
PowerEnJoy
Software Engineering II

# Project Plan Document

*Giovanni Scotti, Marco Trabucchi*

Document version: 1
January 12, 2017

# Contents

# Section 1

# Introduction

## 1.1 Purpose and Scope

The purpose of this document is to provide a reasonable estimate of the complexity of the *PowerEnJoy* project in terms of the development team effort.

In the first section of the document, two complementary estimation models, *Function Point Analysis (FPA)* and *COCOMO II*, are going to be used in order to support the estimation process; the result will be an estimate of the number of lines of source code (SLOC) to be written and the average effort required for the development process itself.

In the second section, the organizational structure of the project plan will be laid down; here will be defined a possible schedule to cover all activities within the deadline and with a proper redistribution of tasks among the team members.

The last section of the document contains an analysis of all kinds of risks that the project could be facing throughout its life-cycle, from the development phase to the deployment phase, up to the maintenance and support phase. Here are also defined the safety measures, both reactive and preemptive, to face the eventual occurrence of the risks mentioned above.

## 1.2 Definitions, Acronyms and Abbreviations

**RASD:** Requirements Analysis and Specification Document

**DD:** Design Document

**ITPD:** Integration Test Plan Document

**FP:** Function Point

**FPA:** Function Point Analysis

**COCOMO:** COnstructive COst MOdel

**SLOC/KSLOC:** Source Lines Of Code / Kilo Source Lines Of Code

**ILF:** Internal Logic Files

**EIF:** External Interface Files

**EI:** External Inputs

**EO:** External Outputs

**EQ:** External Inquiries

## 1.3 Reference Documents

The indications provided in this document are based on the ones stated in the previous deliverables for the project, the RASD document [1], the DD document [2] and the ITPD document [3].

Moreover it is strictly based on the specifications concerning the RASD assignment [4] for the Software Engineering II project, part of the course held by professors Luca Mottola and Elisabetta Di Nitto at the Politecnico di Milano, A.Y. 2016/17.

To support the application of the COCOMO II model estimate, the COCOMO II Model Definition Manual [5] has been followed.

# Section 2

# Function Points Estimation

This chapter is devoted to the Function Point Analysis for the *PowerEnJoy* project, aimed at obtaining a reasonable estimate of the size of the project, which will be later used within the COCOMO II estimation model to compute an average effort factor for the development process.

## 2.1 Internal Logic Files (ILFs)

Internal Logic Files are defined as follows [5]:

> *"Internal Logic Files count each major logical group of user data or control information in the software system as a logical internal file type. They include each logical file (e.g., each logical group of data) that is generated, used, or maintained by the software system."*

In practice, they can be identified as a homogeneous set of data used and managed by the application itself.

The identified ILFs for the application are:

- User
- Car
- Payment
- Reservation
- Ride
- AlternativeChargesSituation

- SafeArea
- PowerGridStation

## 2.2 External Interface Files (EIFs)

External Interface Files are defined as follows [5]:

> *"Files passed or shared between software systems should be counted as External Interface File types within each system."*

In practice, they can be identified as a homogeneous set of data used by the application, but generated and maintained by other applications.

The identified EIFs for the application are:

- Payment records in the Payment Handlers databases
- Maintenance intervention records in the Maintenance System database
- The data streams related to the *Google Maps* service

## 2.3 External Inputs (EIs)

External Inputs are defined as follows [5]:

> *"External Inputs count each unique user data or user control input type that enters the external boundary of the software system being measured."*

In practice, they can be identified as elementary operations to elaborate data coming from the external environment.

The identified EIs for the application are:

- The registration process
- The login process
- The update process for user profiles
- Data streams from the sensors and equipment of cars
- Data about the availability of cars
- The car reservation process
- The car unlocking process
- The user authentication process for the rides

## 2.4 External Outputs (EOs)

External Outputs are defined as follows [5]:

> *"External Outputs count each unique user data or control output type that leaves the external boundary of the software system being measured."*

In practice, they can be identified as elementary operations that generate data for the external environment.

The identified EOs for the application are:

- E-mail notifications
- User notifications

## 2.5 External Inquiries (EQs)

External Inquiries are defined as follows [5]:

> *"External Inquiries count each unique input-output combination, where input causes and generates an immediate output."*

In practice, they can be identified as elementary operation that involve input and output, without significant elaboration of data from logic files.

The identified EQs for the application are:

- The visualization of user profile data by the user him/herself
- The detection of alternative charges situation during rides

# Section 3

# COCOMO II Effort Estimation

# Section 4

# Schedule Planning and Resource Allocation

## 4.1 Tasks and Schedule

The main tasks of which the *PowerEnJoy* project is composed of are the following:

1. **Requirements Analysis and Specification Document (RASD)** delivery - deliver a document containing the description of all goals, domain assumptions, functional and non-functional requirements for the project;

2. **Design Document (DD)** delivery - deliver a document describing the architectural design of the software system to be produced;

3. **Integration Test Plan Document (ITPD)** delivery - deliver a document containing the strategy to perform integration testing on the components of the system;

4. **Project Plan Document (PPD)** delivery - deliver a document containing the description of the schedule and tasks for the project and an estimation of the effort and size of the project itself, as well as an analysis of the risks that the project could face during its life-cycle;

5. **Implementation** - implement the software product and thoroughly write unit tests for all the code;

6. **Integration testing** - test the integration of all the software components of the project;

During the project life-cycle, some of these task can not begin if others are not completed yet. To illustrate the precedence constraints of the case, a **dependency graph** is provided in Figure 4.1.

Note that, however, since a software project is highly subject to change in requirements and evolves continuously, there might be the need of coming back to previous tasks one or more times after the conclusion of said tasks themselves.

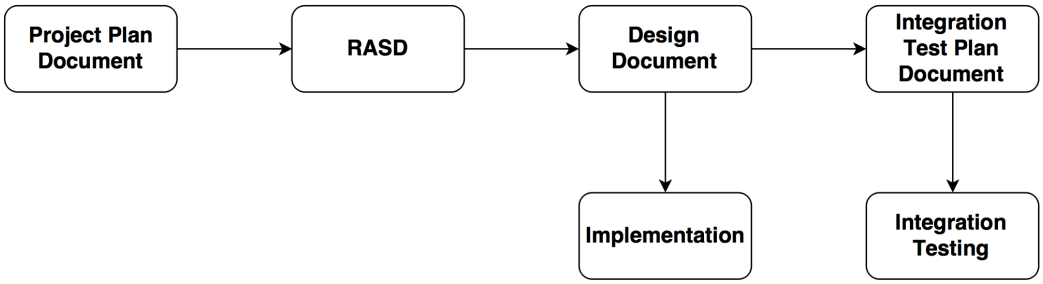


**Figure 4.1:** Dependency graph illustrating precedence constraints.

The following is a Gantt chart to describe the chosen schedule for the project:

## 4.2 Resource Allocation

# Section 5

# Risk Management

The *PowerEnJoy* system project might be threatened by many risks. The analysis of the said risks follows a specific approach that splits them up into three groups: project, technical and business risks.

## 5.1 Project Risks

Project risks pose a threat to the project plan and make the project schedule slip, increasing the overall costs. The following risks have been found and analyzed, providing a response strategy.

**Changes concerning requirements:** during the development of the project the requirements can change unexpectedly. This kind of risk cannot be prevented, but it can be reduced by using a style of programming that takes advantage of reusable and extensible code.

**Deadlines are not met:** it can occur that the project requires more time than expected to be carried out. In this case only some functionalities will be offered in a first release while less essential features will be developed later. The above-mentioned first release can provide the target services by the only means of the Mobile Application. The Web Application and the Web Tier may be built afterwards.

**Lack of experience in using specific frameworks:** if the team has never used Java Enterprise Edition and has no actual experience in programming with the said framework, the development process will definitely be slowed down. This is that case.

**Lack of communication:** team members usually work separately. This can cause misunderstandings and conflicts about the division of the

different tasks. To prevent the risk, crystal clear and complete specification and design documents must be provided. Moreover the responsibilities of each group member must be defined in this document.

**Team cohesion:**

## 5.2 Technical Risks

Technical risks threaten the quality and the punctuality of the software product preventing a straightforward implementation.

**Unreadable code:** huge and large projects may have a badly structured and unreadable code. Documenting the code can be a reasonable way to mitigate this risk. Furthermore information provided by a good Design Document can come in handy too.

**Scalability issues:** if the system does not scale properly with the increasing number of users, a work of major redesign will be carried out. To prevent the issue, a correct estimation of the computing power and the number of involved machines is necessary.

**Integration testing failure:**

**Downtime:**

**Deployment difficulties:**

**Data loss and leaks:**

## 5.3 Economical Risks

Economical risks jeopardize the whole software product threatening its viability.

**Bankruptcy:**

**Local regulation and policies:**

**Competitors:**

# Appendix A

# Appendix

## A.1 Software and tools used

- LaTeX, used as typesetting system to build this document.
- draw.io - `https://www.draw.io` - used to draw diagrams and mock-ups.
- GitHub - `https://github.com` - used to manage the different versions of the document and to make the distributed work much easier.
- GitHub Desktop, the GitHub official application that offers a seamless way to contribute to projects.

## A.2 Hours of work

The absolute major part of the document was produced in group work. The approximate number of hours of work for each member of the group is the following:

- Giovanni Scotti:
- Marco Trabucchi:

NOTE: indicated hours include the time spent in group work.

# Bibliography

[1] AA 2016/2017 Software Engineering 2 - *Requirements Analysis and Specification Document* - Giovanni Scotti, Marco Trabucchi

[2] AA 2016/2017 Software Engineering 2 - *Design Document* - Giovanni Scotti, Marco Trabucchi

[3] AA 2016/2017 Software Engineering 2 - *Integration Test Plan Document* - Giovanni Scotti, Marco Trabucchi

[4] AA 2016/2017 Software Engineering 2 - *Project goal, schedule and rules*

[5] 1995 - 2000 Center For Software Engineering, USC - *COCOMO II - Model Definition Manual* - Version 2.1