Software Engineering 2 Project: PowerEnJoy

Project **P**lan **D**ocument

Marco Ieni, Francesco Lamonaca, Marco Miglionico
Politecnico di Milano, A.A. 2016/2017

January 22, 2017
v1.0

Contents

Chapter 1

Introduction

This is the project plan document for the PowerEnJoy project. In this document we will provide an estimate of the complexity of the project and an initial schedule of the work. In the second chapter we will evaluate the functions points, used in chapter three to estimate the cost and effort needed to develop PowerEnJoy, using COCOMO II. In chapter four we will provide an initial work schedule for the project and in chapter five an initial subdivision of the tasks among all the three components of the group. Finally in chapter six we will estimate which risks the project could incur in and how we think to avoid them.

1.1 Used Acronyms

FP Function Points

ILF Internal Logic File

ELF External Logic File

EI External Input

EO External Output

EQ External Inquiries

DBMS Database Management System

UI User Interface

RASD Requirements Analysis and Specification Document

DD Design Document

ITPD Integration Test Plan Document

PPD Project Plan Document

1.2 Reference Documents

This document refers to the following documents:

- Project goal, schedule and rules of the Software Engineering 2 project;
- PowerEnJoy Requirement Analysis and Specification Document: rasd.pdf;
- PowerEnJoy Design Document: dd.pdf;
- PowerEnJoy Integration Test Plan Document: itpd.pdf;
- The Project Plan Document example;

Chapter 2

Function Points Estimation

A function point is a "unit of measurement" used to compute a functional size measurement (FSM) of software. In the following we will use it to give an estimation of the size and complexity of our project.

This estimation is based on the usage of figures obtained through statistical analysis of real projects, which have been properly normalized and condensed in the following tables:

For Internal Logic Files and External Logic Files

	Data Elements		
Record Elements	*1-19*	*20-50*	*51+*
1	Low	Low	Avg
2-5	Low	Avg	High
6+	Avg	High	High

For External Output and External Inquiry

	Data Elements		
File Types	*1-5*	*6-19*	*20+*
0-1	Low	Low	Avg
2-3	Low	Avg	High
4+	Avg	High	High

For External Input

	Data Elements		
File Types	*1-4*	*5-15*	*16+*
0-1	Low	Low	Avg
2-3	Low	Avg	High

4+	Avg	High	High

UFP Complexity Weights

	Complexity Weight		
Function Type	*Low*	*Average*	*High*
Internal Logic Files	7	10	15
External Logic Files	5	7	10
External Inputs	3	4	6
External Outputs	4	5	7
External Inquiries	3	4	6

2.1 Internal Logical Files (ILFs)

Internal Logical Files (ILFs) represent a user identifiable group of logically related data that resides entirely within the application boundary and is maintained through External Inputs. An internal logical file has the inherent meaning it is internally maintained, it has some logical structure and it is stored in a file.

The ILFs of our system match closely the database tables, for which we remind at Chapter 2.2.1 of the Design Document.

Using the previously defined tables, this is the count we obtain:

ILF	Complexity	FPs
User	High	20
Ride	Average	12
Car	Low	7
Safe Area	Low	4
Power Grid Station + Plug	Low	3
Operator	High	18
Total		64

2.2 External Logic Files (ELFs)

External Interface Files (EIF) represent a user identifiable group of logically related data that is used for reference purposes only and that resides entirely outside the application boundary and is maintained by another applications external inputs.

The only EIF of the system consists in the interface with Google Maps services, which was mentioned in the 2.7.2 subsection of the Design Document.

ILF	Complexity	FPs
Maps	High	22
Total		22

2.3 External Inputs (EIs)

External Inputs (EI) represent an elementary process in which data come from the external environment (like a data input screen or another application).

Our system involves many kind of interactions with different categories of inputs. We are now going to summarize the impact of the offered features, grouping them by user category.

Non registered user: A Non Registered User is someone who has not been logged yet to the system. They can send inputs to the system mainly through these methods:

- **Login/Logout**: the most complex part of these operation is the management of the password field in a secure way. Overall they contribute 3 FP each and they have an average complexity (together).
- **Registration:** We have to register all the fields provided by the user and to validate the document information. In addition to this we have to send a password to the user if the registration is successful. It contributes 8 FP.

User: A user is someone who has been logged to the system. They can send inputs to the system mainly through these methods:

- **Edit profile data:** This is a trivial operation, because we have only to update the elements in the database, except for the editing of the password and the document information, that has to be validated again. It contributes 4 FP and has an average complexity.

- **Delete account:** It contributes 2 FP and has a low complexity.
- **Look for a car with user position or with a specified address:** These are both very complex operations, that involve a large number of components. For this reason they account for 11 FP each and they have an high complexity.
- **Reserve and unlock a car:** These a trivial operation, because they require only to change the state of the car. They contribute 2 FP each and have a low complexity.
- **Start the car:** The server has to check if the password that the user inserts in the tablet is correct and then it has to change the state of the car. It contributes 4 FP and has a low complexity.
- **Enable saving money option:** This is an operation with high complexity, because the system has to compute the best station where to leave the car. It accounts for 18 FP.

HandyCar Board: The board installed on every car of PowerEnjoy. It can send inputs to the system mainly through these methods:

- **End the ride:** When the car is parked in a safe area and the user exits the car the HandyCar Board signal this event to the system. It has a very low complexity, so it contributes 1 FP.
- **Car recharged:** The user connects the car to a plug of a power grid station, so the HandyCar Board signal this event to the system. It has a very low complexity, so it contributes 1 FP.

Operator: The PowerEnjoy employees that has to manage the system from an high level point of view. They can send inputs to the system mainly through these methods:

- **Add a Safe Area:** After some consistency check, the system has to update the database simply adding the new safe area. It has a very low complexity, so it contributes 1 FP.
- **Delete a Safe Area:** After some consistency check, the system has to update the database and wait enough time to avoid creating inconvenience to the user that are using that safe area. It contributes 4 FP and has a low complexity.
- **Add a Power Grid Station:** After some consistency check, the system has to update the database simply adding the new power grid station. It has a very low complexity, so it contributes 1 FP.

- **Delete a Power Grid Station:** After some consistency check, the system has to update the database by eliminating that power grid station. It has a very low complexity, so it contributes 1 FP.

The final results are shown in the following table:

EI	Complexity	FPs
Login/Logout	Average	2x3
Registration	Average	8
Edit profile data	Average	4
Delete account	Low	2
Look for a car with user position/address	High	2x11
Reserve and unlock a car	Low	2x2
Start the car	Low	4
Enable saving money option	High	18
End the ride	Low	1
Car recharged	Low	1
Add a Safe Area	Low	1
Delete a Safe Area	Low	4
Add a Power Grid Station	Low	1
Delete a Power Grid Station	Low	1
Total		77

2.4 External Inquiries (EQs)

External Inquiry (EQ) represents an elementary process with both input and output components that result in data retrieval from one or more internal logical files and external interface files. There must be no significant elaboration of data from logic files.

Our system supports the following EQs:

- Consulting history of the rides;
- Get info about a given car identified by its plate, like the model, the number of passengers and so on;
- Retrieve the list of all the safe areas;
- Retrieve the list of all the power grid stations;

- Retrieve the actual values of the sensors and actuators of a given car.

The resulting table is the following:

EQ	Complexity	FPs
Consult ride history	Low	2
Get car info	Low	2
Retrieve list of safe areas	Low	2
Retrieve list of power grid stations	Low	2
Retrieve car sensors and actuators	Low	2
Total		10

2.5 External Outputs (EOs)

External Outputs (EO) represent an elementary process in which derived data passes across the boundary from inside to outside. From the internal data are created output files, that are sent to other applications.

The EOs of the system are the following:

- Confirmation mails after a successful registration of a user;
- Notify a user that he is in a safe area;
- Verify the validity of the document information;
- Notify a user that he was banned from the system.

The resulting table is the following:

EQ	Complexity	FPs
Confirmation mails	Low	2
Notify a user that he is in a safe area	Low	4
Verify the validity of the document information	Low	3
Notify a user that he was banned from the system	Low	1
Total		10

2.6 Overall estimation

The following table summarizes the results of our estimation activity:

Function Type	Value
Internal Logic Files	64
External Logic Files	22
External Inputs	77
External Inquiries	10
External Outputs	10
Total	183

Relying on the QSM Function Point Languages Table v5.0[1] and considering Java Enterprise Edition as a development platform, leaving the aspects related to the implementation of the mobile applications (which can be thought as pure presentation with no business logic), in the following we can estimate the total number of lines of code.

Average approximation:

```
SLOC = 183 * 46 = 8418
```

Upper bound:

```
SLOC = 183 * 67 = 12261
```

[1] http://www.qsm.com/resources/function-point-languages-table

Chapter 3

Cost and Effort Estimation

3.1 Cost and effort estimation: COCOMO II

In this section we are going to use the COCOMO II approach to estimate the cost and effort needed to develop PowerEnJoy.

3.1.1 Scale Drivers

In order to evaluate the values of the scale drivers, we refer to the following official COCOMO II table:

Scale Factors	**Very Low**	**Low**	**Nominal**	**High**	**Very High**	**Extra High**
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF_j	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF_j	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little(20%)	some(40%)	often(60%)	generally (75%)	mostly(90%)	full(100%)
SF_j	7.07	5.65	4.24	2.83	1.41	0.00

TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF_j	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	Level 1 Lower	Level 1 Upper	Level 2	Level 3	Level 4	Level 5
SF_j	7.80	6.24	4.68	3.12	1.56	0.00

A brief description for each scale driver:

- **Precedentedness**:it reflects the previous experience of our team in the development of similar project. In our case, since we don't have this kind of experience the value will be low.
- **Development Flexibility**:it reflects the degree of flexibility in the development process with respect to the external specification and requirements.We set it to nominal because we have to follow a prescribed process, but we had a certain degree of flexibility in the definition of the requirements and in the design process.
- **Architecture/Risk resolution**: it reflects the level of awareness and reactiveness with respect to risks. The risk analysis we performed is quite extensive, so we have a clear definition of budget and schedule,for this reason the value will be set to very high.
- **Team cohesion**: it reflects how well the development team know each other and work together. We set it to very high, since the cohesion among the three of us is optimal .
- **Process maturity**: it reflects the process maturity of the organization. We set it to 3 because the process comes very close to achieving the specific objectives such as quality, cost, and schedule. Besides processes, standards, procedures and tools were already defined at the organizational level.

The estimated scale drivers for our project, together with the formula to compute the exponent E are the following:

Scale Drivers	Factor	Value
Precedentedness(PREC)	Low	4.96
Development flexibility(FLEX)	Nominal	3.04
Risk Resolution(RESL)	Very High	1.41
Team Cohesion (TEAM)	Very High	1.10
Process Maturity (PMAT)	Level 3	3.12
Total		13.63
Total	E=0.91 + 0.01 x $\sum_{1<=j<=5} SF_j$	1.0463

3.1.2 Cost Drivers

Cost Drivers are the parameters of the Effort Equation that reflect some characteristics of the developing process and act as multiplicators on the effort needed to build the project. They appear as factors in the Effort Equation. Cost Drivers are described in detail in the COCOMO Manual [6, p. 25]. The cost drivers are the following:

- **Required Software Reliability**: Since this is not the only car sharing system in the city,an eventual malfunctioning could led to moderate financial losses,for this reason we set the RELY cost driver to nominal.

RELY Cost Drivers

RELY Descriptors	slightly inconvenience	easy recoverable losses	moderate recoverable losses	high financial losses	risk to human life	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	0.82	0.92	1.00	1.10	1.26	n/a

- **Database size**: This measure considers the effective size of our database. We don't have the exact size, but our estimation given the tables and fields we have is to reach a 1GB database. Since it is distributed over 8.000-12.000 SLOC, the ratio D/P (measured as testing DB bytes/program SLOC) is between 80 and 120 resulting in the DATA cost driver being high.

DATA Cost Drivers

DATA Descriptors		$\frac{D}{P} < 10$	$10 \leq \frac{D}{P} \leq 100$	$100 \leq \frac{D}{P} \leq 1000$	$\frac{D}{P} > 1000$	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	0.82	0.92	1.00	1.10	1.26	n/a

- **Product Complexity**: Set to high according to the COCOMO II rating scale.

CPLX Cost Drivers						
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	0.73	0.87	1.00	1.17	1.34	1.74

- **Required Re-usability**: In our case, the re-usability requirements are just across projects, so the RUSE cost driver is set to nominal.

RUSE Cost Drivers						
RUSE Descriptors		None	Across projects	Across programs	Across product line	Across multiple product line
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	n/a	0.95	1.00	1.07	1.15	1.24

- **Documentation match to life-cycle needs**:This parameter is evaluated in terms of the suitability of the project's documentation to its life-cycle needs. In our case, every need of the product life-cycle is already present in the documentation, so the DOCU cost driver is set to nominal.

DOCU Cost Drivers

DOCU Descriptors	Many life-cycle needs uncovered	Some life-cycle needs uncovered	Right-sized to life-cycle needs	Excessive for life-cycle needs	Very excessive for life-cycle needs	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	0.81	0.91	1.00	1.11	1.23	n/a

- **Execution time constraint**:This parameter is a measure of the CPU usage imposed upon a software system. The rating is expressed in terms of the percentage of available execution time expected to be used by the system or subsystem consuming the execution time resource. In our case,since the software is quite complex,our our expectancy is that its CPU usage will be high.

TIME Cost Drivers

TIME Descriptors			$\leq 50\%$ use of available execution time	70% use of available execution time	85% use of available execution time	95% use of available execution time
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	n/a	n/a	1.00	1.11	1.29	1.63

- **Storage Constraint**: This parameter represents the degree of main storage constraint imposed on a software system or subsystem.As the modern disk drives can easily contain several terabytes of storage, this value is set to nominal.

STOR Cost Drivers

STOR Descriptors			$\leq 50\%$ use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	n/a	n/a	1.00	1.05	1.17	1.46

- **Platform Volatility**:"Platform" is used here to mean the complex of hardware and software (OS, DBMS, etc.) the software product calls on to perform its tasks. In our case we don't expect our fundamental platforms to change very often. However, the client applications may require at least a major release once every six months to be aligned with the development cycle of the main mobile operating systems. For this reason, this parameter is set to nominal.

PVOL Cost Drivers

PVOL Descriptors		Major change every 12 mo;minor change every 1	Major change every 6mo;minor change every 2wk	Major change every 2mo;minor change every 1wk	Major change every 2wk;minor change every 2 days	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	n/a	0.87	1.00	1.15	1.30	n/a

- **Analyst Capability**:Analysts are personnel that work on requirements, high level design and detailed design. The major attributes that should be considered in this rating are Analysis and Design ability, efficiency and thoroughness, and the ability to communicate and cooperate.We think the analysis of the problem has been conducted in very precise and complete way with respect to a potential real world implementation. For this reason, this parameter is set to high.

ACAP Cost Drivers

ACAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.42	1.19	1.00	0.85	0.71	n/a

- **Programmer Capability**:Evaluation should be based on the capability of the programmers as a team rather than as individuals. Major factors which should be considered in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate. The experience of the programmer should not be considered here; it is rated with AEXP.We have not implemented the project, so this parameter is just an estimation; however we think our programming ability are good,but the most important thing is that we know each other and the level of communication and cooperation is optimal, for this reason we set this parameter to high.

PCAP Cost Drivers

PCAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.34	1.15	1.00	0.88	0.76	n/a

- **Application Experience**:This rating is dependent on the level of applications experience of the project team developing the software system or subsystem. We have some experience in the development of Java applications,in particular we made just one project of considerable size,for this reason and for the fact that we never had experience with Java EE we set this parameter to low.

APEX Cost Drivers						
APEX Descriptors	≤ 2 months	6 months	1 year	3 years	6 years	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.22	1.10	1.00	0.88	0.81	n/a

- **Platform Experience**: As we said before we don't have any experience with the Java EE platform, but we have some previous experience with databases, user interfaces and server side development. For this reason, we set this parameter to nominal.

PLEX Cost Drivers						
PLEX Descriptors	≤ 2 months	6 months	1 year	3 years	6 years	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.19	1.09	1.00	0.91	0.85	n/a

- **Language and Tool Experience**:This parameter measure the level of programming language and software tool experience of the project team developing the software system or subsystem. We don't have any experience with the Java EE platform, but we have some previous experience with databases, user interfaces and server side development. We also know the development environment, so we're going to set this parameter to nominal.

LTEX Cost Drivers

LTEX Descriptors	≤ 2 months	6 months	1 year	3 years	6 years	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.20	1.09	1.00	0.91	0.84	n/a

- **Personnel continuity**:The rating scale for PCON is in terms of the project's annual personnel turnover. In our case since we have very strict deadlines and since the time we can spend on this project is limited we set this parameter to very low.

PCON Cost Drivers						
PCON Descriptors	48% / year	24% / year	12% / year	6% / year	3% / year	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.29	1.12	1.00	0.90	0.81	n/a

Usage of Software Tools: Our application environment is quite complete and well integrated, so we set this parameter as nominal.

TOOL Cost Drivers

TOOL Descriptors	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.17	1.09	1.00	0.90	0.78	n/a

Multisite development:Even if we live in different cities, we have met a considerable number of times and we have also collaborated hugely through Internet services including social networks,Skype,email and Whatsapp. For this reason, we set this parameter to very high.

SITE Cost Drivers

SITE Collocation Descriptors	International	Multi-city and multi-company	Multi-city or multi-company	Same city or metro area	Same building or complex	Fully collocated
SITE Communication Descriptors	Some phone,email	Individual phone,fax	Narrow band email	Wideband electronic communication	Wideband electronic comm. and occasionally video conf.	Interactive multimedia
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.22	1.09	1.00	0.93	0.86	0.80

- **Required development schedule**: This rating measures the schedule constraint imposed on the project team developing the software. The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort. Our efforts were well distributed over the available development time, the definition of all the required documentation took a consistent amount of time. For this reason, this parameter is set to high.

SCED Cost Drivers

SCED Descriptors	75% of nominal	85% of nominal	100% of nominal	130% of nominal	160% of nominal	
Rating level	very low	low	nominal	high	very high	extra high
Effort multipliers	1.43	1.14	1.00	1.00	1.00	n/a

To summarize the total results are expressed in the following table:

Cost Drivers	**Factor**	**Value**
Required Software Reliability(RELY)	nominal	1.00
Database Size(DATA)	nominal	1.10
Product Complexity(CPLX)	high	1.17
Required Reusability(RUSE)	nominal	1.00
Documentation match to life-cycle needs(DOCU)	nominal	1.00
Execution Time Constraint(TIME)	high	1.11
Main Storage Constraint(STOR)	nominal	1.00
Platform Volatility(PVOL)	nominal	1.00
Analyst Capability(ACAP)	high	0.85
Programmer Capability(PCAP)	high	0.88
Application Experience(APEX)	low	1.10
Platform Experience(PLEX)	nominal	1.00
Language and Tool Experience(LTEX)	nominal	1.00
Personnel Continuity(PCON)	very low	1.29
Usage of Software Tools(TOOL)	nominal	1.00
Multisite development(SITE)	very high	0.86
Required Development Schedule(SCED)	high	1.00
Total		1.3040

3.1.3 Effort equation

This final equation gives us the effort estimation measured in Person-Months (PM):

```
Effort = A * EAF * KSLOC^E
```

where:

```
A = 2.94 (for COCOMO II.2000)
EAF = product of all cost drivers (1.3040)
E = exponent derived from the scale drivers. It is computed as
  :
   B + 0.01 * ∑_i SF[i] = B + 0.01 * 13.63 = 0.91 + 0.1363 =
  1.0463
   in which B is equal to: 0.91 for COCOMO II.
```

With this parameters we can compute the effort value, which has a lower bound of:

```
Effort = A * EAF * KSLOC^E = 2.94 * 1.3040 * 8418^1.0463 = 49.042
  PM ≈ 49 PM
```

and an upper bound of:

```
Effort = A * EAF * KSLOC^E = 2.94 * 1.3040 * 12261^1.0463 = 72.685
  PM ≈ 73 PM
```

3.1.4 Schedule estimation

Regarding the final schedule, we are going to use the following formula:

```
Duration = 3.67 * Effort^F
```

As a lower bound, we consider

```
F = 0.28 + 0.2 * (E − B) = 0.28 + 0.2 * 0.1363 = 0.30726
Effort = 49.042 PM
Duration = 3.67 * (49.042)^0.30726 = 12.14 months
```

while as an upper bound, we consider

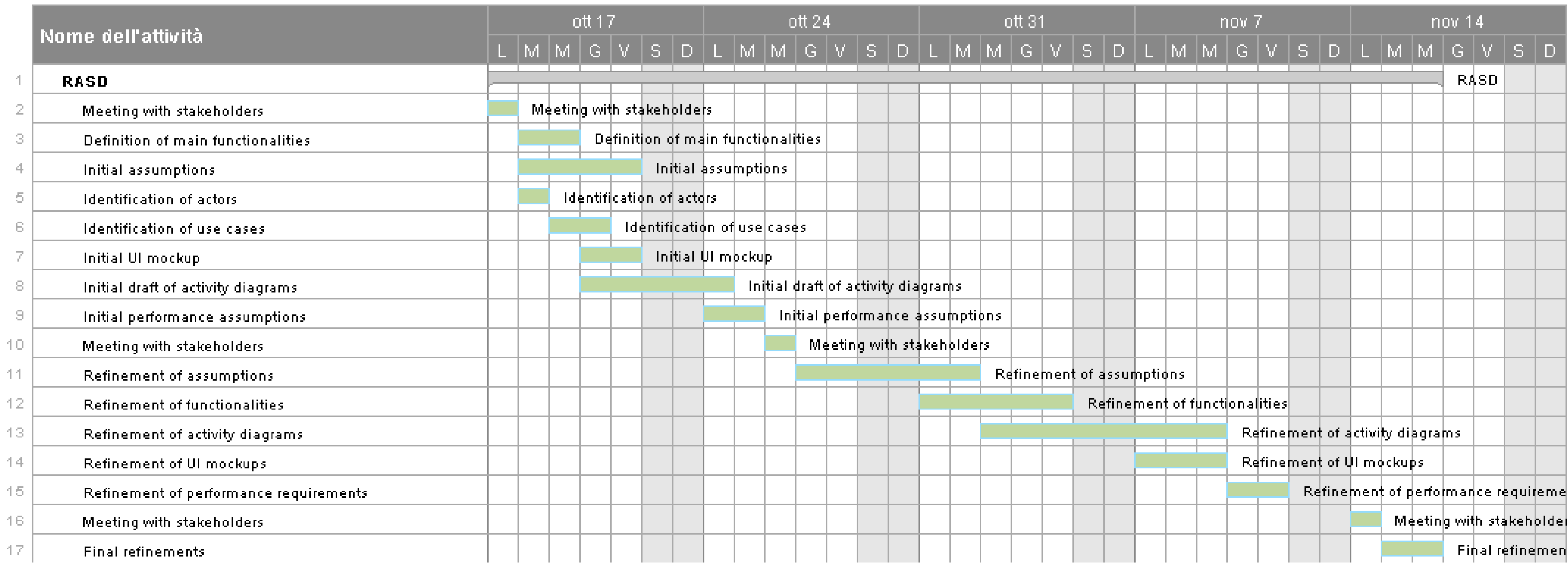
```
F = 0.28 + 0.2 * (E − B) = 0.28 + 0.2 * 0.1363 = 0.30726
Effort = 72.685 PM
Duration = 3.67 * (72.685)^0.30726 = 13.70 months
```

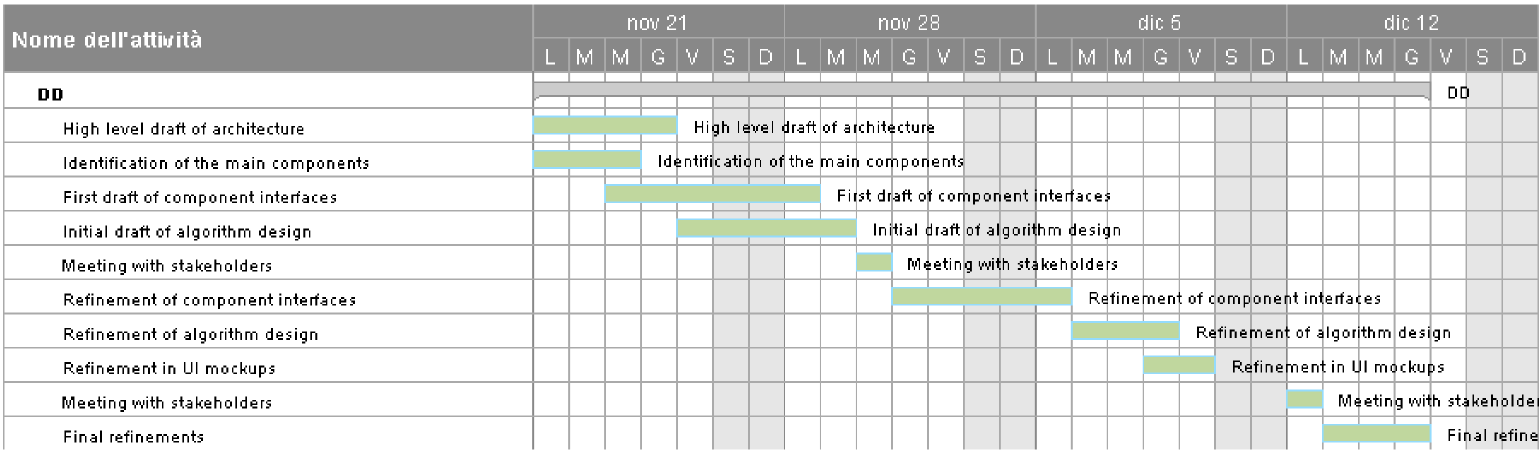
which seem to be both reasonable estimates.

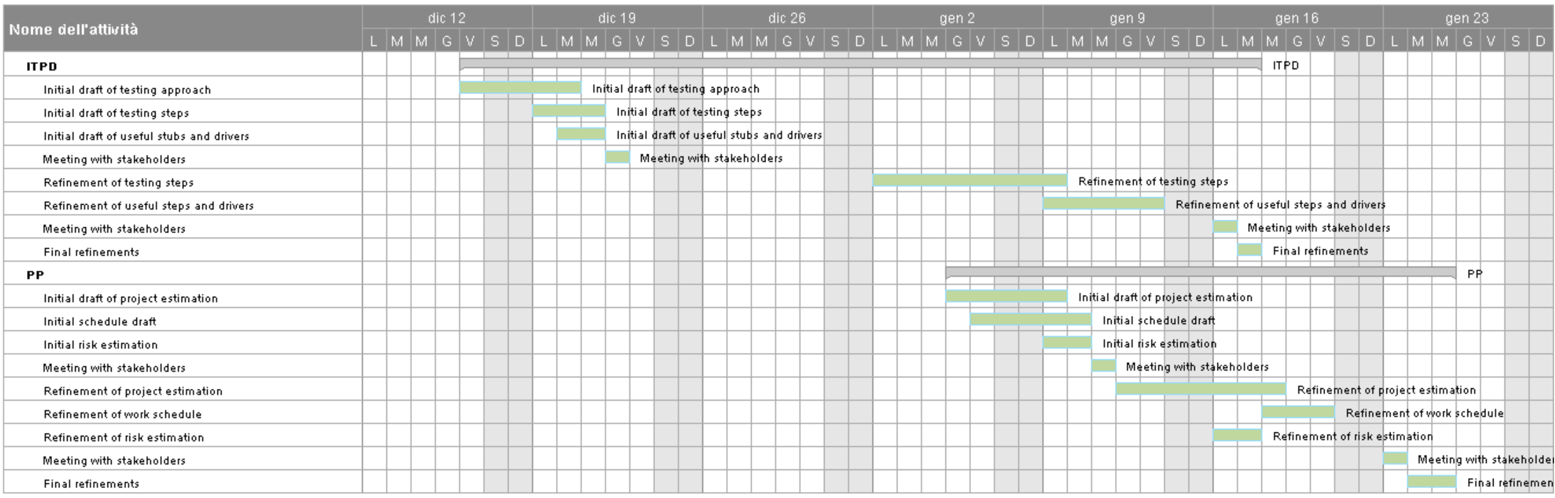
Chapter 4

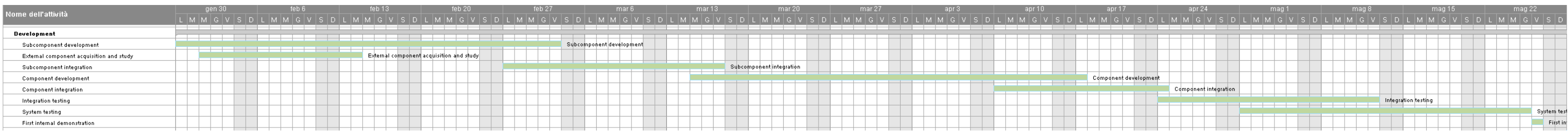
Schedule

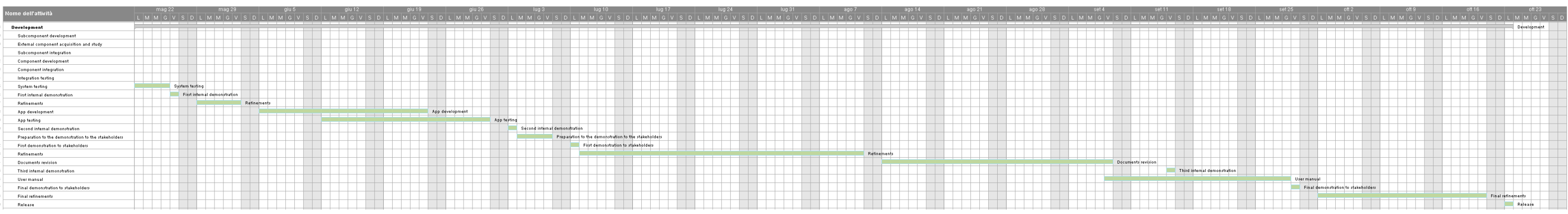
In this chapter, we will provide a general project schedule. More refined schedule will be provided during project development, in order to reflect eventual changes. Even if this project is made only for didactic purpose we will provided a hypothetical schedule of the development phase of the project, that we will not deal with. This is to mimic how a real project should work. In order to maintain readability, we split the schedule into five parts: one for the RASD, one for the DD, one for the ITPS and PPD and two for the development.




Nome dell'attività
nov 21
nov 28
dic 5
dic 12
L M M G V S D
18 DD
19 High level draft of architecture
20 Identification of the main components
21 First draft of component interfaces
22 Initial draft of algorithm design
23 Meeting with stakeholders
24 Refinement of component interfaces
25 Refinement of algorithm design
26 Refinement in UI mockups
27 Meeting with stakeholders
28 Final refinements


Nome dell'attività
dic 12
dic 19
dic 26
gen 2
gen 9
gen 16
gen 23
L M M G V S D
29 ITPD
30 Initial draft of testing approach
31 Initial draft of testing steps
32 Initial draft of useful stubs and drivers
33 Meeting with stakeholders
34 Refinement of testing steps
35 Refinement of useful steps and drivers
36 Meeting with stakeholders
37 Final refinements
38 PP
39 Initial draft of project estimation
40 Initial schedule draft
41 Initial risk estimation
42 Meeting with stakeholders
43 Refinement of project estimation
44 Refinement of work schedule
45 Refinement of risk estimation
46 Meeting with stakeholders
47 Final refinements


Nome dell'attività
Development
Subcomponent development
External component acquisition and study
Subcomponent integration
Component development
Component integration
Integration testing
System testing
First internal demonstration

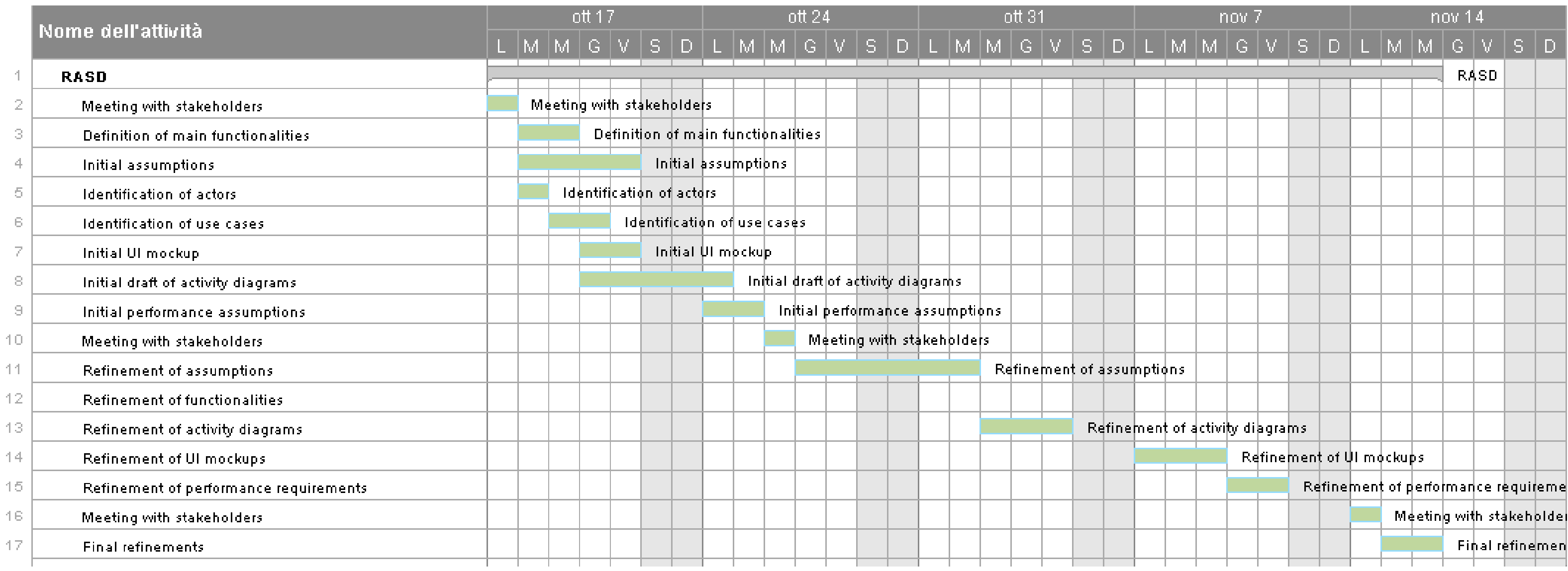


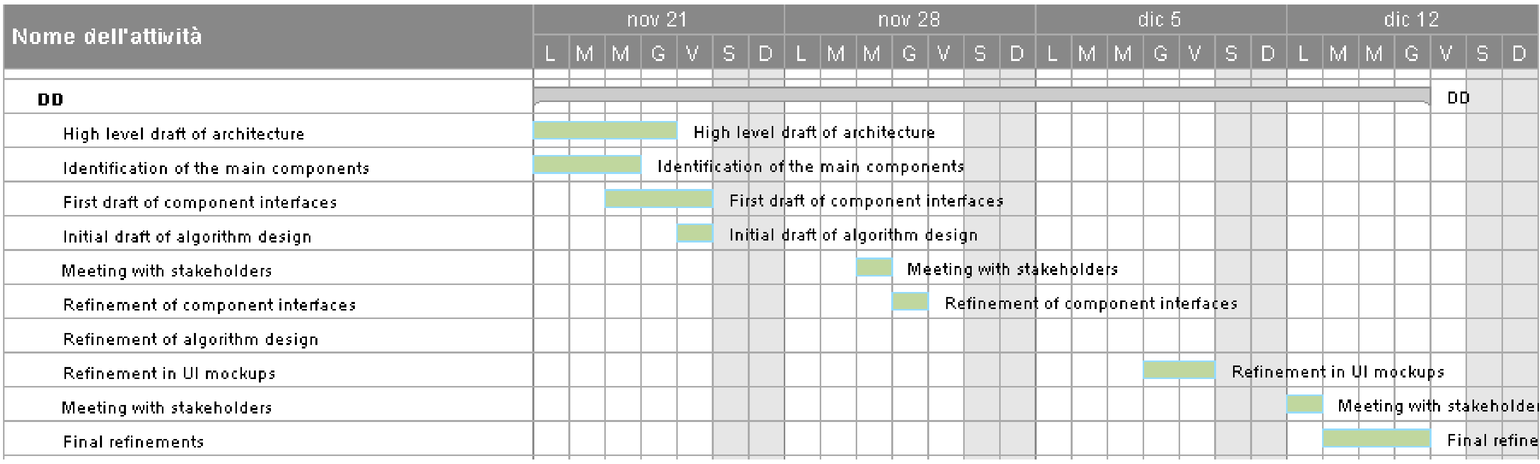
Chapter 5

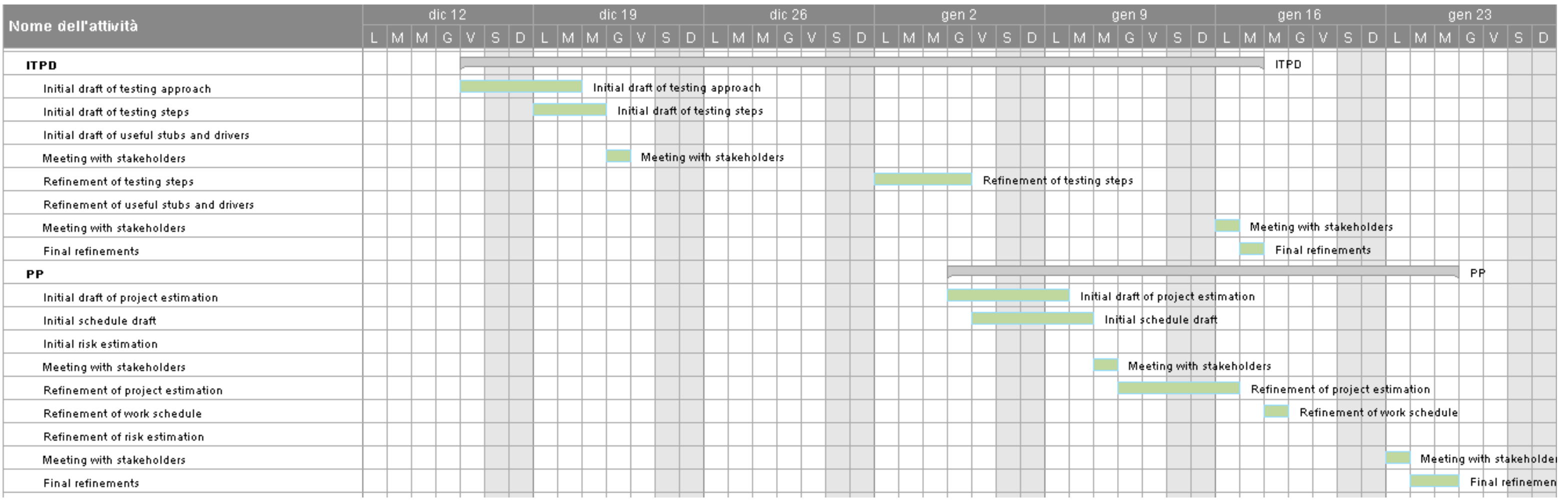
Resource Allocation

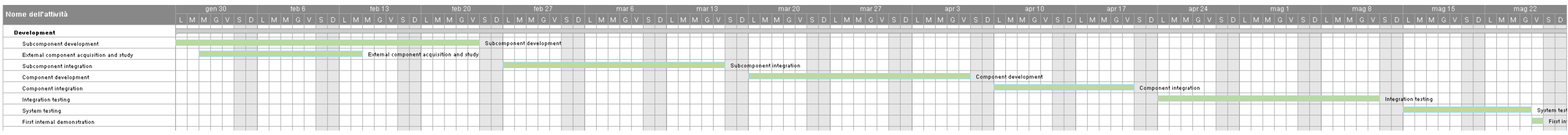
In this chapter, we will describe how the tasks defined into the previous chapter will be divided between the three members of the group. As already mention we consider also the development phase of the project, and in order to maintain readability we split the schedule into five part.

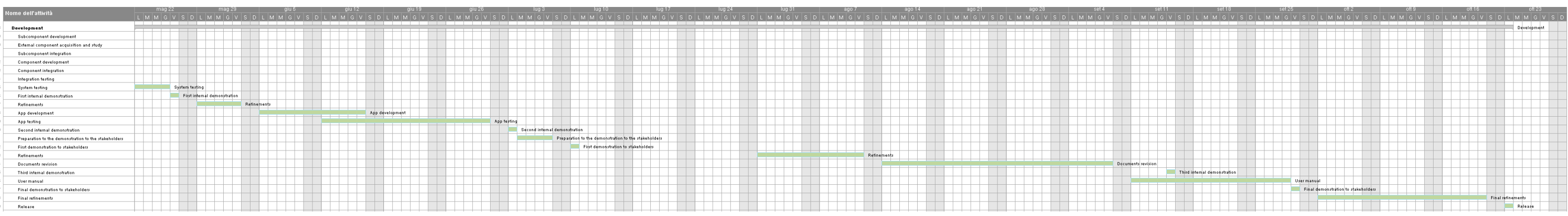
5.1 Marco Ieni



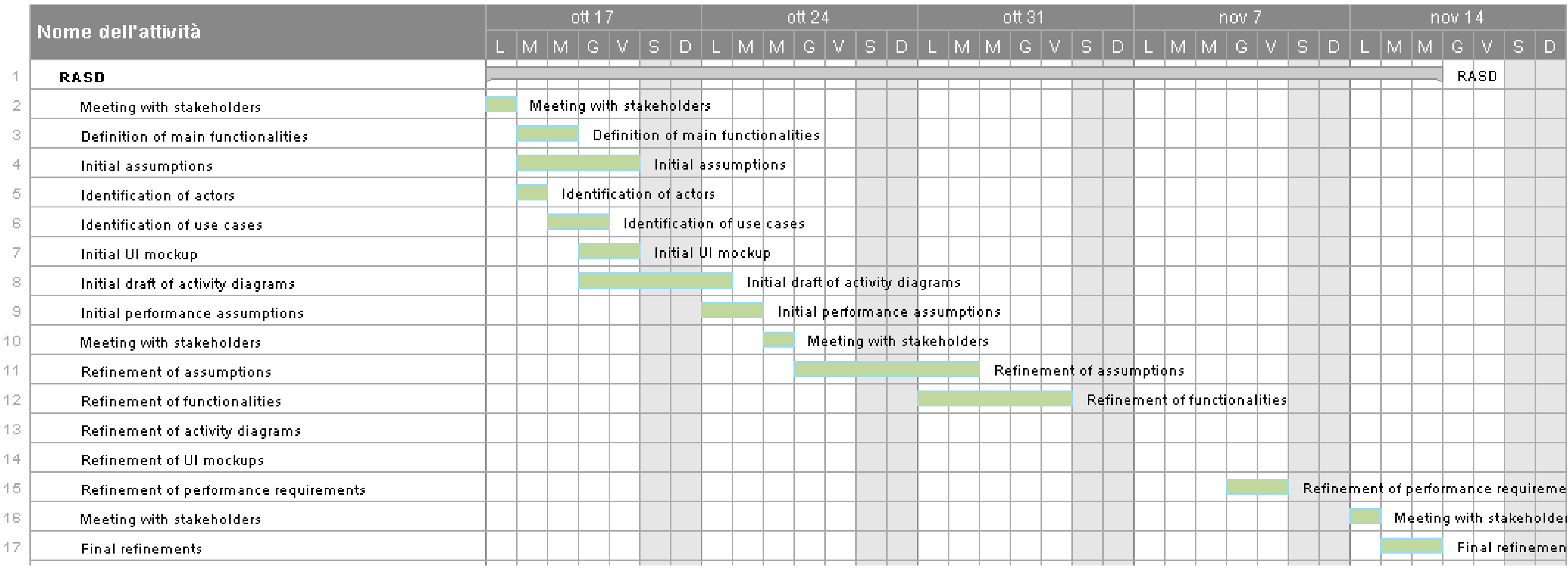

Nome dell'attività
nov 21
nov 28
dic 5
dic 12
18 DD
19 High level draft of architecture
20 Identification of the main components
21 First draft of component interfaces
22 Initial draft of algorithm design
23 Meeting with stakeholders
24 Refinement of component interfaces
25 Refinement of algorithm design
26 Refinement in UI mockups
27 Meeting with stakeholders
28 Final refinements

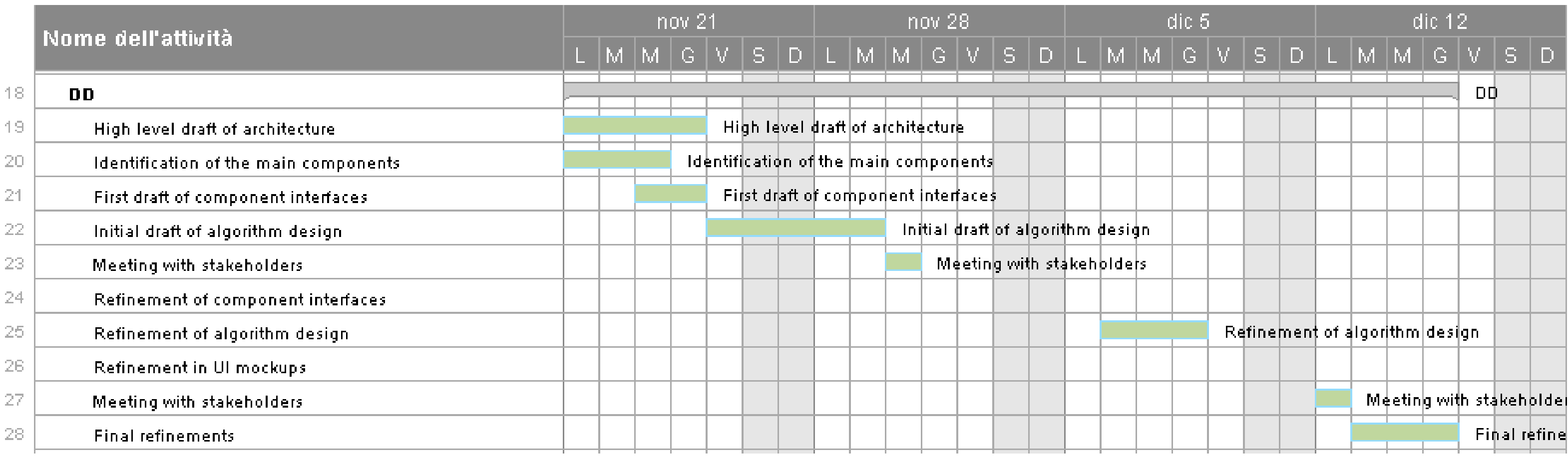

Nome dell'attività
dic 12
dic 19
dic 26
gen 2
gen 9
gen 16
gen 23
29 ITPD
30 Initial draft of testing approach
31 Initial draft of testing steps
32 Initial draft of useful stubs and drivers
33 Meeting with stakeholders
34 Refinement of testing steps
35 Refinement of useful stubs and drivers
36 Meeting with stakeholders
37 Final refinements
38 PP
39 Initial draft of project estimation
40 Initial schedule draft
41 Initial risk estimation
42 Meeting with stakeholders
43 Refinement of project estimation
44 Refinement of work schedule
45 Refinement of risk estimation
46 Meeting with stakeholders
47 Final refinements

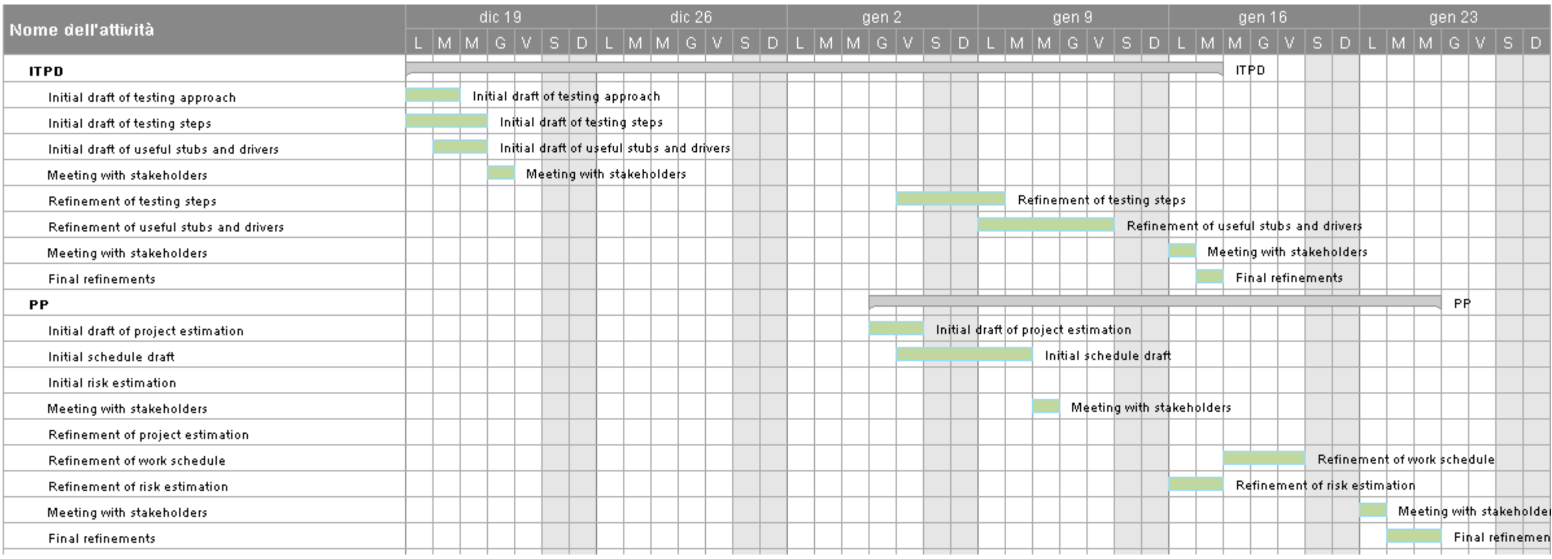


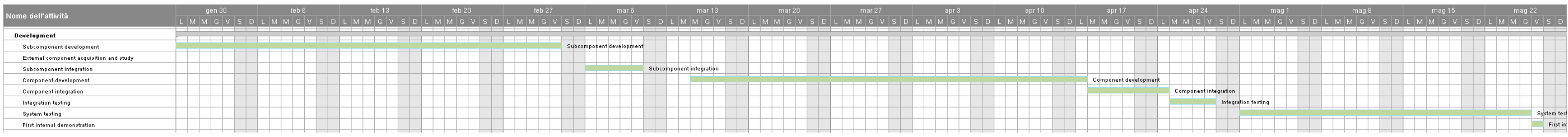


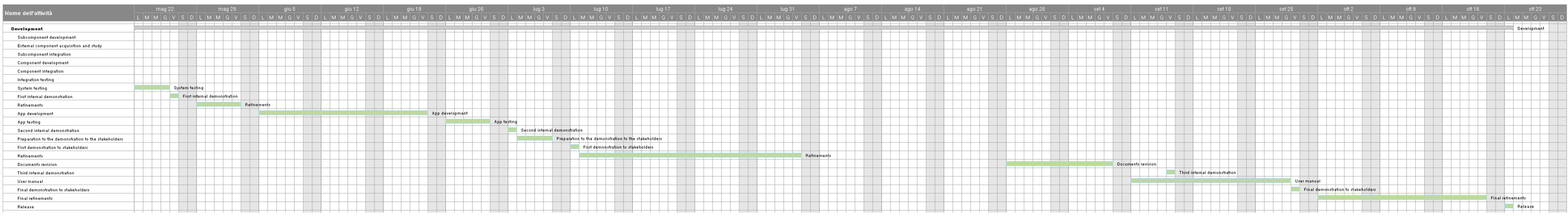
5.2 Francesco Lamonaca



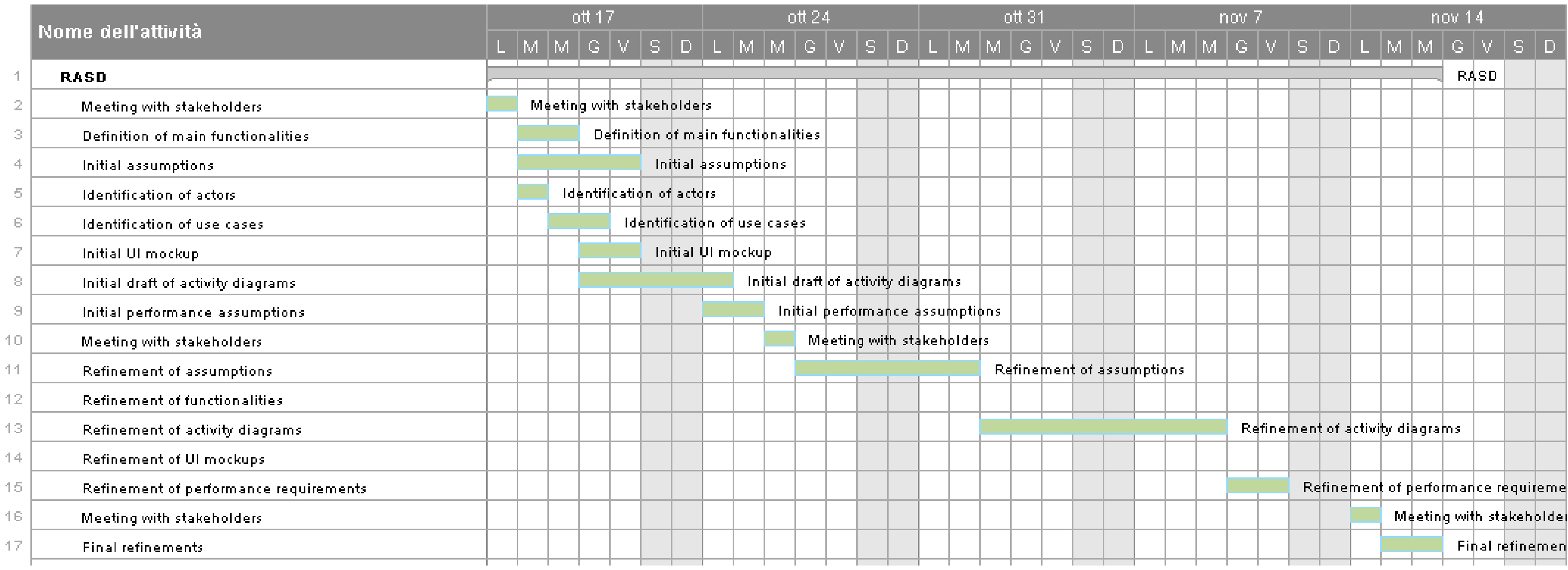


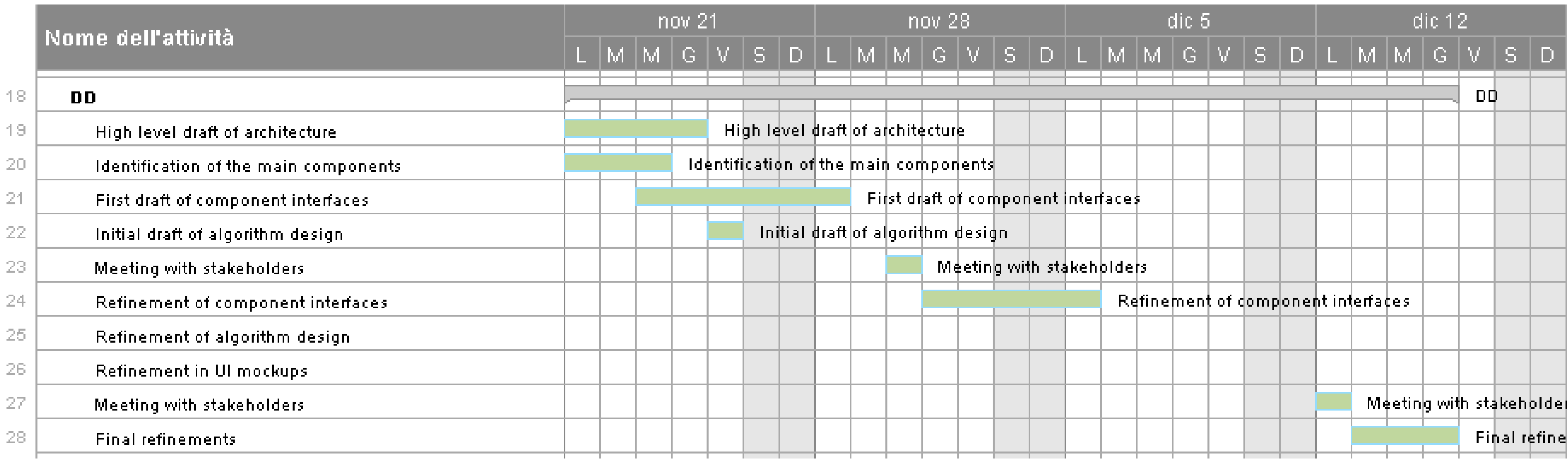

Nome dell'attività
dic 19
dic 26
gen 2
gen 9
gen 16
gen 23
29 ITPD
30 Initial draft of testing approach
31 Initial draft of testing steps
32 Initial draft of useful stubs and drivers
33 Meeting with stakeholders
34 Refinement of testing steps
35 Refinement of useful stubs and drivers
36 Meeting with stakeholders
37 Final refinements
38 PP
39 Initial draft of project estimation
40 Initial schedule draft
41 Initial risk estimation
42 Meeting with stakeholders
43 Refinement of project estimation
44 Refinement of work schedule
45 Refinement of risk estimation
46 Meeting with stakeholders
47 Final refinements

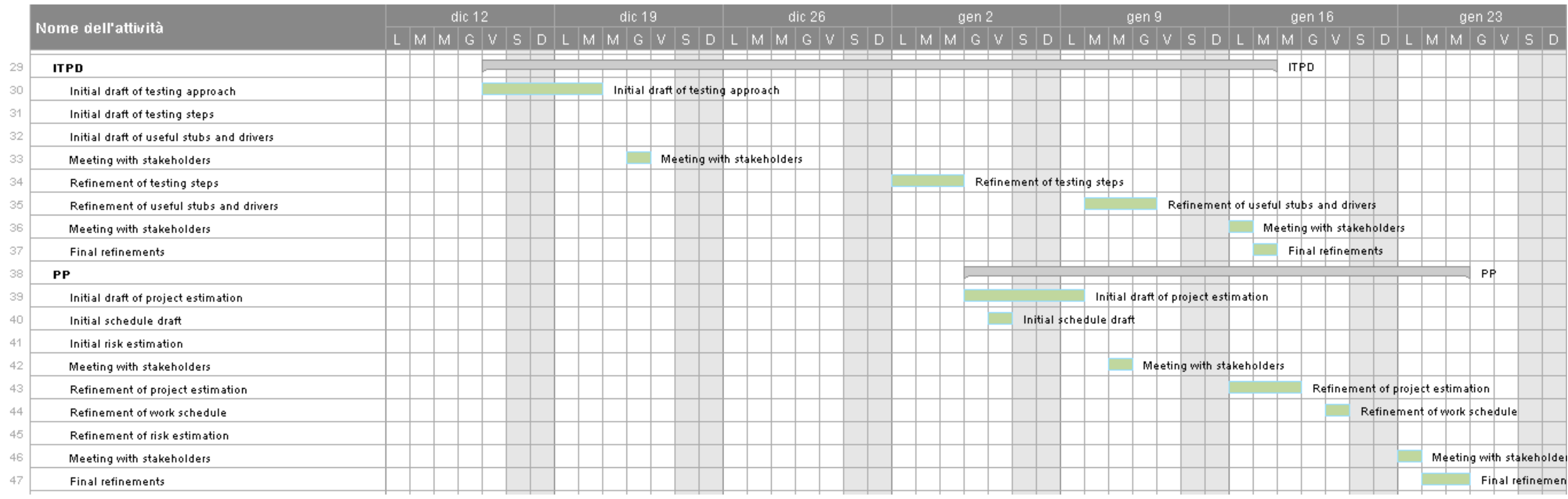

Nome dell'attività
Development
Subcomponent development
External component acquisition and study
Subcomponent integration
Component development
Component integration
Integration testing
System testing
First internal demonstration

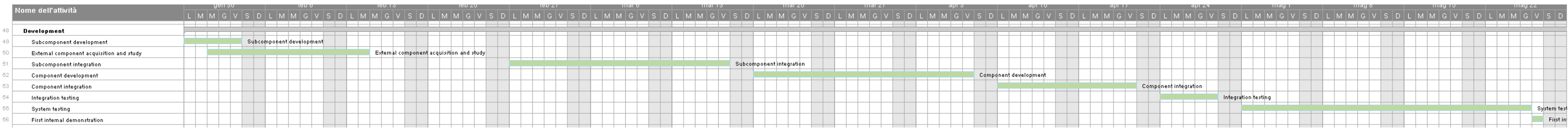


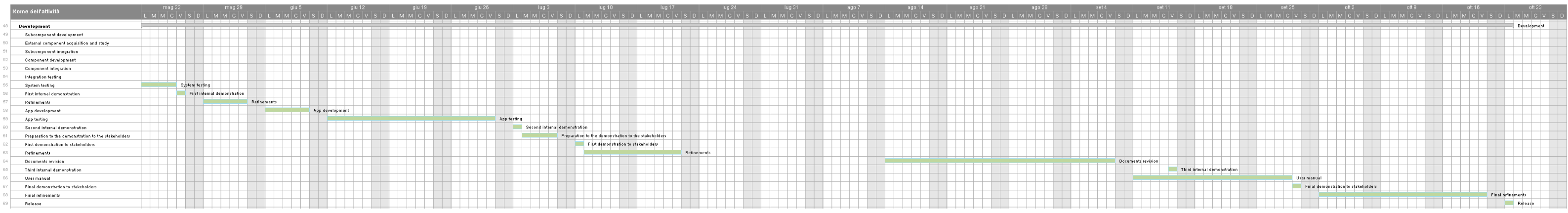
5.3 Marco Miglionico






Nome dell'attività
dic 12
dic 19
dic 26
gen 2
gen 9
gen 16
gen 23
L M M G V S D
29 ITPD
30 Initial draft of testing approach
31 Initial draft of testing steps
32 Initial draft of useful stubs and drivers
33 Meeting with stakeholders
34 Refinement of testing steps
35 Refinement of useful stubs and drivers
36 Meeting with stakeholders
37 Final refinements
38 PP
39 Initial draft of project estimation
40 Initial schedule draft
41 Initial risk estimation
42 Meeting with stakeholders
43 Refinement of project estimation
44 Refinement of work schedule
45 Refinement of risk estimation
46 Meeting with stakeholders
47 Final refinements
ITPD
Initial draft of testing approach
Meeting with stakeholders
Refinement of testing steps
Refinement of useful stubs and drivers
Meeting with stakeholders
Final refinements
PP
Initial draft of project estimation
Initial schedule draft
Meeting with stakeholders
Refinement of project estimation
Refinement of work schedule


Nome dell'attività
Development
Subcomponent development
External component acquisition and study
Subcomponent integration
Component development
Component integration
Integration testing
System testing
First internal demonstration



Chapter 6

Risk Management

A project development is not without any risk. In this section we will discuss some risks that our project may incur, how likely are they, what are their effects and how we would fix them. We can divide the risks into two main categories: development risk and organizational ones.

6.1 Development risks

Delays on the deadlines This is the most common kind of risk during a project. However a well-defined work organization and a good parallelization of the tasks can minimize this risk. In the remote case of a delay on the final release, it may be useful to release a preliminary version of the system, with all the basic functionalities, and adding the advanced ones later.

Lack of communication between members Our group members can work from home. In this way, they can maximize the efficiency and work in an easier and more comfortable place. However the communication during the development may be more difficult but a rigorous tasks division and a well-written Project Plan may avoid some misunderstandings between members.

Requirement change Some requirements can change during development. This risk is very difficult to prevent, but writing a more flexible and modular code may reduce the effects.

Code loss Losing part or all the code is a potentially catastrophic risk. A careful back-up management may avoid this risk.

Integration test failure One or more failure during the integration test phase could create some delays one the deadlines. To minimize the effect of this risk it is needed to start the integration test phase as soon as possible.

OS update incompatibility Our application will be used on Android and iOS smartphone. In order to avoid that an update of the OS creates a conflict with our app it is needed to follow all the news made by Google and Apple for developer about further update.

6.2 Organizational risks

Car issues Car stock is a critical resource for our system. In order to guarantee an adequate number of cars it is important to schedule a periodic maintenance.

Car theft The theft of one or more of our cars is a risk not to underestimate. Each car must be equipped with a antitheft system and a GPS to locate eventual lost cars. It is also important to insure each car.

Data loss and data leak Our system will manage some pieces of personal information, such as the payment ones. We have to protect these data to avoid a loss or a leak. Multiple back-up and adopting industry security standard can minimize this risk.

Dependency from external services Our project will use third party services such as Google Maps or the HandyCar system. A change in the terms and conditions of one of these services may reflect on the usability of our system. A general solution for this risk does not exist, but it is our duty to contact third party to negotiate new arrangements.

Competitivity Our service is not the first rent service to start. In order to be profitable the competitiveness of our product must be continuously enhanced by introducing innovative features and by keeping our prices competitive.

Appendix A

Appendix

A.1 Used software and tools

- LaTeX [1], for typesetting this document.
- Texmaker[2], for the writing of this document.
- GitHub[3] for version control and distributed work.
- GitHub desktop[4] used to collaborate in the team and to keep track of the changes.
- Smartsheet [5] for Gantt chart.

A.2 Work hours

The statistics about commits and code contribution are available on the GitHub repository of the project[6].

These are our estimation of the work hours spent on this project:

- Marco Ieni: 9 hours
- Francesco Lamonaca: 9 hours
- Marco Miglionico: 9 hours

[1] `https://www.latex-project.org/`
[2] `http://www.xm1math.net/texmaker/`
[3] `https://github.com/`
[4] `https://desktop.github.com/`
[5] `https://www.smartsheet.com/`
[6] `https://github.com/marcomiglionico94/Software-Engineering-2-Project`