Politecnico Di Milano
A.A. 2015/2016
Software Engineering 2:
Project Plan Document

Paramithiotti Andrea (Matr. 788794)
Rompani Andrea (Matr. 854052)
Zoia Lorenzo (Matr. 852392)

v1.0
02/02/2016

Contents

1 Introduction

1.1 Revision History

Version	date	Modifications
1.0	2016/2/22	First version

1.2 Purpose and Scope

1.2.1 Purpose

This document points to evaluate the time and resources necessary to the development of the “MyTaxiService” application. The Functional Point approach has been used for the estimation of the project size, and the COCOMO model for the effort and cost. Furthermore, the project has been dived into tasks that have been distributed among the development team members.

1.2.2 Scope

The aim of the project is to create a brand new system for the management and the organization of a city taxi service. This system offers a mobile application and a web interface in order to give the customers the possibility to benefit from the taxi service. Furthermore, the system provides an additional communication interface for the taxi drivers.
The mobile application and the web interface accept requests and reservations for taxis from the users, with the possibility to organise a share ride among different users. The taxi driver is supposed to communicate his availability, acceptances and rejections of requests through the communication interface.
The system is created to simplify the access of passengers to the service and to guarantee a fair management of the taxi queues.

2 Cost and Effort Estimation

2.1 Function Point Approach

2.1.1 Introduction

The Function Point estimation approach is based on the amount of functionalities in a software and their complexity. Function Point estimators are useful since they are based on information that are available early in the project life cycle.
In order to perform this estimation, it is necessary to identify these functional type in the project: Internal Logic File, External Logic File, External Input, External Output, and External Inquiry. Then for each of these types, it is needed to define the complexity and evaluate the cost using the appropriate weight.

This table was used to define the weights value:

Function Types	**Weight**		
	Simple	Medium	Complex
N. Inputs	3	4	6
N. Outputs	4	5	7
N. Inquiry	3	4	6
N. ILF	7	10	15
N. EIF	5	7	10

2.1.2 FP Estimation

2.1.2.1 Internal Logic File

The system includes a number of ILFs that are used to save and manage information about:

- **User** The system saves only these simple information: name, surname, phone number, mail address, username and password
- **Taxi Driver** The system stores and uses these information: personal information like the user, id, and the related information (id, position) about the associated taxi
- **City Zone** The system uses the geographical information and the adjacency among each zone. Due to the possible high complexity of the data structure storing the adjacency this ILF can be considered Medium
- **Request** The information stored for each reqest are different depending on the type:

    - **Normal Request** Origin and identifier of the requestor
    - **Reservation** Origin, identifier of the requestor, date and time
    - **Shared Request** Origin, identifier of the requestor and destination
- **Ride** The system stores the total fee, the personal info and the info of the associated taxi

Here are shown the complexity and the related weights based on the information written above.

ILF	Complexity	FP
User	Simple	7
Taxi Driver	Medium	10
City Zone	Medium	10
Request	Medium	10
Ride	Simple	7

Total FP for the ILF: 44

2.1.2.2 External Interface File

The system asks to an external maps provider only the route during the process to evaluate if a new user could join or not an existing shared ride. The complexity of this data strongly depends on the origin/destination point, so the complexity could be evaluated in average medium.

Total FP for the ILF: 7

2.1.2.3 External Input

The system interacts both with the user and the taxi driver.

User operations

- **Registration/Authentication** These are simple operations related to the possibility to access/register to the system. So the total contribution is : **2x3=6**
- **Request a Taxi** This is the main functionality, the system has to generate the request and handle it immediately in order to quickly send a taxi to the user. So it is possible to consider it complex. The total contribution is : **6**
- **Reserve/Delete a Taxi** This operation simply consists in creating/deleting a request. So the total contribution is : **2x3=6**
- **Request a Shared Ride** This is the most complex operation related to the user. In fact, the system has to calculate both if there are other shared rides joinable and the personal fee of each passenger. So the total contribution is : **6**

Taxi Driver operations

- **Set Availability** This operation can be considered as medium. If the taxi driver wants to become available the system has to compute in which queue zone he must be stored. So the total contribution is : **4**
- **End a Ride** This operation is very similar to the availability one, because the system has to compute in which queue zone the taxi driver must be stored. So the total contribution is : **4**
- **Call Emergency Number / Tow Truck / MyTaxiService Operator** These operations are all simple because they are managed directly by the phone application installed in the taxi driver interface. So the total contribution is : **3x3=9**
- **Request Another Taxi** This last operation has the same complexity of the "Request a Taxi" of the user, as the system behaves in the same way. So the total contribution is : **6**

Total FP for the External Input: 47

2.1.2.4 External Inquiries

The system allows an authenticated user to request information about:

- **Personal Information** This feature allows the user to visualize the information stored in the system
- **Pending Reservation** This feature allows the user to visualize all the pending reservations
- **Ride** This feature allows the user to visualize all his rides. It can be a very big number so it can be evaluated as medium complexity

EI	Complexity	FP
Personal Info	Simple	4
Pending Reservation	Simple	4
Ride	Medium	6

Total FP for the EI: 14

2.1.2.5 External Output

The system does not provide any services that can be considered as External Output.

Total FP for the EO: 0

2.1.3 Conclusions

Type	FP
ILF	44
EIF	7
External Input	47
EI	14
EO	0
TOTAL	112

In order to complete the analysis, it is necessary to convert the FP estimation into lines of code. In order to do that it's fundamental to know the language used for the development of the project. Due to the fact that the project was not developed, it was decided to assume that it was written in JEE because the internet-oriented nature of this language is suitable for the project.

Referring to the official FP table http://www.qsm.com/resources/function-point-languages-table, the multiplicative value (SLOC/UFP) for JEE is 46.
It is possible to convert FP into source lines of code (SLOC) using this formula:

$$SLOC = UFP * \frac{SLOC}{UFP} = 112 * 46 = 5152$$

2.2 COCOMOII Approach

2.2.1 Introduction

The COCOMOII approach uses the SLOC size of the project in order to evaluate the effort, the duration and the number of required people, taking in account the characteristics of the project and the staff.

In order to compute the estimation there are two values to define: the Scale Drivers and the costs Driver contribution.

All the tables used in this analysis were extracted from the COCOMOII Model Definition Manual:
http://csse.usc.edu/csse/research/COCOMOII/cocomo2000.0/CII_modelman2000.0.pdf

2.2.2 Scale Drivers

This table was used to estimate the Scale Drivers value

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	The estimated Equivalent Process Maturity Level (EPML) or					
	SW-CMM Level 1 Lower	SW-CMM Level 1 Upper	SW-CMM Level 2	SW-CMM Level 3	SW-CMM Level 4	SW-CMM Level 5
SF_j:	7.80	6.24	4.68	3.12	1.56	0.00

- **Precedentedness (PREC)**
  This is the first project of this type that the team have done, so the PREC is set to Low

- **Development Flexibility (FLEX)**
  Due to the fact that the Requirements were assigned with a general high level of detail, the FLEX is set to High

- **Risk Resolution (RESL)**
  The RESL value is set to Nominal because the risk management was done only textually in this document

- **Team Cohesion (TEAM)**
  This is not the first project that the team has done before, so they know each other's capabilities and developing skills. The TEAM value is set to Very High

- **Process Maturity (PMAT)**
  The PMAT value is set to nominal. It cannot be evaluated due to the fact that the project was not developed

Scale Driver	Complexity	Value
PREC	Low	4.96
FLEX	High	2.03
RESL	Nominal	4.24
TEAM	Very High	1.10
PMAT	Nominal	4.68
TOTAL		17.01

2.2.3 Costs Driver

All the values below were extracted from the COCOMOII Manual described in the Introduction.

- **Required Software Reliability (RELY)**
  The reliability of the software is an important point of the system, so RELY is set to High because lost data or failures lead to financial loss. For example, if a user cannot call a taxi, he will not use the service again and will contact another company

- **Data Base Size (DATA)**
  This value is set to Nominal because we cannot measure the DP value

- **Product Complexity (CPLEX)**
  This value is set to Nominal because it is very difficult to estimate without development

- **Developed for Reusability (RUSE)**
  The nature of the design of the system is a high reusability of the components for future upgrades, so the RUSE is set to High

- **Documentation Match to Life-Cycle- Needs (DOCU)**
  The documentation was written in order to cover all the problems that would occur, without loss of generality. So the DOCU value is set to Nominal

- **Execution Time Constraint (TIME)**
  No time constraints were imposed to the development of the system so the TIME value is set to Low

- **Main Storage Constrain (STOR)**
  The STOR value is set to Nominal because the performance and the quality of the hardware that could be used for this system were not discussed

- **Platform Volatility (PVOL)**
  In this system it is reasonable to consider as platforms the DBMS, the mobile OS where the apps run, the Browser and the Operating system where the server applications run. On average, all this components could change every 12 months, so the PVOL is set to Low

- **Analyst Capability (ACAP)**
  This is the first project where it was requested to the team to design and analyse the requirements. So the PCAP value is set to Nominal only because it was spent a lot of time to evaluate the Requirements

- **Programmer Capability (PCAP)**
  Only one of the team members has a high experience in programming, the other two have done only university projects. So the PCAP is set to Nominal

- **Personnel Continuity (PCON)**
  The PCON value is set to Very Low because all the team members are students

- **Applications Experience (APEX)**
  This project is the first project of this type that was developed by the team, so the APEX is set to Very Low

- **Platform Experience (PLEX)**
  As for the APEX the PLEX is set to Low

- **Language and Tool Experience (TLEX)**
  Although this is the first project, the team already knows the language (supposed to be JEE) and the Tools (Net Beans). So the TLEX is set to Nominal

- **Use of Software Tools (TOOL)**
  The TOOL is set to Nominal because if Net Beans could be used, the team would know his functionality quite well

- **Multisite Development (SITE)**
  The evaluation of this parameter was based on the production of the previous deliveries. In order to write them, the team used phones, mails, Skype and screen sharing, so SITE is set to Extra High

- **Required Development Schedule (SCED)**
  Like for the SITE, the SCED evaluation is based on the deliveries. The schedule was defined at the beginning of the project, so the team had to respect deadlines. For all this reasons the SCED is set to High

Cost Driver	Complexity	Value
RELY	High	1.10
DATA	Nominal	1.00
CPLEX	Nominal	1.00
RUSE	High	1.07
DOCU	Nominal	1.00
TIME	Low	n/a
STOR	Nominal	1.00
PVOL	Low	0.87
ACAP	Nominal	1.00
PCAP	Nominal	1.00
PCON	Very Low	1.29
APEX	Very Low	1.22
PLEX	Low	1.09
TLEX	Nominal	1.00
TOOL	Nominal	1.00
SITE	Extra High	0.80
SCED	High	1.00
TOTAL		1.4

2.2.4 Effort Evaluation

This formula was used in order to evaluate the Effort:

$$EFF = 2.94 * EAF * (KSLOC)\text{^}E$$

Where:

- **EFF** Effort of the system
- **EAF** Effort Adjustment Factor derived from product of all the Cost Drivers, in this case 1.4
- **KSLOC** Kilo Source Line of Code in this case 5.152
- **E** Exponent derived from the Scale Driver with this formula:

$$0.91 + 0.01 * (\sum SD) = 1.08$$

So in this case the EFF is equal to 24.18 PM

2.2.5 Schedule Estimation

It is possible to calculate the Duration with this formula

$$Duration = 3.67 * EFF\text{^}SE$$

Where

- **E** is calculated with this formula

$$SE = 0.28 + 0.2 * (E - 0.91) = 0.314$$

The Duration is 9.98 Months, so approximately 10 Months

It is possible to calculate the Number of people with this formula

$$Number = Effort/Duration$$

In this case the number of people is 2.42 so approximately 3 person

All the formula used in the two last chapters were extracted from the Model described in the COCOMOII Introduction

3 Tasks

The project is divided into different tasks following the design document.
Every service requestor is a task, as well as every service provider.

In the evaluation of the amount of hours needed for every task, it was considered that every task includes the unit testing for that part.

It was estimated that a certain number of hours and days would be needed for the development of the project. That estimation was made considering that the team members would work eight hours per day.

As the specifics asked to consider the actual availability of the team, another estimation was made counting that the team members would work a maximum of two hours per day, due to lessons attendance and travel time.
In this way the number of days needed for the total development increased fourfold, but obviously not the total amount of hours.

3.1 Tasks definition

ID	Description	Dependencies	Length [days]
T1	Requests manager	T6	7
T2	Shared rides manager	T1	14
T3	Reservations manager	T15	7
T4	Registration manager	T16	7
T5	Authentication manager	T16	10
T6	Taxi management	T15	20
T7	Emergency manager	T6	10
T8	Taxi driver interface	T11,T12,T13	30
T9	Web interface	T11,T12,T13	30
T10	Mobile interface	T11,T12,T13	30
T11	Service broker		10
T12	Notification center		7
T13	Internal message dispatcher	T1,T2,T3,T7,T11	14
T14	Web server	T12	2
T15	Datawarehouse		5
T16	User Database		2
T17	Analyzer	T15	3
T18	Documents creation		25
T19	Integration testing	Everything done	30
T20	Interfaces translation	T18	2

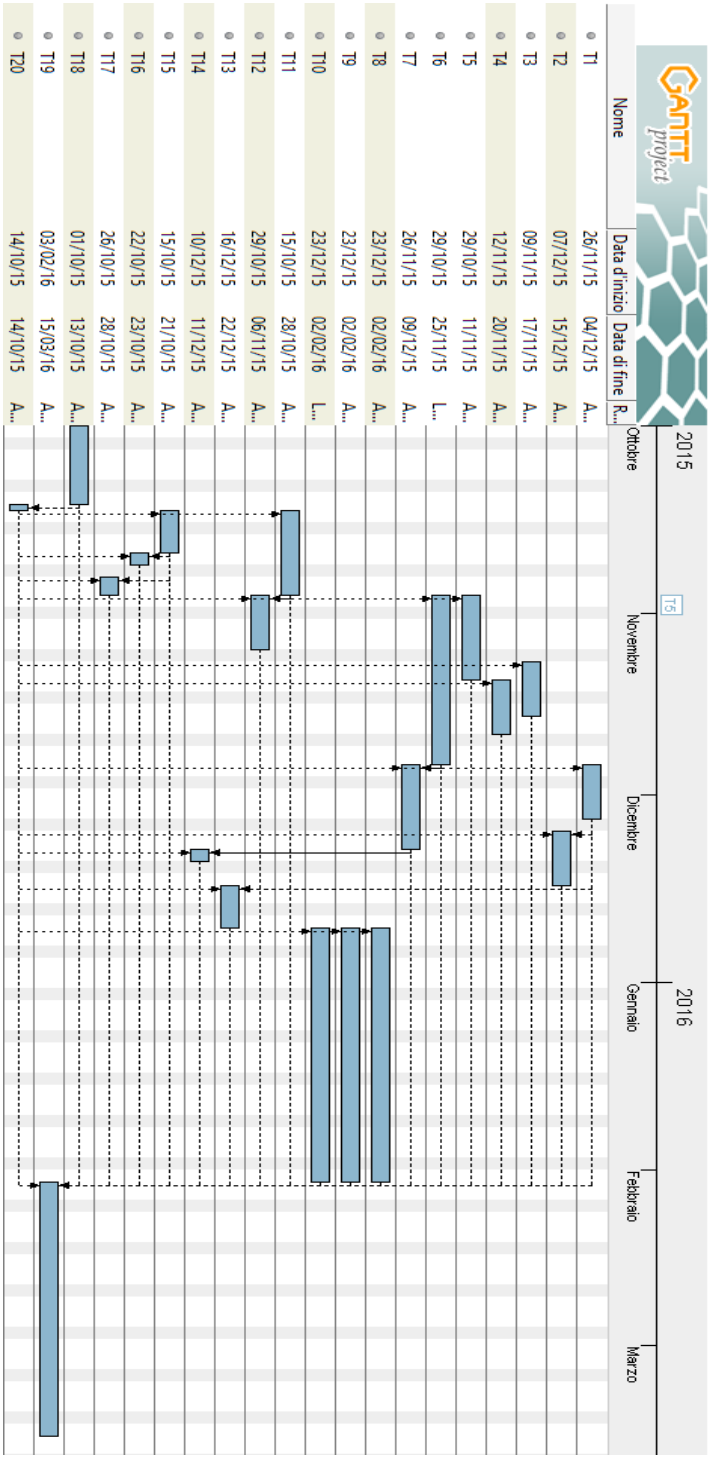
3.2 Tasks Scheduling

The total amount of days needed is 266, so approximately 89 days per person in the case of a normal employee. As sais above, in the case of a student this number is increased fourfold.

3.2.1 Normal Employee

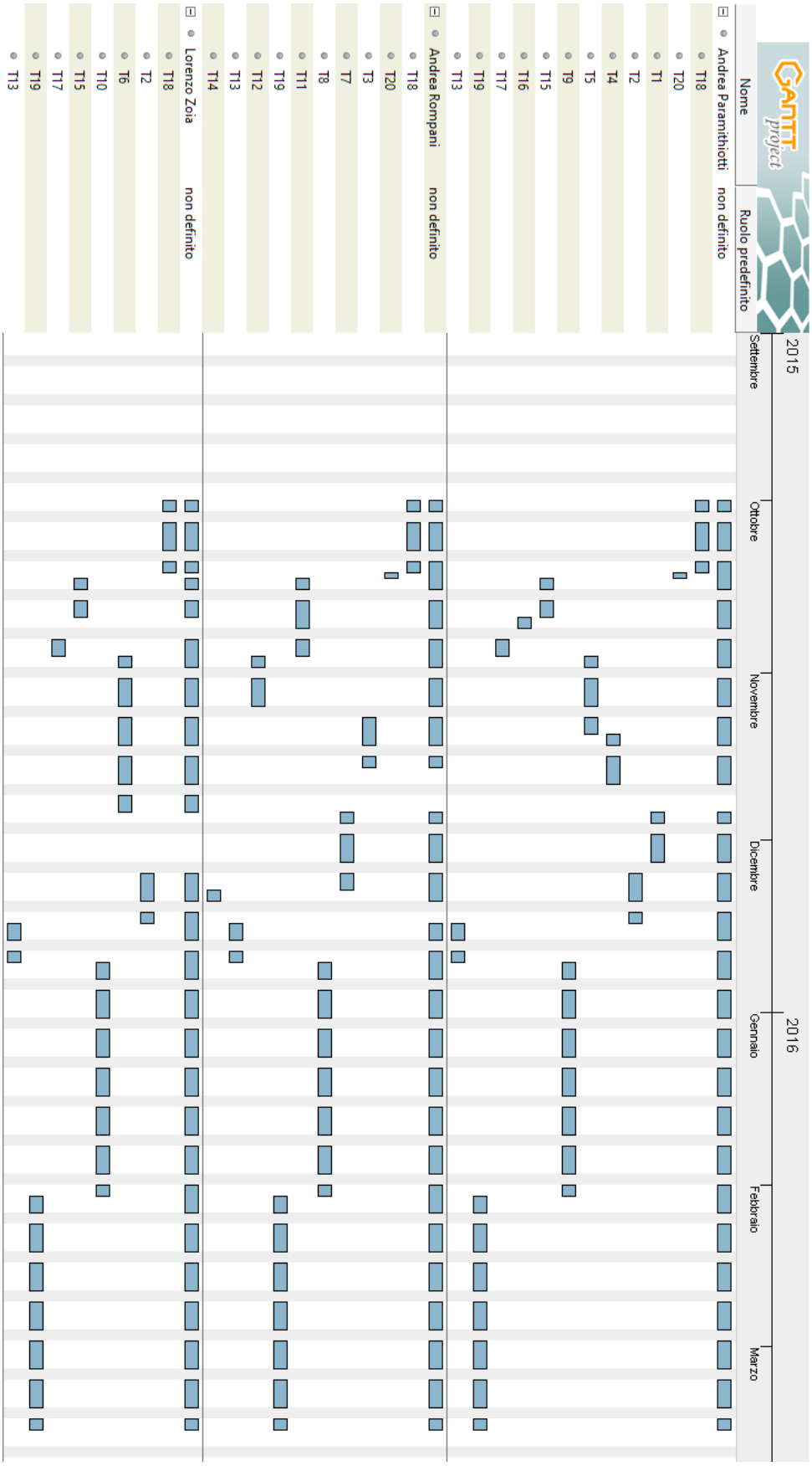
This is the scheduling if the team was composed of normal employers, working 8 hours per day.

3.2.1.1 Overall schedule



3.2.1.2 By Team Member

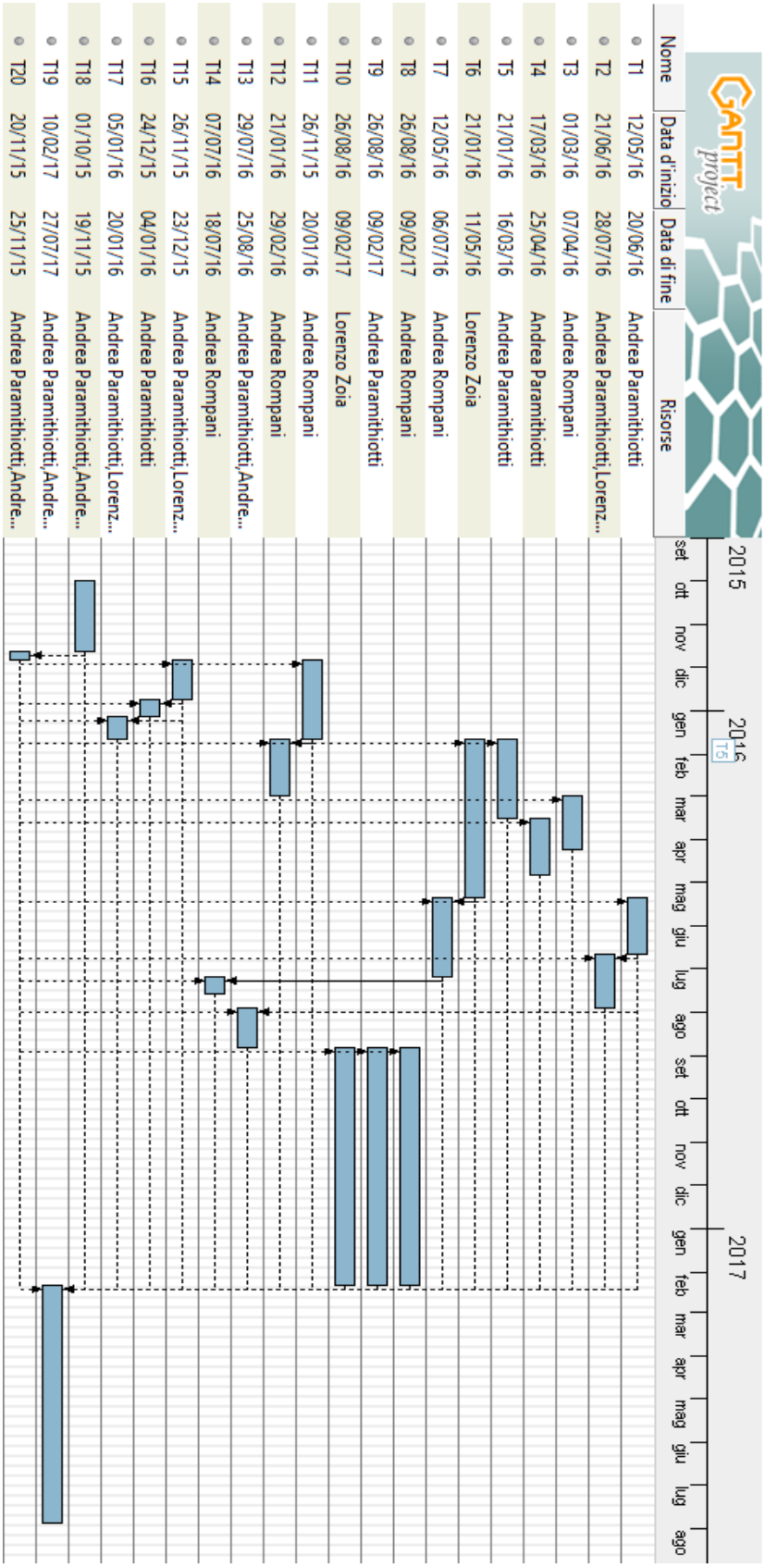
This is the schedule divided by team members



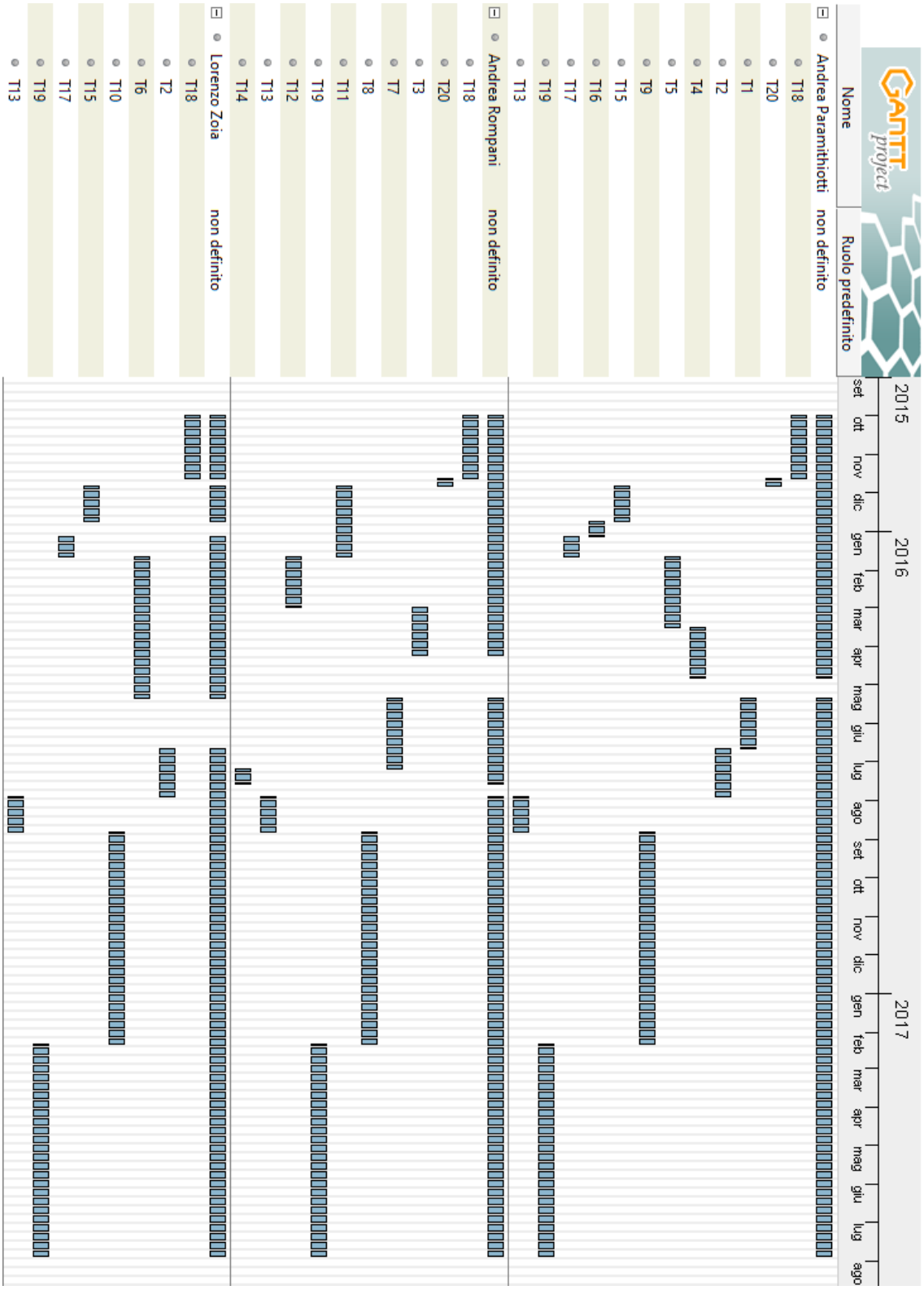
3.2.2 University Student Case

This is the case where the team is composed of university students, working at the project for a maximum of 2 hours per day

3.2.2.1 Overall schedule



3.2.2.2 By Team Member



3.3 Risks

A risk is a potential problem that might or might not happen during the development of the project.

3.3.1 Risks Definition

These are the risks that were pinpointed and that could occur in the development, with the related probability and the associated effects that they could have.

ID	Risk	Probability	Effects
1	Organizational financial problems force reductions in the project budget.	Low	Catastrophic
2	Unrealistic schedule and underestimation	Low	Serious
3	It is impossible to recruit staff with the skills required for the project.	Low	Catastrophic
4	Key staff are ill at critical times in the project.	Moderate	Serious
5	Faults in reusable software components have to be repaired before these components are reused.	Moderate	Serious
6	Changes to requirements that require major design rework are proposed	Low	Serious
7	The organization is restructured so that different management are responsible for the project	Low	Serious
8	The database used in the system cannot process as many transactions per second as expected	Low	Serious
9	Because of bottom up design, the integration testing takes more time than expected	Moderate	Serious
10	Network availability reduction due to ISP problems	Low	Catastrophic
11	Major changes in mobile operating systems programming specifications	Low	Moderate
12	Client requires changes in the user interface design	Low	Moderate
13	Unclear requirements	Low	Moderate
14	The server is unable to manage all the requests	Low	Negligible
15	The communication protocol between services becomes unusable or deprecated by industry standard	Very Low	Moderate

3.3.2 Risks Strategies

These are the strategies that should solve the problems caused by the potential risks.

Risk	Strategy and recovery actions
1	Underline to the management how the project is relevant and fundamental for the business, showing the pros of carrying on the project
2	Reorganize the team work in order to release a working demo for the client
3	Alert client for the recruitment difficulties, and in consequence of the possible delays
4	Reorganize the team in order to have the other team members working on the important missing parts
5	Replace the non-working components with reliable compatible and reliable versions
6	Warn the client of the possible raise in terms of cost and time
7	Prepare a document to express the importance of the project and another document to explain the key points of the project to the new management
8	Investigate the possibility of buying an higher performance database or investigate problems in queries performance
9	Explain the motivations to the client and the manager, underlining the importance of the integration testing on the reliability of the program
10	Contact ISP customer support and investigate the possibility of changing ISP
11	Consider the possibility to support the new operating system in a second moment, to avoid a delay in the schedule. Have the programmers keep up to date with the new features
12	Trace the changes impact on the schedule and communicate to the customer the possible delays caused by the changes
13	Ask the client for better requirements and try to avoid other misunderstandings
14	Due to the partitioning in services and the possibility to have multiple instances of the same service, a possible peak hour usage can be avoided by adding more instances of the most used services
15	Consider to have a protocol agnostic communication between services

4 Conclusions

The results of the analysis are definitely different because both of them were made without making a real development of the project.
Both the analysis, especially the COCOMO, are intrinsically dependent from the development of a project.
The functional points could be more precise, but it was decided to leave them this way in order to avoid to make up false evaluations.
The risk analysis and the assignment of the hours necessary to each task are also affected by the inexperience of the team members.