us.
- **FLEX:** We set it to **Very High** because we have been given only the definition of the goals of the system, without non-functional or technological constraints.
- **RESL:** We set it to **Nominal**, because a risk analysis was made and exposed in chapter 5.
- **TEAM:** We set it to **High**.
- **PMAT:** We set it at **High**, which corresponds to CMM Level 3.

Then E is compute in the following way:

SF	**Factor**	**Value**
PREC	Low	4.96
FLEX	Very High	1.01
RESL	Nominal	4.24
TEAM	High	1.10
PMAT	High	1.56
	TOT	**12.87**

$$E = B + 0.01 \cdot \sum_{j=1}^{5} S \cdot F_j = 0.91 + 0.01 \cdot 12.87 = 1.0387$$

2.2.3 Cost driver

- *RELY:* We set it to **Low** because a failure won't have critical consequences.
- *DATA:* We set it to **Nominal** because we do not need extremely large amount of data to test the functionality of the system.
- *CPLX:* According with the COCOMO II CPLX rating scale, we set this cost driver to **High**.
- *RUSE:* We choose a **Low** value because we assume that we do not need to reuse components from this project.
- *DOCU:* We set it to **Nominal** because the documentation developed for the project is sufficient to cover the its life cycle.
- *TIME:* We set it to **Nominal**.
- *STOR:* We choose a **Nominal** value.
- *PVOL:* We set it to **Low**.
- *ACAP:* We choose a **Nominal** value because we are not considering our level of experience but only the quality of the design work.
- *PCAP:* We set it to **Nominal**.

- *APEX:* We choose a **Very Low** value for this cost driver because we are relatively new and unexperienced in this sector of applications.
- *PLEX:* We set this driver to a **Low** value due to our limited knowledge of the enterprise platforms.
- *LTEX:* We set it to **Nominal**.
- *PCON:* We obviously choose the **Very High** value because our team is fixed.
- *TOOL:* We set it to **Nominal**.
- *SITE:* We set this cost driver to **Nominal** because the team members live in different cities and the work was often done from home.
- *SCED:* We set it to **Nominal** because we managed to follow the given deadline correctly without rushing or procrastinate.

Cost Driver	**Factor**	**Value**
RELY	Low	0.92
DATA	Nominal	1.00
CPLX	High	1.17
RUSE	Low	0.95
DOCU	Nominal	1.00
TIME	Nominal	1.00
STOR	Nominal	1.00
PVOL	Low	0.87
ACAP	Nominal	1.00
PCAP	Nominal	1.00
APEX	Very low	1.22
PLEX	Low	1.09
LTEX	Nominal	1.00
PCON	Very High	0.81
TOOL	Nominal	1.00
SITE	Nominal	1.00
SCED	Nominal	1.00
	$\prod$EM	**0.9583**

2.2.4 Effort, duration and size of the team

We can now calculate the Person-Month estimation:

$$PM = A \cdot Size^{E} \cdot \prod_{i=1}^{n} EM_i =$$
$$= 2.94 \cdot 6.307^{1.0387} \cdot 0.9583 = 19.08 \text{ person-month}$$

2.2.5 Detailed report

The COCOMO II formula to estimate the project duration is:

$$TDEV = [C \cdot (PM_{NS})^{(D+0.2\cdot(E-B))}]$$

$$\text{where } C = 3.67, D = 0.28, B = 0.91$$

From the formula above, we obtain an estimated duration of **9.04** months with an optimal number of team member (2.11); since our team is composed by 3 person, a duration estimation could be: 19.08/3 = 6.36 months, which we can approximate to **7 months**.

Chapter 3

Tasks

In this chapter we're going to provide a general, high-level project schedule of the tasks.

In order to maintain readability, we have split the schedule in some parts, in according to a specific task.

We considered all real assignments except the code inspection, since it is not related to this application, with the real deadlines.

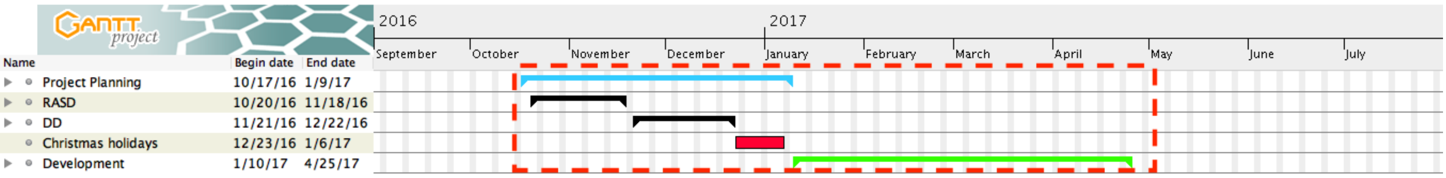
We used the duration provided by the COCOMO and a Gantt diagram to show the tasks with the deadlines.

NB

Every task timeline is highlighted with a red dashed rectangle.

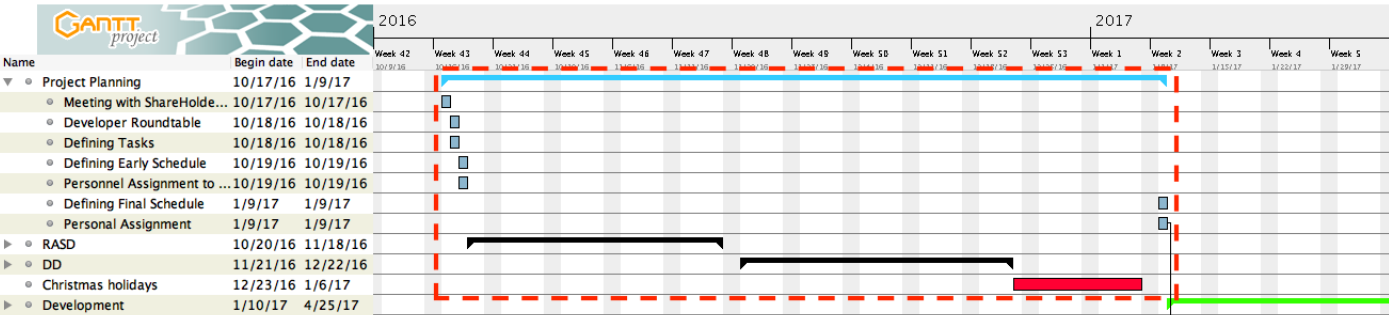
- The Slate Grey timeline in the diagram is common to all compoment of the project.
- The Turquoise timeline in the diagram is assigned to eng. Marco FERNI.
- The violet timeline in the diagram is assigned to eng. Angelo Claudio RE.
- The orange timeline in the diagram is assigned to eng. Gabriele Termignone.

3.1 General Overview



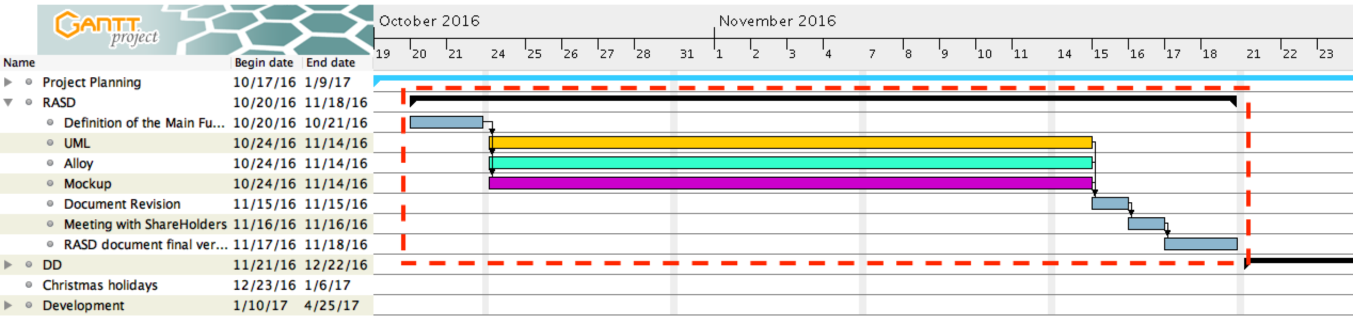
Name	Begin date	End date
Project Planning	10/17/16	1/9/17
RASD	10/20/16	11/18/16
DD	11/21/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17

3.2 Project Planning Task



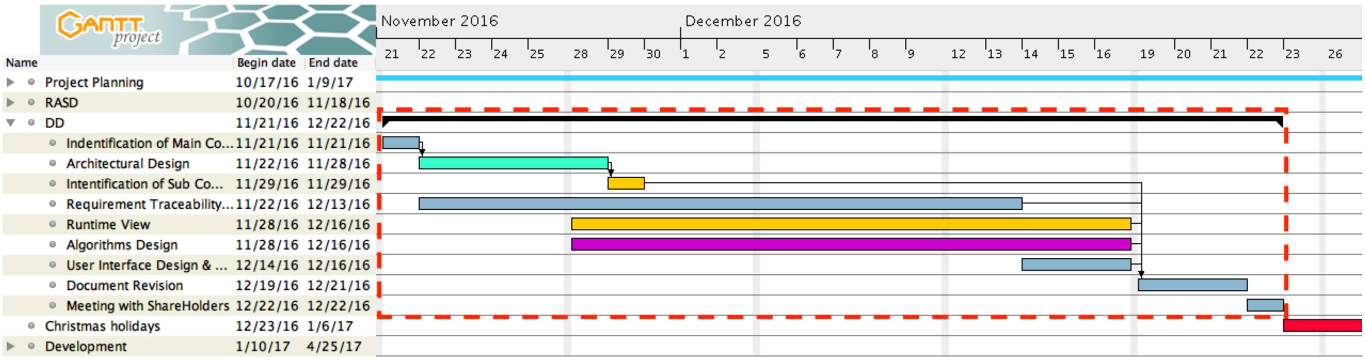
Name	Begin date	End date
Project Planning	10/17/16	1/9/17
Meeting with ShareHolde...	10/17/16	10/17/16
Developer Roundtable	10/18/16	10/18/16
Defining Tasks	10/18/16	10/18/16
Defining Early Schedule	10/19/16	10/19/16
Personnel Assignment to ...	10/19/16	10/19/16
Defining Final Schedule	1/9/17	1/9/17
Personal Assignment	1/9/17	1/9/17
RASD	10/20/16	11/18/16
DD	11/21/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17

3.3 Requirements and Specification Task



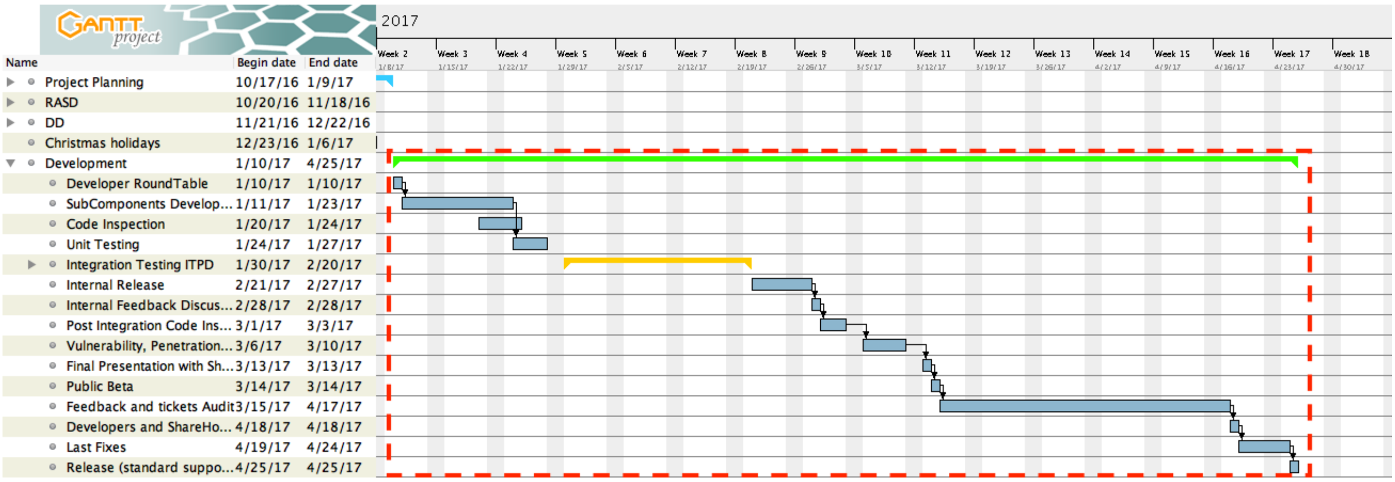
Name	Begin date	End date
Project Planning	10/17/16	1/9/17
RASD	10/20/16	11/18/16
Definition of the Main Fu...	10/20/16	10/21/16
UML	10/24/16	11/14/16
Alloy	10/24/16	11/14/16
Mockup	10/24/16	11/14/16
Document Revision	11/15/16	11/15/16
Meeting with ShareHolders	11/16/16	11/16/16
RASD document final ver...	11/17/16	11/18/16
DD	11/21/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17

3.4 Design Task



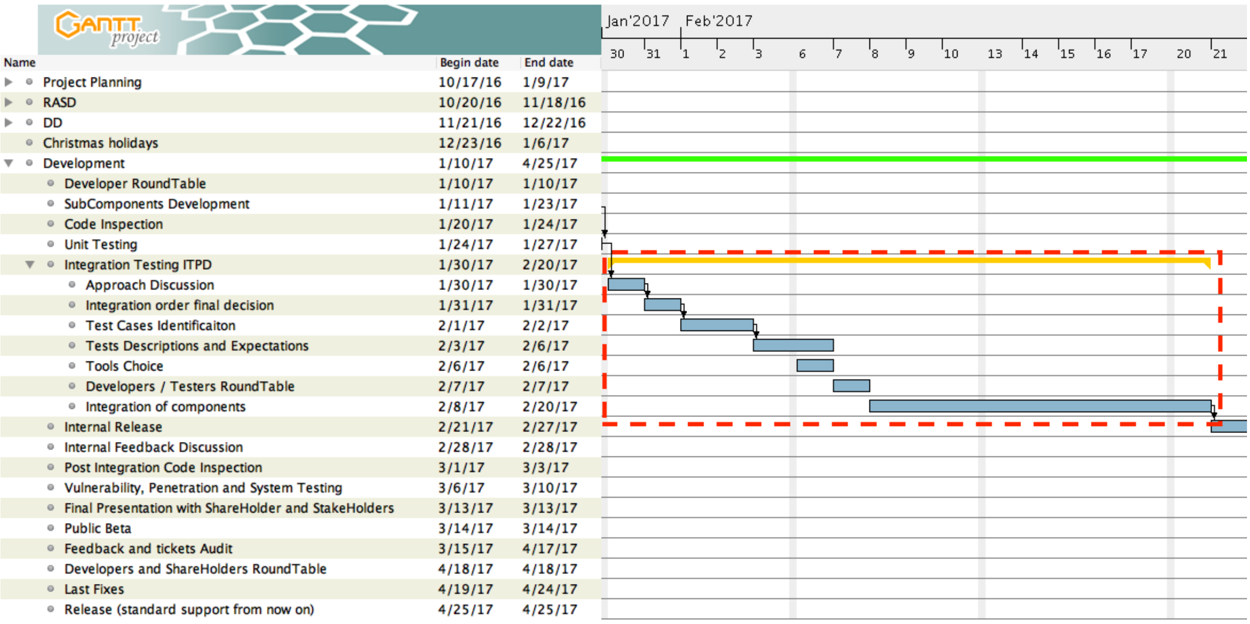
Name	Begin date	End date
Project Planning	10/17/16	1/9/17
RASD	10/20/16	11/18/16
DD	11/21/16	12/22/16
Indentification of Main Co...	11/21/16	11/21/16
Architectural Design	11/22/16	11/28/16
Intentification of Sub Co...	11/29/16	11/29/16
Requirement Traceability...	11/22/16	12/13/16
Runtime View	11/28/16	12/16/16
Algorithms Design	11/28/16	12/16/16
User Interface Design & ...	12/14/16	12/16/16
Document Revision	12/19/16	12/21/16
Meeting with ShareHolders	12/22/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17

3.5 Development Task



Name	Begin date	End date
Project Planning	10/17/16	1/9/17
RASD	10/20/16	11/18/16
DD	11/21/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17
Developer RoundTable	1/10/17	1/10/17
SubComponents Develop...	1/11/17	1/23/17
Code Inspection	1/20/17	1/24/17
Unit Testing	1/24/17	1/27/17
Integration Testing ITPD	1/30/17	2/20/17
Internal Release	2/21/17	2/27/17
Internal Feedback Discus...	2/28/17	2/28/17
Post Integration Code Ins...	3/1/17	3/3/17
Vulnerability, Penetration...	3/6/17	3/10/17
Final Presentation with Sh...	3/13/17	3/13/17
Public Beta	3/14/17	3/14/17
Feedback and tickets Audit	3/15/17	4/17/17
Developers and ShareHo...	4/18/17	4/18/17
Last Fixes	4/19/17	4/24/17
Release (standard suppo...	4/25/17	4/25/17

3.5.1 Integration Testing Task



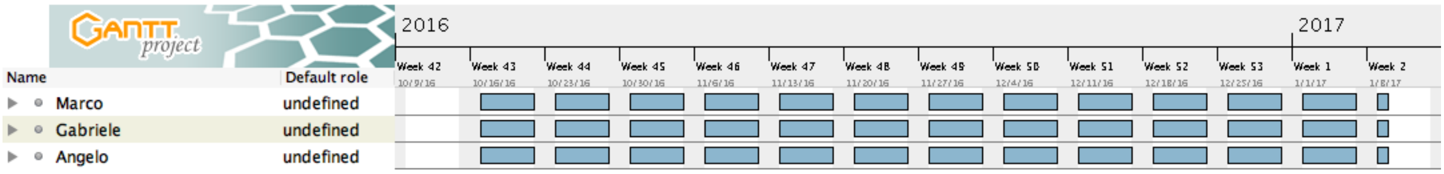
Name	Begin date	End date
Project Planning	10/17/16	1/9/17
RASD	10/20/16	11/18/16
DD	11/21/16	12/22/16
Christmas holidays	12/23/16	1/6/17
Development	1/10/17	4/25/17
Developer RoundTable	1/10/17	1/10/17
SubComponents Development	1/11/17	1/23/17
Code Inspection	1/20/17	1/24/17
Unit Testing	1/24/17	1/27/17
Integration Testing ITPD	1/30/17	2/20/17
Approach Discussion	1/30/17	1/30/17
Integration order final decision	1/31/17	1/31/17
Test Cases Identificaiton	2/1/17	2/2/17
Tests Descriptions and Expectations	2/3/17	2/6/17
Tools Choice	2/6/17	2/6/17
Developers / Testers RoundTable	2/7/17	2/7/17
Integration of components	2/8/17	2/20/17
Internal Release	2/21/17	2/27/17
Internal Feedback Discussion	2/28/17	2/28/17
Post Integration Code Inspection	3/1/17	3/3/17
Vulnerability, Penetration and System Testing	3/6/17	3/10/17
Final Presentation with ShareHolder and StakeHolders	3/13/17	3/13/17
Public Beta	3/14/17	3/14/17
Feedback and tickets Audit	3/15/17	4/17/17
Developers and ShareHolders RoundTable	4/18/17	4/18/17
Last Fixes	4/19/17	4/24/17
Release (standard support from now on)	4/25/17	4/25/17

Chapter 4

Allocate resources

In this chapter we're going to provide a general overview of how the tasks defined by the schedule in the previous section will be divided between the three members of the development team.

Since for the parts really done everyone worked with analogous working hours, we considered that for all parts everyone works on them with the same amount of hours.



Name	Default role
Marco	undefined
Gabriele	undefined
Angelo	undefined

Chapter 5

Risks

5.1 Risk Factors

In this section we're going to assess the main risks that the project development may face.
We adopted a proactive risk strategy and detected the following risks divided in 3 main category.

- Project Risks
- Technical Risks
- Business Risks

5.1.1 Project risks

- **RISK01**[requirements volatility]: requirements are changed by the committee.
- **RISK02**[Unrealistic schedule/budget]: Estimates may be unrealistic with respect to the requirements.
- **RISK03**[Wrong functionality]: May have a variety of causes, such as an imperfect understanding of the customer needs, the complexity of communication with the client, insufficient domain knowledge of the developers and designers.

- **RISK04**[Bad external components]: The quality or functionality of externally supplied components may be below what is required for this project (e.g. google maps service or payment interface unavailable).
- **RISK05**: Key staff are ill at critical times in the project.
- **RISK06**: The size of the software is underestimated.
- **RISK07**: The time required to develop the software is underestimated.

5.1.2 Technical risks

- **RISK08**[Real-time shortfalls]: The real-time performance of (parts of) the system may be inadequate (users requests to application server).
- **RISK09**[Wrong user interface]: In certain situations, the user-friendliness of the interface is critical to its success.
- **RISK10**[Gold plating]: Developers may wish to develop features not asked for by the customer.
- **RISK11**[the database used in the system cannot process as many transactions per second as expected.
- **RISK12**: The rate of defect repair is underestimated

5.1.3 Business risks

- **RISK13**[too few clients]: the sales force did not find enough customers for the product.
- **RISK14**[competitors]: others companies can develop a similar product, reducing our market opportunities.

5.2 Risk Exposure

In this section we mappes each risk to its likelihood of occurrence.

Risk ID	**Probability**
RISK01	Low
RISK02	Low
RISK03	Moderate
RISK04	Very Low
RISK05	Moderate
RISK06	High
RISK07	High
RISK08	Moderate
RISK09	Low
RISK10	Low
RISK11	Moderate
RISK12	Moderate
RISK13	Moderate
RISK14	Moderate

5.3 Rank Risks

In this section we estimated the impact of each risk.

Risk ID	**Effects**
RISK01	Critical
RISK02	Marginal
RISK03	Serious
RISK04	Critical
RISK05	Catastrophic
RISK06	Tolerable
RISK07	Serious
RISK08	Critical
RISK09	Low
RISK10	Marginal
RISK11	Critical
RISK12	Tolerable
RISK13	Critical
RISK14	Critical

5.4 Contingency Plan

In this section we designed a possible strategy to deal with the risks.

Risk ID	Strategy
RISK01	Derive traceability information to assess requirements change impact; maximize information hiding in the design.
RISK02	Investigate buying-in components.
RISK03	Involve the stakeholders in the requirements and design phase with surveys
RISK05	Reorganize team so that there is more overlap of work and people therefore understand each other's jobs.
RISK06	investigate use of a program generator.
RISK07	Investigate buying-in components; investigate use of a program generator.
RISK08	Overrate application servers resources, or investigate the possibility of buying a higher-performance application server or move the system to an highly scalable platform
RISK09	Involve the stakeholders with discussions on the interface design.
RISK10	Stakeholders have an active role in the development of the project, in the requirement analysis and design phases as well as in the implementation phase. Activities in this direction may include periodical reviews and meetings, demonstrations, discussions on the interface design and so on
RISK11	Investigate the possibility of buying a higher-performance database or move the system to an highly scalable platform
RISK12	Replace potentially defective components with bought-in components of known reliability.
RISK13	Promote our product using some advertising
RISK14	Promote our product using some advertising

Chapter 6

Used tools

- **Github:** to publicize our work
- **Texpad:** to design PPD with LaTeX
- **Gattproject:** to draw Gantt project

Chapter 7

Effort Spent

In this section you will include information about the number of hours each group member has worked towards the fulfillment of this deadline.

Date	Marco (h)	Angelo (h)	Gabriele (h)
03/01/2017		1	
14/01/2017			3
15/01/2017			5
17/01/2017	3		2
19/01/2017	3	2	1
20/01/2017	3	5	
21/01/2017		3	
21/01/2017	0.5		
TOTAL	9.5	11	11

We also report here the overall mean amount of hours per person required by the project.

Document	**HOW**
Requirements Analysis and Specifications Document (RASD)	30
Design Document (DD)	30
Integration Test Plan Document (ITPD)	10
Project Plan Document (PPD)	10
Code Inspection Document (CI)	20
Overall document revision	10
Total	**110**