Project Plan Document- PowerEnJoy

POLITECNICO
MILANO 1863

Version 1.0

- Bagna Francesco Matteo (mat. 878556)
- Barletta Carmen (mat. 877129)

Commit date: 22/01/17

Summary

1: Introduction

1.1: Revision history

This document had no revisions.

1.2: Purpose and scope

This is the Project Planning Document for the PowerEnJoy system.
Its main functionality is to give a sketch of the project complexity, along with a definition of the size and the risks of our project. It will also define the schedule of the tasks to create the documents of the project and the allocation of the resources (people) in the various tasks.
Here is a brief summary of the contents of each chapter.

- 1: Introduction. This chapter contains the first words about this document, along with useful acronyms, abbreviations and definitions that have been used while preparing this document and the references we used.
- 2: Project size, cost and effort estimation. Contains a full estimation of the size of the project; it will use the functions points approach for the size estimation and COCOMO II for cost and effort.
- 3: Schedule. This chapter contains the schedule of the project, considering it began in October, 2016. The schedule will be divided in documents to be made and, at a lower level, in individual tasks. Every task will have a period of time in which it should be done.
- 4: Resource allocation. This chapter will contain schemas for the tasks given to the resources (people) involved in the project.
- 5: Risk Management. Contains an evaluation of possible risks for the project and, if any, the solutions that could be applied if those risks were to come true.
- 6: Effort Spent. The number of hours each member of the group spent in doing this document.

1.3: Acronyms, abbreviations and definitions

For other acronyms, abbreviations and definitions please refer to the RASD, DD and ITPD documents.

- PPD (Project Planning Document)
- FP (Function Points)
- ILF (Internal Logic Files)
- ELF (External Logic Files)
- EI (External Inputs)
- EQ (External Inquiries)
- EO (External Outputs)
- Schedule: sequence of tasks that will led the project to be completed. Each task has a time in which it must be completed and one or more resources (people) allocated to it.

- Risk: any sort of problem, delay or fact that could lead the project to further delays and generally to the impossibility of completing the project's goals in the established time.

1.4: References

- Our RASD, DD, ITPD documents
- Project planning example document.pdf
- Project Management Basics + Advanced Dec. 1.pdf

2: Project size, cost and effort estimation

In this section we will try to estimate the source line of code that there will be needed in order to develop the Power EnJoy system and then we will estimate the effort, measured in Person-Month.

In the size estimation we will use the Function Points approach, taking into account the main functionalities needed to develop the system and in particular its business logic.

For the cost and effort estimation we will instead rely on the COCOMO II approach, using the source line of code computed, to calculate the effort.

2.1: Size estimation, function points

The Functional Points approach is a technique used to assess the effort needed to design and develop custom software application. It is based on a combination of program characteristics, like: data structures, inputs and outputs, inquiries and external interfaces, and statistical analysis on real projects.

2.1.1: Internal Logic Files (ILFs)

The ILFs are the homogeneous set of data used and managed by the PowerEnJoy application. There are many sets of this type on which, the PowerEnjoy systems rely on.

First the system has to manage and store info about the users together with their payment info and their driving license info. In order to store these information we need three tables: the first one concerning the User's personal info has as attributes the Id, the name and surname, the SSN, the password, the e-mail, the phone number, his last ride and his last reservation; the second one for the Payment Info has as attributes, the owner, the credit card type, the card number, the card secure code and the expiry date and the third is the Driving License info table, whose attributes are, the owner, the code, the expiry date and the type.

In order to store the operator info and the administrator info, we need two tables with an almost similar structure. Both entities have the same attributes: the id, the name, the surname, the email and the password.

The information about the cars that the PowerEnjoy company owns are stored into a different table with those attributes: the register, the type, the position and the car availability ("Charging", "Out of service", "Available", "Reserved"). The car state information are retrieved by the GEB and contain all the sensors' states, such as the doors' one, the engine's one etc. There are a lot of data that are used to control the car, indeed this entity can have many attributes, so for this reason we can consider the car's info having an average ILF complexity.

The PowerEnjoy system needs also to save the position of its Safe and Special Safe areas. We can store this information in a single table whose attributes are: the id, the position and the state, which is null for the Safe Areas while for the Special Safe Area can be ("Available" or "Full").

The info about all the Users' Reservations and Ride are stored into two other entities: the Reservation has an id, a date-time, while the Ride is associated to a Reservation has an id, a price, a distance and some flags concerning the payment ("paid" or "not paid") and all the types of discounts and penalties.

ILF	Complexity	FPs
User's info	Low	7
Driving license info	Low	7
Payment info	Low	7
Administrator info	Low	7
Operator info	Low	7
Car's info	Avg	10
(Special) Safe Area info	Low	7
Reservation	Low	7
Ride	Low	7
Total		**66**

2.1.2: External Logic Files (ELFs)

The ELF are homogeneous set of data used by the PowerEnJoy system but generated and maintained by external applications. One of the most important interactions between our system and the external entity, is the one with the Google Maps API, which happens through an HTTP interface and a JSON file format as response. This interaction allow the system to show to the user the map of the city with available cars nearby and to the internal system to calculate different discount and penalties and to properly geolocate with extremely precision the position of different actors of the system (users, cars, operators). In order to be useful, the info retrieved by the Google Maps API need to be processed, by some more or less complicated algorithms.

The system needs also an interaction with an external Payment system to perform payment transactions and another interaction with the authority of the driving license system to verify the validity of the users' driving license.

ELF	Complexity	FPs
Car searching computation	High	10
Payment Transactions	Avg	7
Driving License Verification	Low	5
Total		**22**

2.1.3: External Inputs (EIs)

The EIs are elementary operations that are performed to elaborate data coming from the external environment.

The Power EnJoy system has to manage a lot of different inputs coming from various types of clients.

Users:

- Login/Logout operations, that can be considered simple because they involve only the User Manager.
- Reserve a car, and this is a more complex operation because it requires the interaction with the User Manager, the Reservation Manager, the Location Manager and the Car Manager
- Cancellation of a reservation instead is simpler
- Registration of a new account requires many checking in the validity of the credentials and involves at least three components, so It can be considered of average complexity

Administrators:

- Insert, delete, update safe and special safe areas requires the interaction of at least four components. It can be seen as having an average complexity
- Insert, delate and update the operators' information are quite easy to implement functionalities
- Insert, delate and update cars' info are more complex operation because they require the interaction with the GEB, the Location Manager and the database.
- Consult the historical information about reservations, rides and payments. These operations can be considered of an average complexity

Operators:

- Answer to a request of maintenance can be considered simple because requires the interaction between two components
- Change the state of a car can be more complex because the operator needs to interact with the GEB and the Database

The following table summarizes the previous considerations:

EIs	**Complexity**	**FPs**
Login/Logout	Low	2 x 3
Reservation	High	6
Delate reservation	Low	3
User Registration	Avg	4
Consult historical reservation info, ride info, payment info	Avg	3 x 4
Insert delate and update (special) safe areas	Avg	3 x 4
Insert, delate, update operators' info	Low	3 x 3
Insert, delate, update cars' info	High	3 x 6
Answer to request of maintenance	Low	3
Change the state of a car	Avg	4
Total		**77**

2.1.4 External Inquiries (EQs)

An inquiry is, referencing to the FP guidelines, a request of data retrieval performed by a user. PowerEnJoy has a few of these inquiries:

1) A user can retrieve information on his profile.

2) A user can search a car by inputting his position or a certain position and obtain info on the cars in the surroundings, and further information about the cars.

3) A user can search the position of special safe areas.

4) A user can obtain the information about his last ride.

5) An administrator can retrieve information about cars, special safe areas, operators and users.

All but number 2), which is fairly complex, are easy to do. This is the resulting table:

EQs	**Complexity**	**FPs**
User retrieve profile info	Low	3
User searches cars	Avg	4
User searches special safe areas	Low	3
User requests last ride info	Low	3
Admin retrieves car info	Low	3
Admin retrieves safe/special safe areas info	Low	3
Admin retrieves operator's info	Low	3
Admin retrieves user's info	Low	3
Total		**25**

2.1.5: External Outputs (EOs)

The external outputs are for the largest part messages and notification to users and operators, and as such they are pretty simple to be done.

- Notify an operator of a recharging request.
- Notify user of the ending of the ride.
- Notify user that he has been blocked.
- Notify user of penalties on the last ride.
- Notify the user that his reservation has been registered.
- Notify user of reservation time expiration.

This is the table for the External Outputs:

EOs	Complexity	FPs
Notification recharging request	Low	4
Notification of ride ending	Low	4
Notification user blocked	Low	4
Notification penalties	Low	4
Notification reservation OK	Low	4
Notification reserve time expiration	Low	4
Total		**24**

2.1.6: Overall estimation

The following table summarizes the results obtained in the precedent analysis:

Function Type	Value
Internal Logic Files	66
External Logic Files	22
External Inputs	77
External Inquiries	25
External Outputs	24
Total	**214**

We will use Java Enterprise Edition to develop our system, so we can estimate the source line of code, basing on the Functional Points, already calculated.

SLOC (lower bound) = 214 * 46= 9844

SLOC (upper bound) = 214* 67= 14338

2.2: Cost and effort estimation, COCOMO II

The cost and effort estimation is one of the most delicate part of a project plan, for this reason we will follow the COCOMO II approach to obtain a reasonable estimation.

2.2.1: Scale drivers

- *Precedentedness*: largely unprecedented, since our team has very few experience in the development of projects of big dimension. **Low→ 4.96 SF** (Scale Factor)
- *Development flexibility*: has an average value, since there are some very important functional requirements that have to be satisfied, but nothing is imposed by the costumer from the point of view of the technologies to use and of the external interfaces that are allowed to be used. **Nominal→ 3.04 SF**
- *Architecture/ Risk resolution*: since we are aware of the possible risks as we performed the risk analysis, we made a clear architectural definition, but we consider that we have no precedent experience in that kind of examination, we set this value to **90% (mostly)→ 1.41 SF**

- *Team cohesion*: our team has a very strong cohesion **seamless interactions→ 0.00 SF**
- *Process maturity*: since we are a very cohesive team but we are at the beginning of our career, we had to overcome some initial difficulties. We can fix this value at **level 3→ 3.12 SF**

In the following table we summarize the previous obtained results:

Scale Driver	Factor	SF Value
Precedentedness (PREC)	Low	4.96
Development flexibility (FLEX)	Nominal	3.04
Risk resolution (RESL)	High	1.41
Team cohesion (TEAM)	Very High	0.00
Process maturity (PMAT)	Level 3	3.12
Total		**12.53**

2.2.2: Cost drivers

In this section we will focus on the definition of the cost drivers related to **post-architecture.**

2.2.2.1: Product Factors cost drivers

- *Required software Reliability (RELY):*
  Since the Power Enjoy system is one of the most important provider of electric cars in the city, the software needs to be very reliable also because a malfunctioning would lead to a **high financial loss** for the company.
- *Database size (DATA):*
  We estimated that the database size will be about 4GB, while the SLOC is about 10.000-15.000. Our ratio D/P (Database bytes/ program SLOC) is between 266 and 400 so we can set this value to **high**.
- *Product Complexity (CPLX)*:
  This value is set to **very high** because the system is composed by complicated control operations, computational operations, device-dependent operations, data management operations and user interface management operations.
- *Developed for reusability (RUSE)*:
  Even if It's always good to develop reusable software, in our case there is no constraint on this objective, so we set this value to **nominal**.
- *Documentation Match to Life-Cycle Needs (DOCU):*
  The relationship between the documentation and the application requirements, for our case is quite equilibrated, so we set this value to **nominal** value.

Descriptor	Rating Level	Effort Multiplier
RELY	Very High	1.26
DATA	High	1.14
CPLX	Very High	1.34
RUSE	Nominal (across project)	1.00
DOCU	Nominal (right-sized to life-cycle needs)	1.00
Total multiplier		**1.924776**

2.2.2.2: Platform Factors cost drivers

- *Execution Time Constraint (TIME):*
  This is a measure of the time constraint imposed upon a software. Since the Power Enjoy system is a quite complicated peace of software, we will set this value to **very high (85% use of available execution time).**
- *Storage Constraint (STOR):*
  We set this value to **nominal** since the since modern disk can contain several terabytes of storage, so the gap between the expected amount of storage usage with respect to the availability of the hardware can be very big (<= 50% use of available storage).
- *Platform Volatility (PVOL):*
  The central system will remain almost the same, after the final release, but from a client point of view (the phone app and the car app) there will be at least 1 major release once every six months, so the value is set to **nominal.**

Descriptor	Rating Level	Effort Multiplier
TIME	Very High	1.29
STOR	Nominal	1.00
PVOL	Nominal	1.00

2.2.2.3: Personnel Factors

- *Analyst capability (ACAP)*:
  The analysis for real word applications has been thorough for our problem, so we set this parameter to **high**.
- *Programmer capability (PCAP)*:
  Being this an academic project, this parameter cannot be properly evaluated. We hope our programmers' capabilities are fairly high, so we set this parameter to **high**.
- *Personnel continuity (PCON):*
  The time we can spend on this project is not so much, so we set this parameter to **low.**
- *Applications experience (APEX):*
  We have already developed applications in java, but never in JEE and never for a project of this size, so we set this parameter to **low.**

- *Platform experience (PLEX):*
  We never had experiences with JEE platform, but we have some experience with databases and general java platforms, so we set this parameter to **low.**
- *Language and tool experience(LTEX):*
  We don't have experience with specific JEE language, but some in general java language and we have already used tools for programming in this language, so we set this parameter to **nominal.**

Descriptor	Rating level	Effort Multiplier
ACAP	High	0.85
PCAP	High	0.88
PCON	Low	1.12
APEX	Low	1.10
PLEX	Low	1.09
LTEX	Nominal	1.00

2.2.2.4: Project Factors cost drivers

- *Use of Software Tool (TOOL):*
  Our application environment is strong, proactive and well integrated, so this value will be set to **very high.**
- *Multisite Development (SITE):*
  Even if we usually work together, we used internet, emails, social network and other interactive multimedia to keep working also during the weekend, when we were far. This value is set to **very high.**

Descriptor	Rating Level	Effort Multiplier
TOOL	Very High	0.78
SITE	Very High	0.86

2.2.2.5: General Factor cost drivers

- *Required development schedule (SCED):*
  We will set this value to **high**, because we planned to spend more time in the early phase of the development process, even if we will try to well distribute the effort.

Descriptor	Rating Level	Effort Multiplier
SCED	High	1.00

2.2.3: Effort equation

Here we calculate the Effort estimated measured in Person-Months (PM) through the final equation:

$$EFFORT = A * EAF * KSLOC^{E}$$

Where:

A= 2.94 (for COCOMO II)

EAF=product of all cost drivers (1.68692)

E = exponent derived from the scale drivers. It is computed as:

B+ 0.01 * Σi SF[i]=B+ 0.01 * 12.53 = 0.91 + 0.1253= 1.0353

in which B is equal to: 0.91 for COCOMO II.

Now we can compute the Effort:

EFFORT (lower bound) = A * EAF * $KSLOC^E$= 2.94 * 1.68692 * $9.844^{1.0353}$ = 53.237 PM ≈ 54 PM

EFFORT (upper bound) = A * EAF * $KSLOC^E$= 2.94 * 1.68692 * $14.338^{1.0353}$ = 78.119 PM ≈ 79 PM

2.2.4: Schedule estimation

Here we will have an estimation of the months/person that will be needed to complete the project.

We will use this formula to calculate the duration:
Duration = 3.67 *$Effort^F$

Where F = 0.28 + 0.2*(E-B) = 0.28 + 0.2*(1.0353 -0.91) = 0.28 + 0.2*(0.1253) = 0.30506

We can now compute the duration:

Duration (lower bound) = 3.67*(EffortLowerBound)^F = 3.67*$(53.237)^{0.30506}$ = 12.33 months

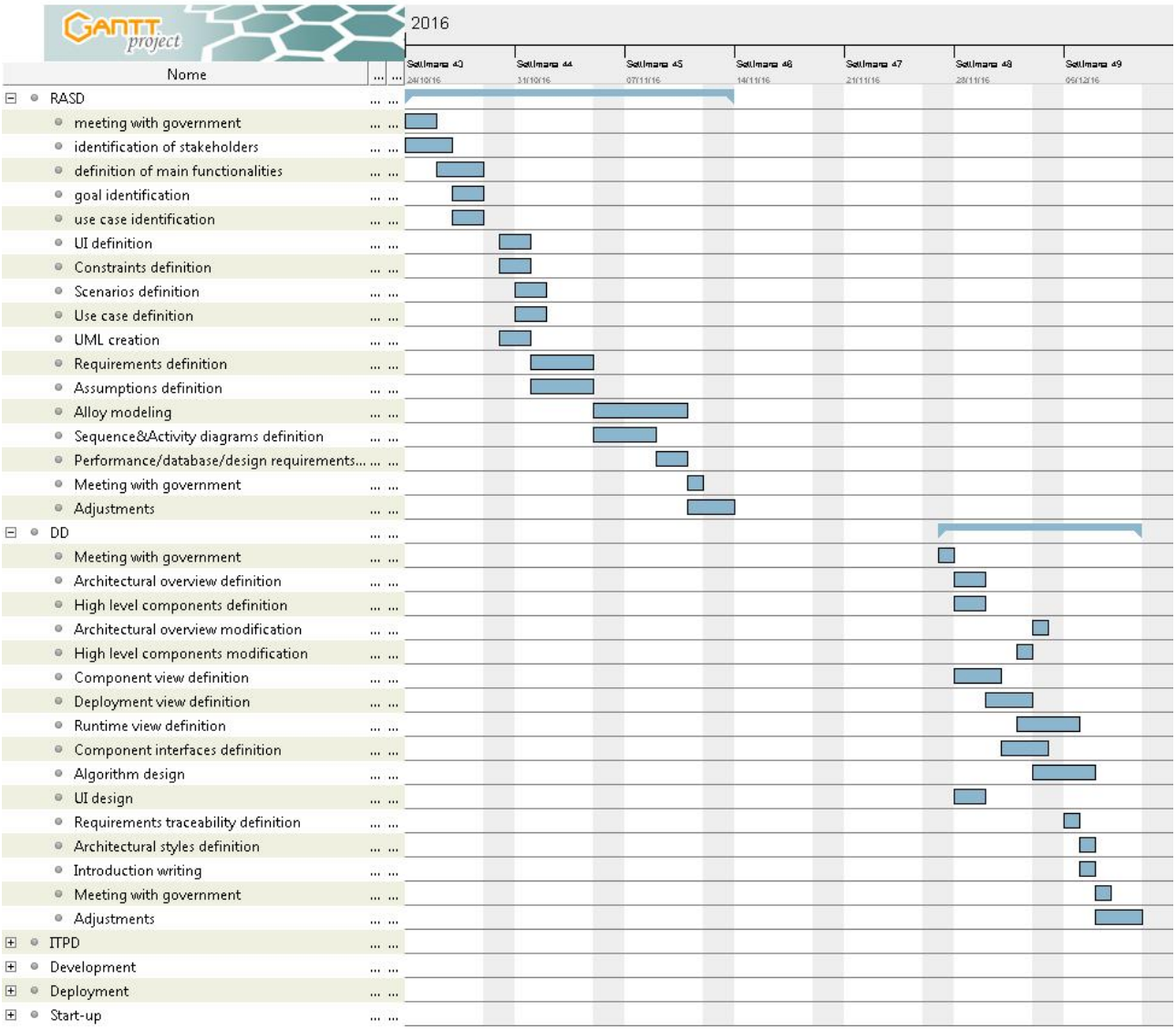
Duration (upper bound) = 3.67*(EffortLowerBound)^F = 3.67*$(78.119)^{0.30506}$ = 13.86 months

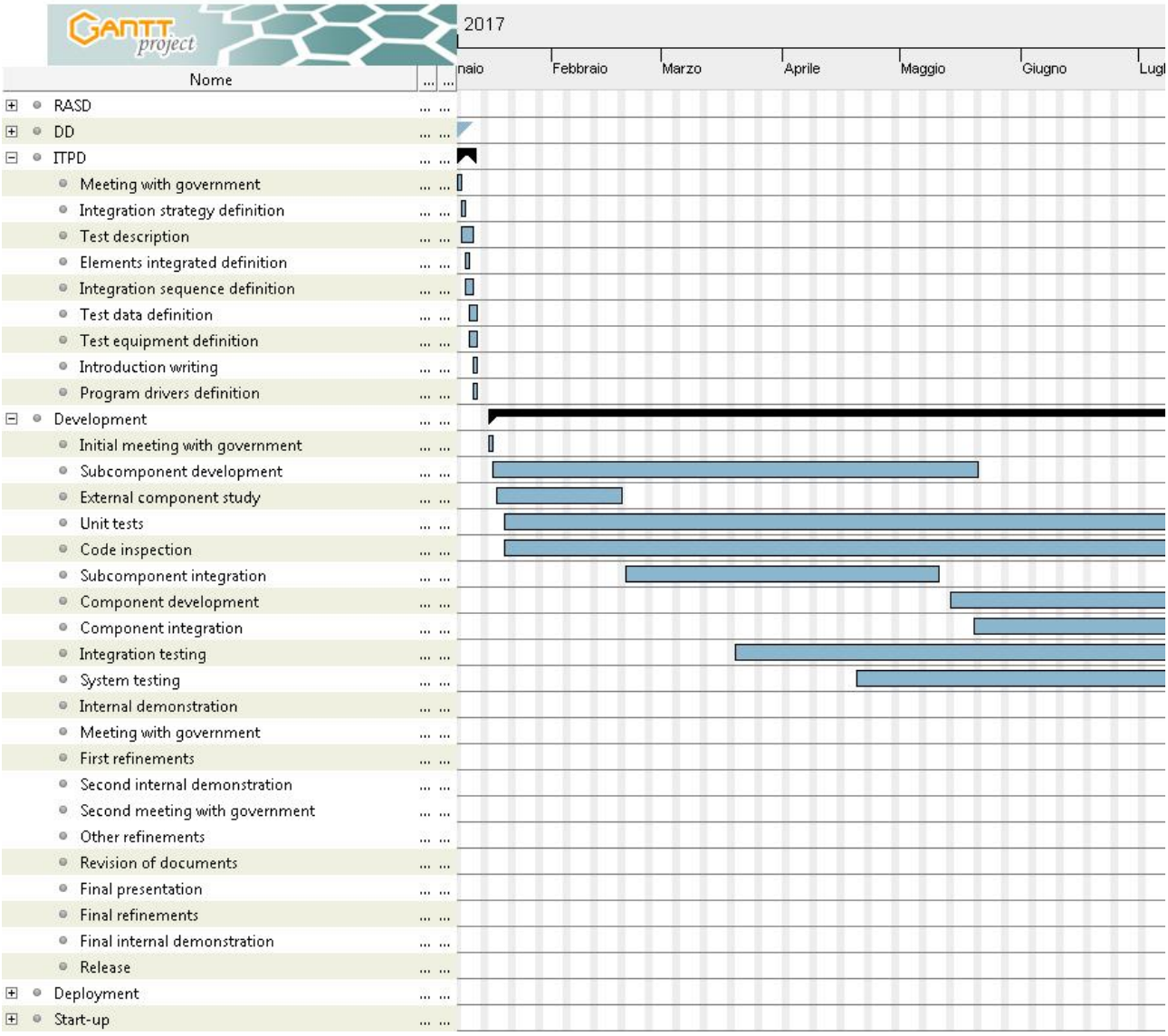
Which are reasonable estimates.

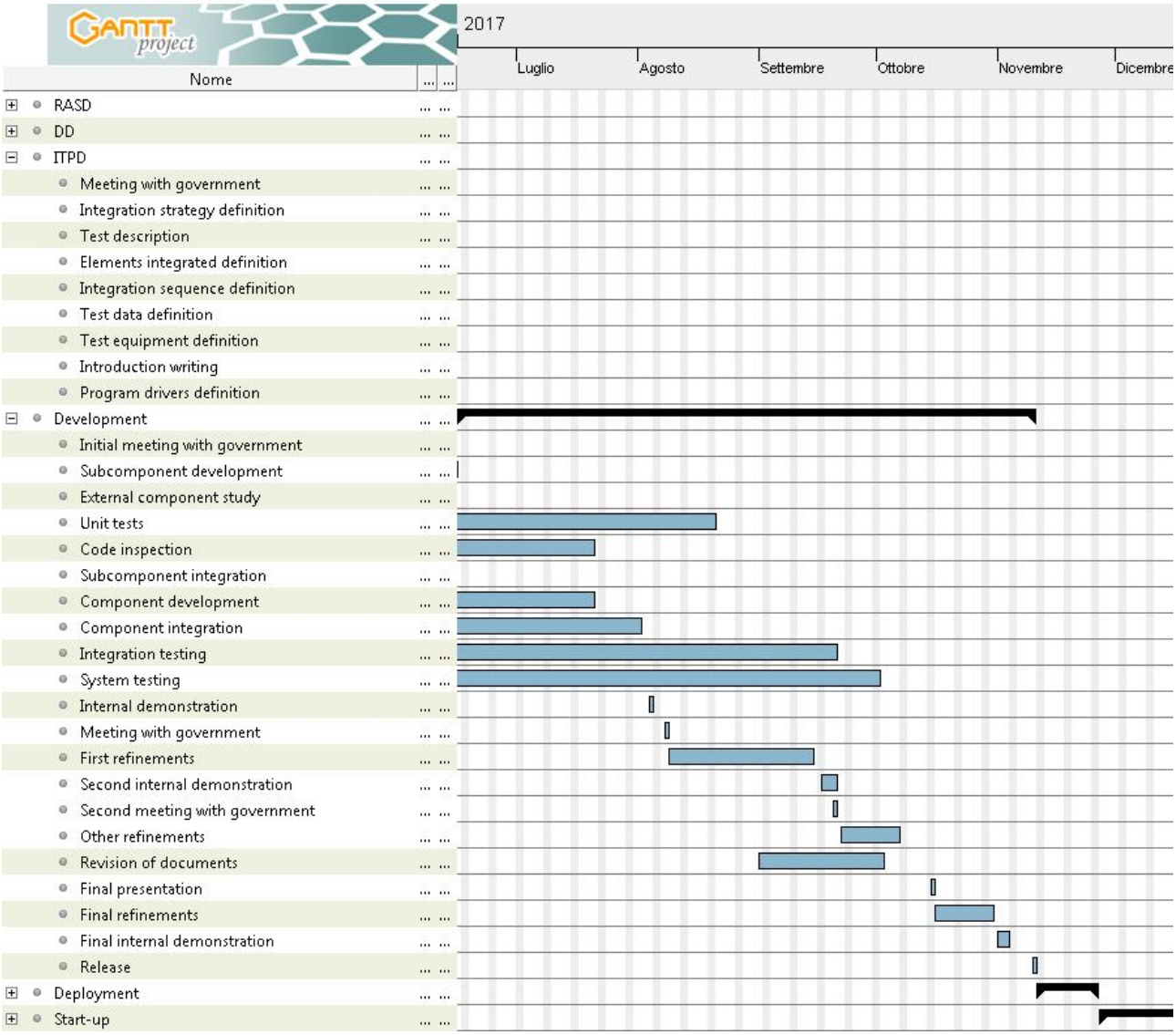
3: Schedule

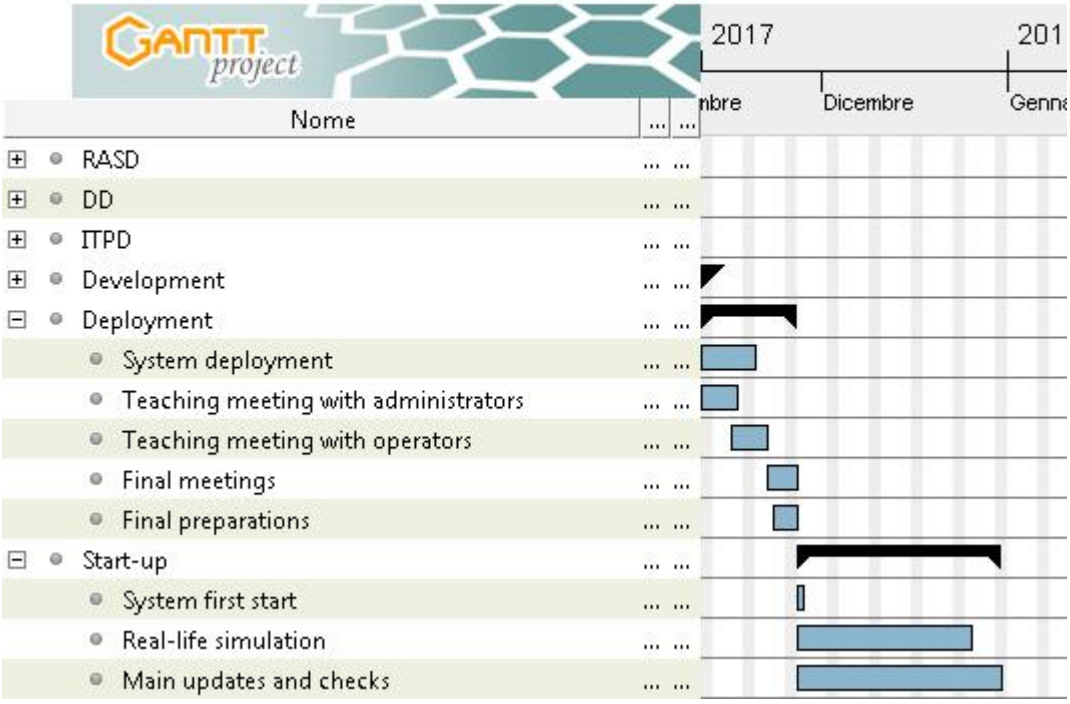
This chapter will contain a first draft of schedule for the development of our system. Please note that this schedule could not be completely followed and discrepancies are ought to appear; further schedules will be drafted during the project.

Being this an academic work and being this a document done after we've effectively done the documents written in the schedule, we tried to determine a schedule more or less based on what we have really done for the project.



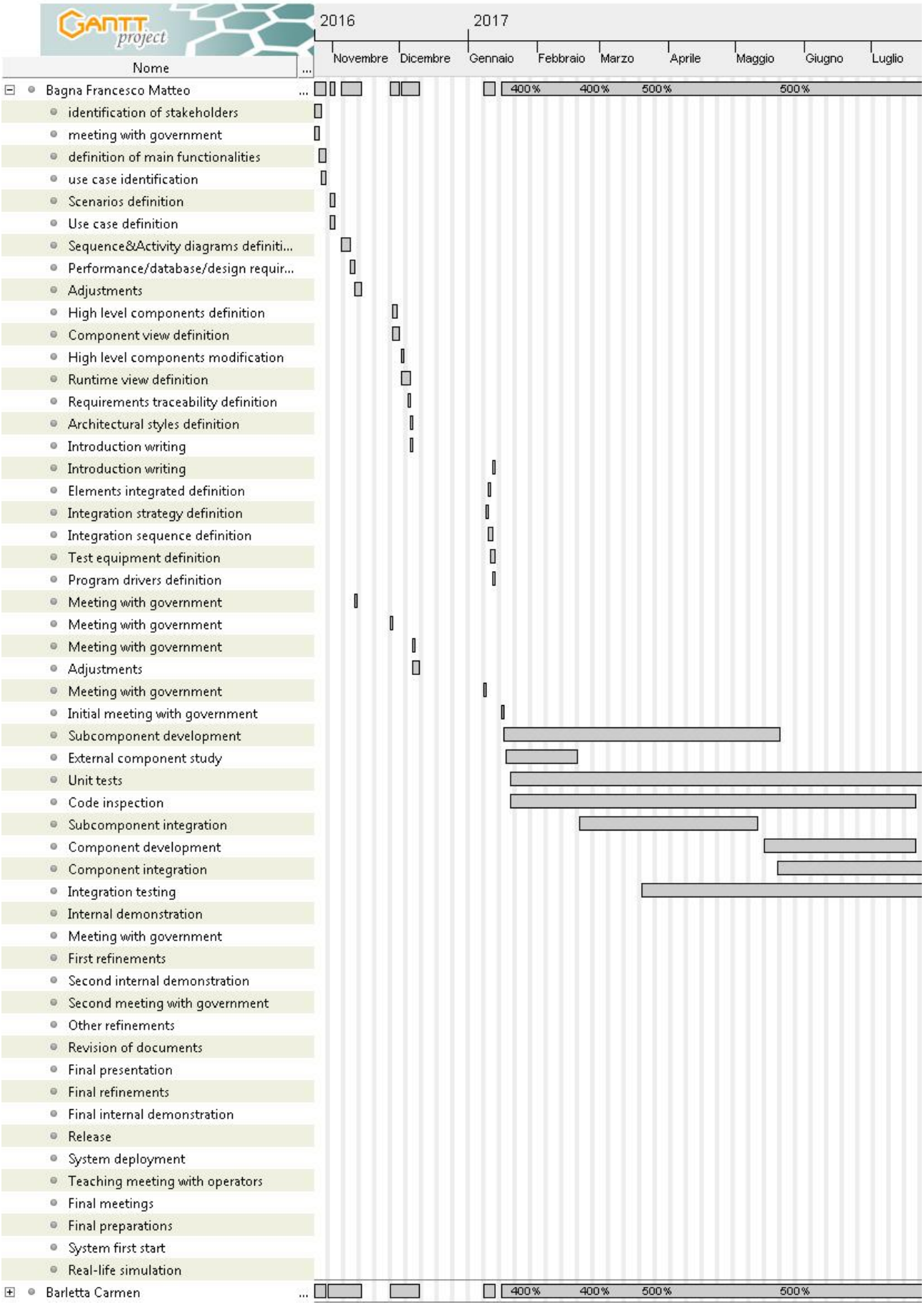

GANTT project
2017
naio
Febbraio
Marzo
Aprile
Maggio
Giugno
Lugl
Nome
RASD
DD
ITPD
Meeting with government
Integration strategy definition
Test description
Elements integrated definition
Integration sequence definition
Test data definition
Test equipment definition
Introduction writing
Program drivers definition
Development
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
System testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Other refinements
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
Deployment
Start-up

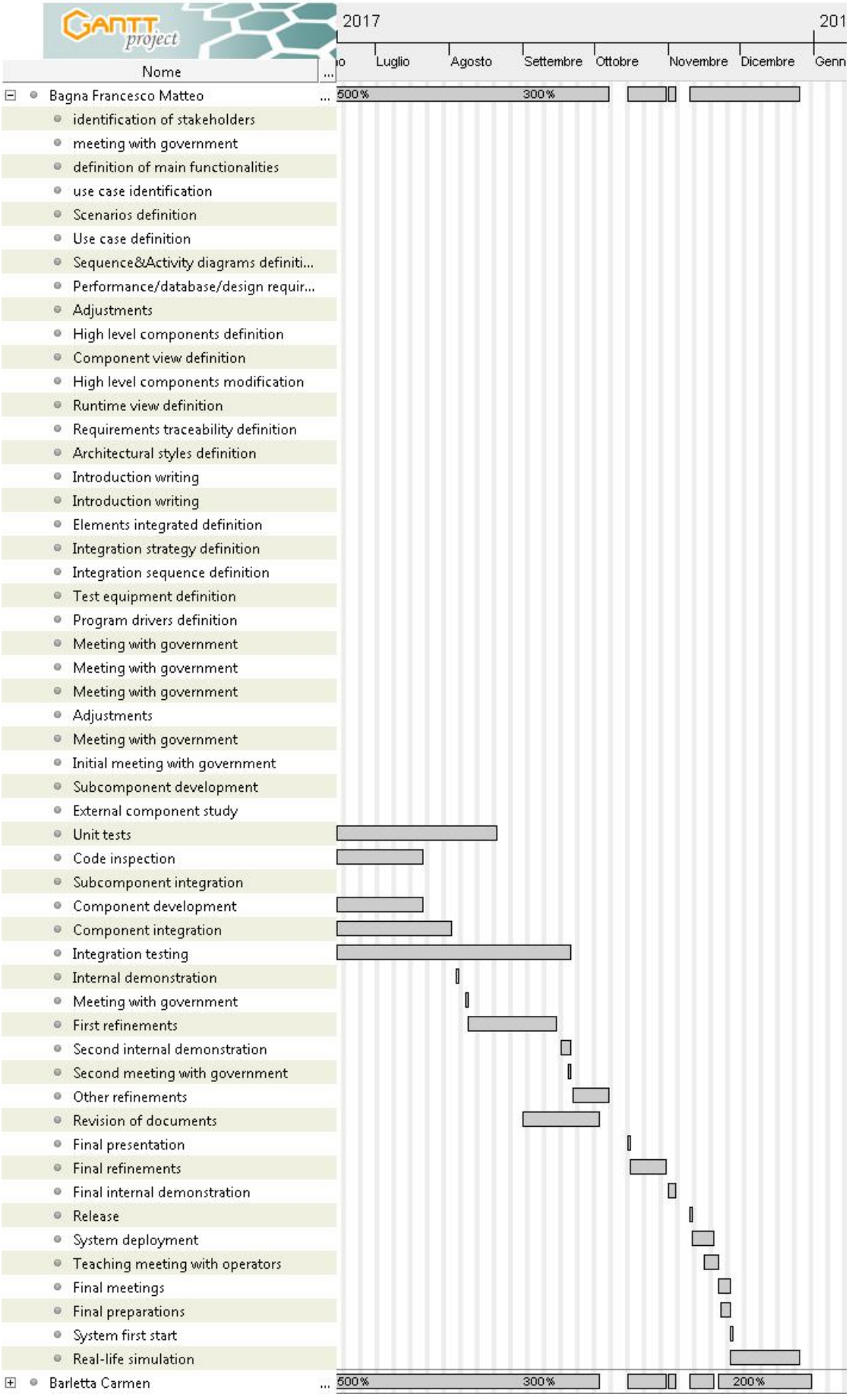

GANTT project
2017
Luglio
Agosto
Settembre
Ottobre
Novembre
Dicembre
Nome
RASD
DD
ITPD
Meeting with government
Integration strategy definition
Test description
Elements integrated definition
Integration sequence definition
Test data definition
Test equipment definition
Introduction writing
Program drivers definition
Development
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
System testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Other refinements
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
Deployment
Start-up

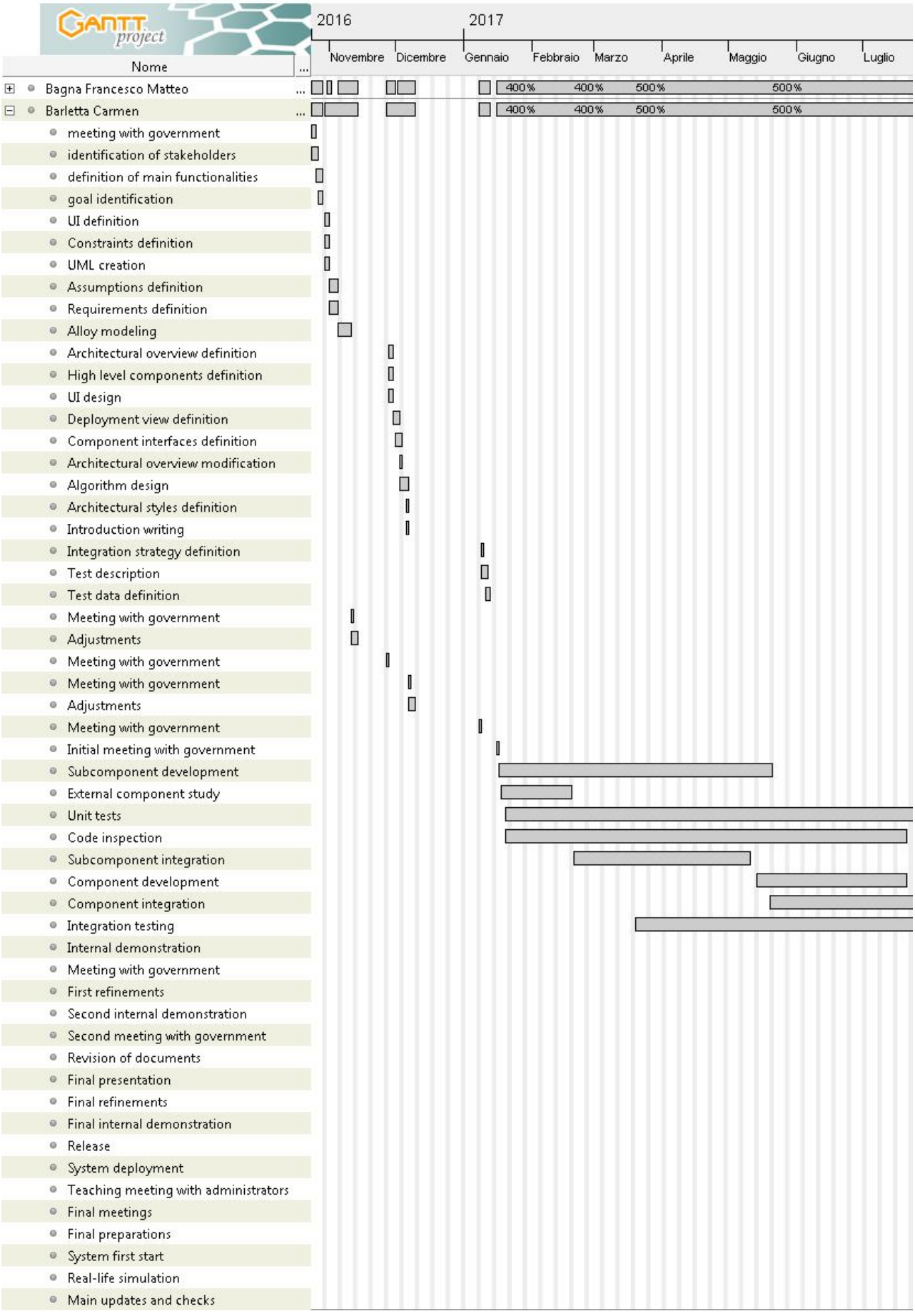

GANTT project
2017
201
nbre
Dicembre
Genna
Nome
RASD
DD
ITPD
Development
Deployment
System deployment
Teaching meeting with administrators
Teaching meeting with operators
Final meetings
Final preparations
Start-up
System first start
Real-life simulation
Main updates and checks

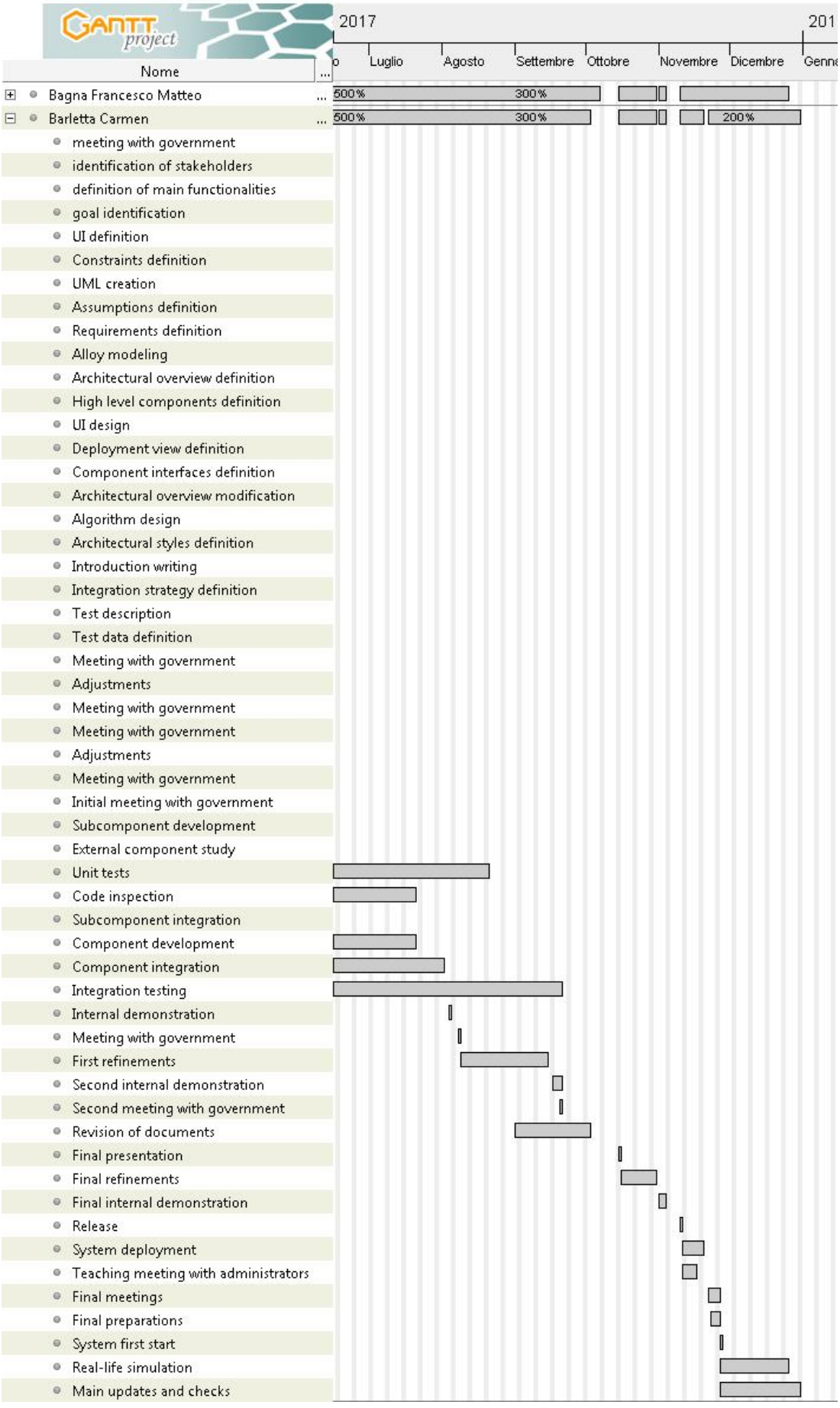
4: Resource allocation

In this chapter we will see how the tasks shown in the previous chapter will be allocated to the various individuals that participate in the project. Like already mentioned, these tasks will be based on our actual work for the project.


GANTT project
2016
2017
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Nome
Bagna Francesco Matteo
identification of stakeholders
meeting with government
definition of main functionalities
use case identification
Scenarios definition
Use case definition
Sequence&Activity diagrams definiti...
Performance/database/design requir...
Adjustments
High level components definition
Component view definition
High level components modification
Runtime view definition
Requirements traceability definition
Architectural styles definition
Introduction writing
Introduction writing
Elements integrated definition
Integration strategy definition
Integration sequence definition
Test equipment definition
Program drivers definition
Meeting with government
Meeting with government
Meeting with government
Adjustments
Meeting with government
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Other refinements
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
System deployment
Teaching meeting with operators
Final meetings
Final preparations
System first start
Real-life simulation
Barletta Carmen
400%
400%
500%
500%
400%
400%
500%
500%


GANTT project
2017
201
Luglio
Agosto
Settembre
Ottobre
Novembre
Dicembre
Genn
Nome
Bagna Francesco Matteo
500%
300%
identification of stakeholders
meeting with government
definition of main functionalities
use case identification
Scenarios definition
Use case definition
Sequence&Activity diagrams definiti...
Performance/database/design requir...
Adjustments
High level components definition
Component view definition
High level components modification
Runtime view definition
Requirements traceability definition
Architectural styles definition
Introduction writing
Introduction writing
Elements integrated definition
Integration strategy definition
Integration sequence definition
Test equipment definition
Program drivers definition
Meeting with government
Meeting with government
Meeting with government
Adjustments
Meeting with government
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Other refinements
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
System deployment
Teaching meeting with operators
Final meetings
Final preparations
System first start
Real-life simulation
Barletta Carmen
500%
300%
200%


GANTT project
2016
2017
Novembre
Dicembre
Gennaio
Febbraio
Marzo
Aprile
Maggio
Giugno
Luglio
Nome
Bagna Francesco Matteo
Barletta Carmen
400%
400%
500%
500%
meeting with government
identification of stakeholders
definition of main functionalities
goal identification
UI definition
Constraints definition
UML creation
Assumptions definition
Requirements definition
Alloy modeling
Architectural overview definition
High level components definition
UI design
Deployment view definition
Component interfaces definition
Architectural overview modification
Algorithm design
Architectural styles definition
Introduction writing
Integration strategy definition
Test description
Test data definition
Meeting with government
Adjustments
Meeting with government
Meeting with government
Adjustments
Meeting with government
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
System deployment
Teaching meeting with administrators
Final meetings
Final preparations
System first start
Real-life simulation
Main updates and checks


GANTT project
2017
201
Luglio
Agosto
Settembre
Ottobre
Novembre
Dicembre
Nome
Bagna Francesco Matteo
500%
300%
Barletta Carmen
500%
300%
200%
meeting with government
identification of stakeholders
definition of main functionalities
goal identification
UI definition
Constraints definition
UML creation
Assumptions definition
Requirements definition
Alloy modeling
Architectural overview definition
High level components definition
UI design
Deployment view definition
Component interfaces definition
Architectural overview modification
Algorithm design
Architectural styles definition
Introduction writing
Integration strategy definition
Test description
Test data definition
Meeting with government
Adjustments
Meeting with government
Meeting with government
Adjustments
Meeting with government
Initial meeting with government
Subcomponent development
External component study
Unit tests
Code inspection
Subcomponent integration
Component development
Component integration
Integration testing
Internal demonstration
Meeting with government
First refinements
Second internal demonstration
Second meeting with government
Revision of documents
Final presentation
Final refinements
Final internal demonstration
Release
System deployment
Teaching meeting with administrators
Final meetings
Final preparations
System first start
Real-life simulation
Main updates and checks

5: Risk Management

Many kind of risks are to be faced when medium to large scale projects are involved.
In our case, the risks could be of various natures: technical, political and financial.
Starting with the first, we may find a lot of technical difficulties.
First of all, the participants in the project may not be as expert as we thought they would be, or they could get ill, even for a long period of time and, in the worst case, near to a key release of the project. This issues can be taken care of by hiring people that have a good curriculum vitae, or with whom we already worked with; more time than necessary can be added at the end of each key phase of the project in order to do adjustments in case some people got ill.
Another classic but fatal technical issue could be the loss of source code: this issue can be easily covered by using algorithms of backup distributed over more than one location.
Also, a change in the terms of agreements for some APIs may prove a great challenge in case it did not let us continue to use the service or external component, because we would have to find some other substitutive service, which could prove difficult especially in little time, and especially if it was the GEB system that changed its terms. Also the change of pricing of the usage of external components (if not already paid in full) or of the cloud service (we remind the readers that our server heavily uses a cloud system) could cause a great deal of financial problems. We can only try to design the components as portable as possible, with little dependencies, so to make more easy the sift to other systems or services.
Another technical problem could be the breakage of the computers used by the operators and of the cars or the GEB system in one of the cars, but these issues can be easily managed by having a good number of substitute computers and a good number of cars.
Another problem could be the project schedule, that could be not properly followed as defined in the precedent chapters: there could be problems or delays. As such, some time for adjustments will need to be taken in account before the main key points of our project, in order to recover the eventually lost time.
Political problems may involve the stakeholders, namely the government of the city we are doing the project for. Changes of priorities may reduce our budget, or a new government not interested in our project may be established, or some other company could try to begin a project like ours and we could lose the favor of the government of the city; all these issues are problematic, and we can only try to involve as much as possible the stakeholders in our project, so to make them happy about our work. The requests could be difficult to meet, but negotiations can be made. As such, at the beginning and at the end (in some cases even in between) of the key parts of the project we will have meetings with the local government in order to hear what we could modify in our project.
Other problems may arise if the requirements for cars and equipment (like the standards for GPS systems) were changed: a more independent and portable code will help us migrate to other systems in case some problems of this kind happen.
Changes in the traffic laws concerning the usage of electric cars and of our equipment could also be a problem; in order to fight these problems, we can hope to have some time to adapt ourselves

before the laws come in vigor.
Also some problems related to the vendors of the technology we use, like computers, car and the GEB system may arise. The first two of these complications can be easily overtaken, being the number of vendors really high; if we were to have problems with the GEB vendor we would have to change completely how the system of sensors and data would function, along with the Ruby on Rails framework that we used. The best solution to this problem is again to code what we can in the most portable way, in order to ease the passage to another system, if necessary.

6: Effort spent

6.1: Hours of work

6.1.1: Barletta Carmen

16/01/17 → 1h

17/01/17 → 1h

18/01/17 → 4h

19/01/17 →3h

6.1.2: Bagna Francesco Matteo

16/01/17 → 1h

17/01/17 → 2h30m

19/01/17→ 2h

20/01/17 → 3h30m

Appendix A: Function points 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 Inquiries

	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 Inputs

	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 weights		
Function types	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

Appendix B: COCOMO II tables

For scale drivers

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	Thoroughly unprecedented	Largely unprecedented	Somewhat unprecedented	Generally familiar	Largely familiar	Thoroughly familiar
SFj	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	Rigorous	Occasional relaxation	Some relaxation	General conformity	Some conformity	General goals
SFj	5.07	4.05	3.04	2.03	1.01	0.00
RESL	Little (20%)	Some (40%)	Often (60%)	Generally (75%)	Mostly (90%)	Fully (100%)
SFj	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
SFj	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
SFj	7.80	6.24	4.68	3.12	1.56	0.00

For Cost drivers

RELY Cost Drivers						
RELY descriptors	Slightly inconvenience	Easily recoverable losses	Moderate recoverable losses	High financial loss	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

DATA Cost Drivers						
DATA descriptors		D/P < 10	10 ≤ D/P ≤ 100	100 ≤ D/P ≤ 1000	D/P ≥ 1000	
Rating level	Very low	Low	Nominal	High	Very High	Extra High
Effort multipliers	n/a	0.90	1.00	1.14	1.28	n/a

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

RUSE Cost Drivers						
RUSE descriptors		None	Across project	Across program	Across product line	Across multiple product lines
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

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

TIME Cost Drivers						
TIME descriptors			≤ 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

STOR Cost Drivers						
STOR descriptors			≤ 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

PVOL Cost Drivers						
PVOL descriptors		Major change every 12 mo., minor change every 1 mo.	Major: 6mo., Minor: 2wk	Major: 2mo., Minor: 1wk	Major: 2wk, Minor: 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

ACAP Cost Drivers						
ACAP descriptors	15^{th} percentile	35^{th} percentile	55^{th} percentile	75^{th} percentile	90^{th} 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

PCAP Cost Drivers						
PCAP descriptors	15^{th} percentile	35^{th} percentile	55^{th} percentile	75^{th} percentile	90^{th} 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

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

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

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

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

TOOL Cost Drivers						
TOOL descriptors	Edit, code, debug	Simple, fronted, 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

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 communica-tions descriptors	Some phone, mail	Individual phone, fax	Narrow band email	Wideband electronic communi-cation	Wideband elec. comm., occasional 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

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