

POLITECNICO DI MILANO

2016-2017

Software Engineering 2: PowerEnJoy

Project Plan

Version 1.0

Peverelli Francesco

Reppucci Federico

**TABLE OF CONTENT**

|  |  |
| --- | --- |
| 1. INTRODUCTION  1.1. REVISION HISTORY  1.2. PURPOSE AND SCOPE  1.3. DEFINITIONS AND ACRONYSMS  1.4. REFERENCE DOCUMENTS  2. COST ESTIMATION  2.1. FUNCTION POINTS  2.1.1. FUNCTION POINTS METHODOLOGY  2.1.2. FUNCTION POINTS CALCULATION  2.2. COCOMO II  2.2.1. COCOMO II METODOLOGY  2.2.2. COCOMO II CALCULATION  3. PLANNING PROCESS  4. RESOURCE ALLOCATION  5. RISK MANAGEMENT  6. USED TOOLS  7. EFFORT SPENT |  |

**1. INTRODUCTION**

**1.1. REVISION HISTORY**

**1.2. PURPOSE AND SCOPE**

**1.3. DEFINITIONS AND ACRONYSMS**

ILF: Internal Logical File

EIF: External Interface File

EO: External Output

EI: External Input

EQ: External Inquiry

DET: Data Element Type

RET: Record Element Type

FTR: File Types Referenced

UFP: Unadjusted Function Point

PREC: Precedentedness

FLEX: Development Flexibility

RESL: Risk Resolution

TEAM: Team Cohesion

**1.4. REFERENCE DOCUMENTS**

<http://www.functionpointmodeler.com/fpm-infocenter/index.jsp>

**2. COST ESTIMATION**

**2.1. FUNCTION POINTS**

**2.1.1. FUNCTION POINTS METHODOLOGY**

The main purpose for this function points analysis is to estimate the size of the application, in order to come up with an accurate prediction of the time and resources required to complete the development process.

The function points methodology categorizes data at rest and data in motion in order to derive the lines of source code required to develop the corresponding functionalities, scaled based on a multiplicative factor depending on the programming language used.

**2.1.2. FUNCTION POINTS CALCULATION**

**INTERNAL LOGICAL FILES**

Any set of data originated and maintained by the application is considered an Internal Logical File (ILF).

ILFs are rated and scored, and the rating is based upon the number of data elements (DETs) and the record element types (RETs).

DET’s are atomic data visible by the user. Record element types have a more elusive definition, but in simple terms can be seen as the number of sub-records contained within the main record of each ILF.

The table below lists both the level (low, average or high) and appropriate score (7, 10 or 15).

In the following paragraph we list the ILFs identified within our application and their corresponding rating:

*User data: record storing information about the users*

DETs: 18

RETs: 3

Corresponding complexity: Low

*Reservation: record storing information about a user-made reservation*

DETs: 5

RETs: 2

Corresponding complexity: Low

*Profile stats: data derived from each user's rides*

DETs: 3

RETs: 1

Corresponding complexity: Low

*Payment: record of a user's payment*

DETs: 9

RETs: 3

Corresponding complexity: Low

*Ride: record containing information about a single ride*

DETs: 5

RETs: 3

Corresponding complexity: Low

*Safe area: a record storing the information about safe areas*

DETs: 3

RETs: 1

Corresponding complexity: Low

*Car state: record storing information about the sate of a car*

DETs: 9

RETs: 1

Corresponding complexity: Low

*Retrieval request: record storing a car retrieval request*

DETs: 5

RETs: 2

Corresponding complexity: Low

*Employee data: record storing information about an employee*

DETs: 6

RETs: 2

Corresponding complexity: Low

**EXTERNAL INTERFACE FILES**

External Interface Files (EIF) are files created and managed by another application, but used by the application for which the function point count is being carried out.

The rating is based upon the number of data elements (DETs) and the record types (RETs). The table below lists both the level (low, average or high) and appropriate score (5, 7 or 10)

*Maps provided by Google*

Considering that the maps are provided by an external service, an accurate analysis of DETs RETs proves to be rather impractical, but we can assume the worst-case amount of data to manage to be high.

Corresponding complexity: High

UFP = 1 \* 10 = 10

**EXTERNAL INPUT**

External Inputs (EI) are transactions (or elementary processes) that bring data from outside the application domain (or application boundary) to inside that application boundary.

EI’s are rated and scored according to the number of data element types (DET’s) and the file types referenced (FTR’s).

File Types Referenced(FTR) are file types referenced by a transaction. An FTR must also be an internal logical file or external interface file.

The table below lists both the level (low, average or high) and appropriate score (3, 4 or 6).

*Registration (with data insertion)*

DETs: 18

FTRs: 1

Corresponding complexity: average

*Car reservation*

DETs: 5

FTRs: 3

Corresponding complexity: high

*Modify profile data*

DETs: 17

FTRs: 1

Corresponding complexity: average

NOTE: the function point count takes into consideration the fact that three different clients will be developed through three different programming languages. As a result of this, each input, output or inquiry involving both the client side and the server of the application will be counted in total four times, each time with the appropriate coefficient depending on the selected programming language

UFP = 2 \* 4 + 6 = 14 (considered four times)

**EXTERNAL OUTPUT**

External Outputs (EO) are transactions (or elementary processes) that take data from a resting position to outside the application domain and present information to a user through processing logic other than, or in addition to, the retrieval of data or control information. The processing logic must contain at least one mathematical formula or calculation, create derived data, maintain one or more ILFs or alter the behavior of the system.

EOs are rated and scored according to the number of data elements (DETs) and the file types referenced (FTRs). The table below lists both the level (low, average or high) and appropriate score (4, 5 or 7).

*Retrieval notification*

DETs: 2

FTRs: 1

Corresponding complexity: low

*Display personal data*

DETs: 19

FTRs: 1

Corresponding complexity: low

*Display payment history*

DETs: 9

FTRs: 1

Corresponding complexity: low

UFP = 2 \* 4 (considered four times) + 4 = 8 (considered four times ) + 4

**EXTERNAL INQUIRY**

External Inquiry (EQ) is an elementary process with both input and output components that results in data retrieval from one or more internal logical files and external interface files. The input process does not update or maintain any FTRs (Internal Logical Files or External Interface Files) and the output side does not contain derived data.

The rating is based upon the total number of unique (combined unique input and out sides) data elements (DETs) and the file types referenced (FTRs). An EQ is rated (Low, Average or High) like an EO, but assigned a value like and EI.

*Payment data validation: validation of the credit card data provided by the user*

DETs: 2

FTRs: 1

Corresponding complexity: low

*Password retrieval*

DETs: 2

FTRs: 1

Corresponding complexity: low

*Payment request*

DETs: 3

FTRs: 1

Corresponding complexity: low

*Browse website (request of the static page)*

DETs: 1

FTRs: 1

Corresponding complexity: low

*Login*

DETs: 2

FTRs: 1

Corresponding complexity: low

*Car positions inquiry*

DETs: 4

FTRs: 1

Corresponding complexity: low

*Map inquiry (to the external system)*

DETs: 2

FTRs: NA

Corresponding complexity: low

UFP = 6 \* 3 (considered four times) + 3 = 18 (considered four times) + 3

*TOTAL COUNT* = 80 + 40 (considered four times)

*SLOC count*: 46 \* 80 + 40 \* (53(Java) + 27(Objective C) + 54(C#) + 46(J2EE)) = 10.880 SLOC

**2.2. COCOMO II**

**2.2.1. COCOMO II METODOLOGY**

**2.2.2. COCOMO II CALCULATION**

**SCALE FACTOR**

Precedentedness (PREC)

[VERY LOW - 0,05]

This is the first project for the development team with this characteristics and size.

Development Flexibility (FLEX)

[NOMINAL - 0,03]

The only pre-established requirements are the pre-existing interfaces of the cars

Architecture / Risk Resolution (RESL)

[Nominal - 4,24]

Team Cohesion (TEAM)

[VERY HIGH - 0,01]

The development team is composed by 2 student attending a MSc Computer Engineering. The communication is agile and efficient since they are colleagues that work together every day, and are on good terms with each other. Furthermore, they have worked together in previous projects.

Process Maturity (PMAT)

[VERY LOW - 0,05]

The development process is unstructured and unsupervised. The division of labor is weekly established and the stakeholders are not involved in our decisions.

**POST-ARCHITECTURE EFFORT ESTIMATION**

***Product factors***

RequiredSoftwareReliability (RELY)

[NOMINAL - 1,00]

A system failure can at most cause the loss of a small amount of money. For instance, during the failure, a user could not be able to reserve a car or the payment procedure could not be well performed.

DataBaseSize(DATA)

[NOMINAL - 1,00]

The DATA cost driver is obtained by calculating the ratio of bytes in the testing database to SLOC in the program.

In the testing database, we assume that each field, being always raw data (string, dates, identifiers, numbers, Booleans), can take up at most 20byte. This is a pessimistic upper bound, useful for a quick estimate of the testing database size.

We consider to have:

- 100 users: each user, considering all its fields, takes up at most 20byte \* 18 = 360byte

- 10 ride/user: each ride, considering all its fields, takes up at most 20byte \* 5 = 100byte

- 10 reservation/user: each reservation, considering all its fields, takes up at most 20byte \* 5 = 100byte

- 20 cars: each car, considering all its fields, takes up at most 20byte \* 9 = 180byte

- 20 retrieval requests: each retrieval requests, considering all its fields, takes up at most 20byte \* 5 = 100byte

- 20 employee: each employee, considering all its fields, takes up at most 20byte \* 6 = 120byte

- 1000 payments: each payment, considering all its fields, takes up at most 20byte \* 9 = 180byte

- 100 profile stats: each profile stats, considering all its fields, takes up at most 20byte \* 3 = 60byte

The safe area is mainly a collection of points. Each point has 2 coordinates. Each coordinate we can suppose takes up 8byte.

We consider to have 100 points the describe the edges of polygons: the total occupied size is 100 \* 8 \* 2 byte = 16000byte

The total size is: 446KB

From the functional point, the SLOC is: 10880 (see Functional Point Count capture)

Hence, the ratio DBbyte/SLOC is: 446000/10880 = 41

ProductComplexity(CPLX)

[average: 1,05]

The analysis is done on the entire system, considering always the worst case.

\* Control Operations [HIGH - 1,17]: the system uses queues and complex data structures, in order to handle users, cars, reservation request.

\* Computational Operations [LOW - 0,87]: the system evaluates very simple computational operations.

Examples:

1) user profile statistics (additions)

2) find the nearest car within a range (distance from 2 points formula)

3) bonus calculation (percentages calculation)

4) payment calculation (addition, multiplication)

\* Device-dependent Operations [NOMINAL 1,00]: I/O processing includes device selection, status checking, and error processing.

\* Data Management Operations [HIGH 1,17]: the system stores data in a structural way through a DBMS. Therefore, the system seeks, reads and writes the data. Data can be also filtered, analyzed, manipulated.

\* User Interface Management Operations [LOW 0,87]: the graphic user interface is composed of simple input forms, buttons, radio buttons, and text. The most complex form is the one that displays the map. In this case, however, the majority of work is performed by Google's API.

DevelopedforReusability(RUSE)

[NOMINAL - 1,00]

This project has been designed following general good practices of reusability of the components. Each component has been thought to be self-contained with a specific purpose.

DocumentationMatchtoLife-CycleNeeds(DOCU)

[NOMINAL - 1,00]

The project is well documented. A set of starting documents (RASD, DD, ITPD) analyze each aspect of the project, from the requirements to the software components. Also, the code is documented, making it easy to ready and quick to modify.

**Platform factors**

ExecutionTimeConstraint(TIME)

[NOMINAL - 1,00]

The system processes mainly request-response actions, without CPU bound processes.

MainStorageConstraint(STOR)

[NOMINAL - 1,00]

This constraint is essentially irrelevant, since there are not contraint on the hardware.

PlatformVolatility(PVOL)

[LOW - 0,87]

The platform of the system is composed of: operating systems, application server, framework (e.g. J2EE), IDEs, the cars' electronics, smartphones' hardware. Each of them is receive a major update (both hardware and software) at most 1 time for year.

**Personnel Factors**

Analyst Capability (ACAP)

[NOMINAL - 1,00]

The analysts of this project are MSc graduated students without experience. Their decisions are driven by guidelines, good practices, and personal knowledge.

Programmer Capability (PCAP)

[NOMINAL - 1,00]

The programmers are the same people that have designed the project. They know very well the principles of objective oriented programming languages and the main patterns.

Personnel Continuity (PCON)

[VERY HIGH - 0,81]

The project has been followed by the team from the beginning to the end.

Applications Experience (APEX)

[LOW - 1,10]

The majority of the software and technologies that the developers is supposed to deal with is almost unknown.

Platform Experience (PLEX)

[LOW - 1,09]

The majority of the software and hardware that the developers is supposed to deal with is almost unknown.

Language and Tool Experience (LTEX)

[LOW - 1,09]

This level is an average value considering a good knowledge of Java programming language, moderate of J2EE, and poor of C# and Objective-C.

**Project Factors**

Use of Software Tools (TOOL)

[VERY LOW - 1,17]

CASE tools were not used during the development of the project.

Multisite Development (SITE)

[Extra High - 0,80]

The team is very small (2 people) and have worked all together, in the same room. They also use instant message clients to talk.

**General Factor**

Required Development Schedule (SCED)

[NOMINAL - 1,00]

The schedule of the project has been completely respected, without rushing before the milestones.

**DATA COLLECTION**

*SCALE FACTOR*

|  |  |
| --- | --- |
| PREC | 0,05 |
| FLEX | 0,03 |
| RESL | 4,24 |
| TEAM | 0,01 |
| PMA | 0,05 |

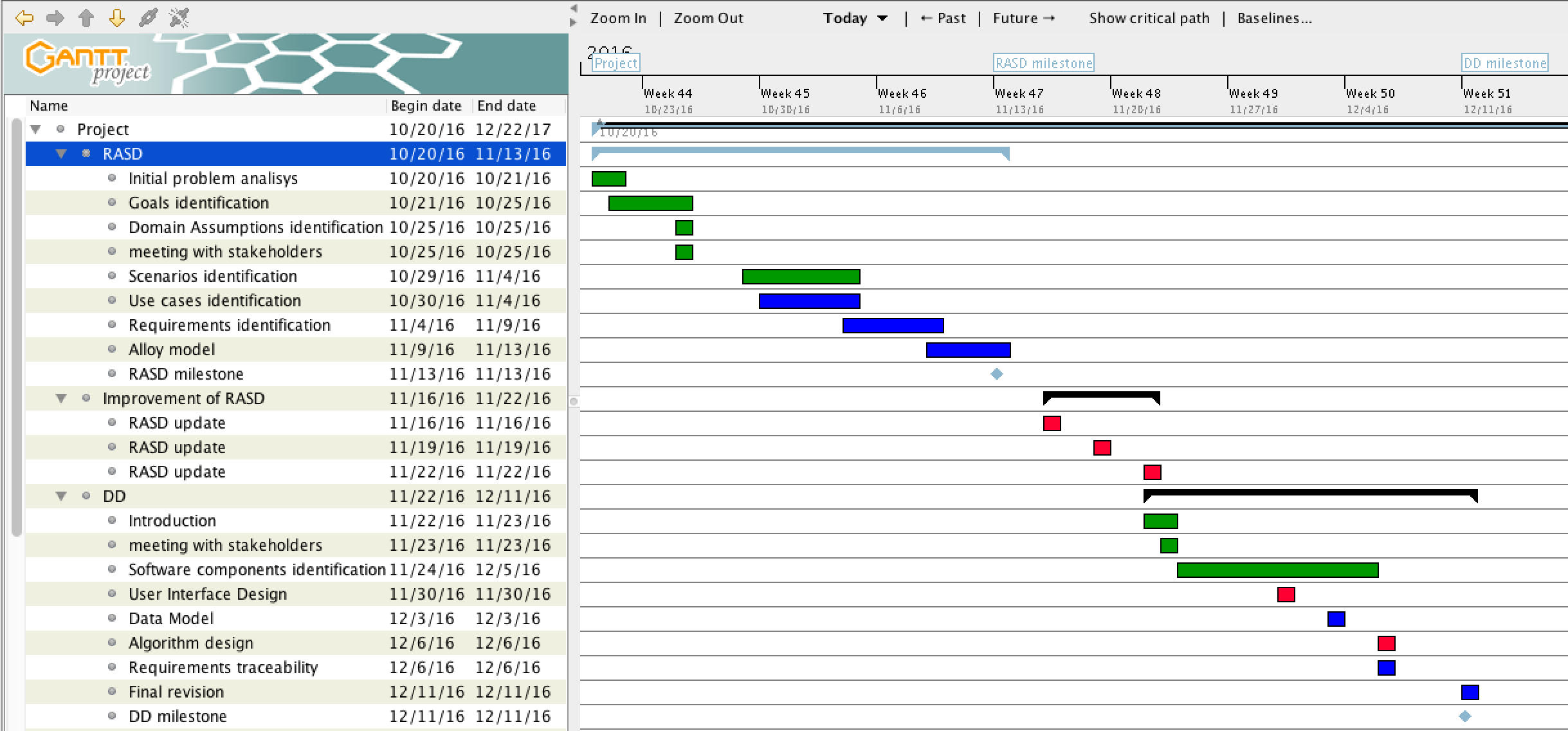
*POST-ARCHITECTURE EFFORT ESTIMATION*

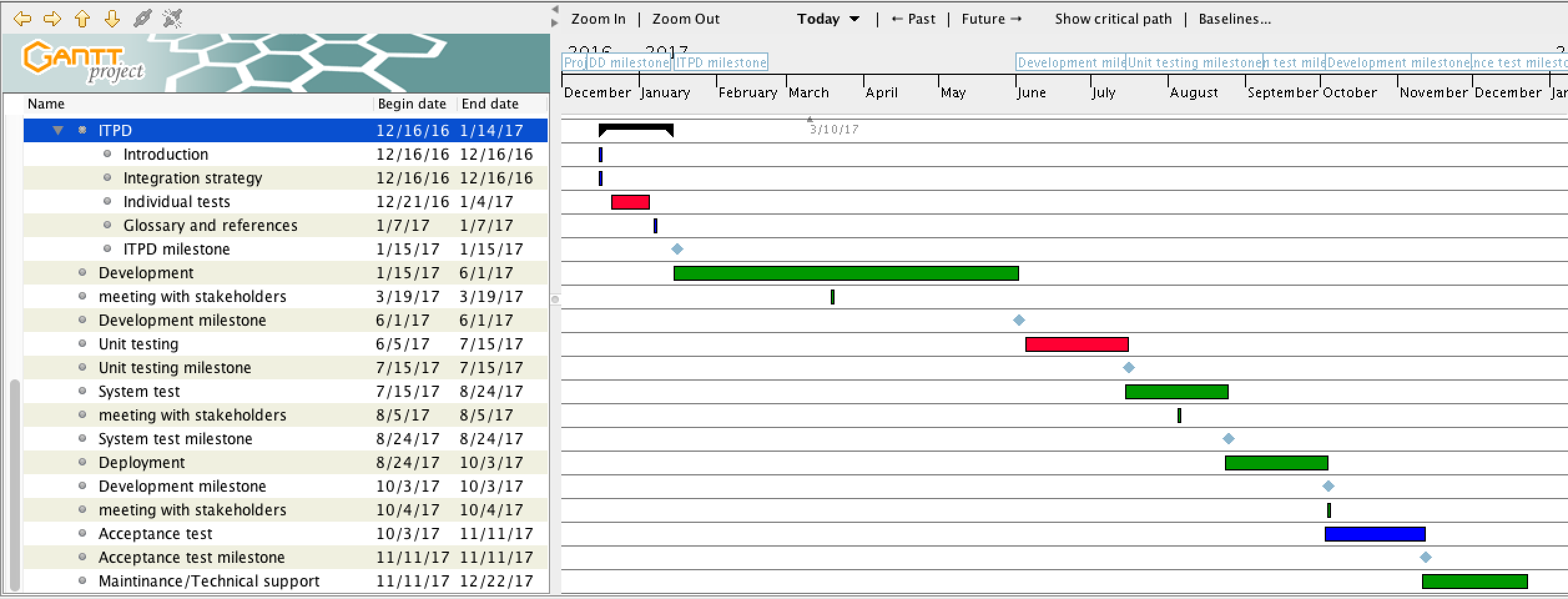
|  |  |
| --- | --- |
| RELY | 1,00 |
| DATA | 1,00 |
| CPLX | 1,05 |
| RUSE | 1,00 |
| DOCU | 1,00 |
| TIME | 1,00 |
| STOR | 1,00 |
| PVOL | 0,87 |
| ACAPO | 1,00 |
| PCAP | 1,00 |
| PCON | 0,81 |
| APEX | 1,10 |
| PLEX | 1,09 |
| LTEX | 1,09 |
| TOOL | 1,17 |
| SITE | 0,80 |
| SCED | 1,00 |

|  |  |
| --- | --- |
| A | 2,94 |
| E | 0,9538 |
| B | 0,91 |
| KSLOC | 10,88 |
| PM | 26 |
| PEOPLE | 2 |
| MONTHS | 13 |

|  |  |  |
| --- | --- | --- |
| **Phase** | **Percentage** | **Months** |
| Analysis and Design | 23 % | 3 |
| Development | 32 % | 4,16 (+1 extra) |
| Unit testing | 10 % | 1,3 |
| System testing | 10 % | 1,3 |
| Deployment | 10 % | 1,3 |
| Acceptance test | 10 % | 1,3 |
| Maintenance/Technical support | 10 % | 1,3 |

**3. PLANNING PROCESS AND RESOURCE ALLOCATION**





**5. RISK MANAGEMENT**

**6. USED TOOLS**

**7. EFFORT SPENT**

**EFFORT SPENT**

Together: [0h]

Reppucci: [7h]

Peverelli: [6h]