POLITECNICO
MILANO 1863

Software Engineering 2: "PowerEnJoy"

Project Plan

Version 1.0

Piccirillo Luca - 790380
Zampogna Gian Luca - 863097
Zini Edoardo - 875275

January 22th, 2017

Contents

1 Introduction

1.1 Purpose and Scope

This document analyze the size, cost and effort of realizing the PowerEnJoy project. It is done relying on Function Points and COCOMO II methodologies. Following the result of these operations, tasks needed to complete the project are identified and grouped in order to obtain a schedule for the project that can be visualized using Gantt chart. At the end of the document, a risks list including appropriate mitigation strategies can be found.

1.2 Acronyms

- FP = Function Points
- GLONASS = GLObal NAvigation Satellite System
- GPS = Global Positioning System
- SPA = Special Parking Area
- RASD = Requirements Analysis and Specification Document
- UFP = Unadjusted Function Points

1.3 Definitions

- **Car:** every vehicle, which respects the requirements, that the system allows the users to use.
- **Galileo:** GPS-like system managed by Europe.
- **Safe (Parking) Area:** pre-defined areas (i.e. streets) where a user is allowed to park.
- **Special Parking Area:** pre-defined areas (i.e. streets) where a user is allowed to park and where the batteries of the Cars can be plugged into the power grid.
- **Base Fare:** amount of money that the user pays for the ride duration only. It does not include any discount or penalty for any other specific condition.
- **Ride Fare:** amount of money that the user pays. It includes any discount or penalty for any other specific condition satisfied during the ride.

1.4 Reference Documents

- Project's Assignment document: AA 2016-2017 Software Engineering 2 - Project goal, schedule, and rules.
- Software Engineering 2: "PowerEnJoy" Requirements Analysis and Specification Document.
- Software Engineering: Principles and Practice, Hans van Vliet, 2007 (source)

2 Size, Cost and Effort Estimation

2.1 Size Estimation: Function Points

UFP Complexity Weights			
Function type	**Low**	**Average**	**High**
Internal Logic Files	7	10	15
External Interface Files	5	7	10
External Inputs	3	4	6
External Output	4	5	7
External Inquiries	3	4	6

2.1.1 Internal Logical File

- **Safe Areas Data**
  It consist of a set of Coordinates (*Longitude*, *Latitude*), so we consider it with a low weight. **7 FPs**

- **SPA data**
  As for the Safe Area this is a list of Coordinates, it can be considered with a low weight. **7 FPs**

- **User Data**
  It has different fields (Name, Surname, E-mail, Password, Credit Card, Driving Licence) but some of those require encryption and special check like Password, Credit Card and Driving Licence, because of that we consider it with a average weight. **10 FPs**

- **Car Data**
  As for the user, it is one entity and contains car attributes, so we consider it with a low weight. **7 FPs**

- **Discount Data**
  It's an entity where all discounts are stored and so is pretty simple, we consider it with a low weight. **7 FPs**

- **Booking Data**
  It has a simple structure: contains (bookingTime, CarID, UserID), so it will be considered with a low weight. **7 FPs**

- **Ride Data**
  It has a simple structure: contains (rideStart, rideEnd, CarID, UserID), so it will be considered with a low weight. **7 FPs**

2.1.2 External Interface File

- **Localization Service**
  Contains the Coordinates of the vehicle, since it has a simple structure we consider it with a low weight. **5 FPs**
- **Map Retrieval**
  Contains a complex structure acquired from the third party map provider, so we assigned an high weight. **10 FPs**
- **Payment Data**
  Contains the Payment Data, since this information needs to be safely stored we consider it with an average weight. **7 FPs**

2.1.3 External Inputs

- **Account Creation**
  It involves only the Account Manager but some data needs to be carefully checked, we assign an average weight. **4 FPs**
- **Login/Logout**
  They are simple operation: involves only the Account Manager. 2x3=**6 FPs**
- **Profile Management**
  As for the Account Creation we may need to do some check, so we assign an average weight. **4 FPs**
- **Car Booking**
  It involves several entities, because of that we assign an high weight. **6 FPs**
- **Reservation Deletion**
  It's a quite simple operation: set the booked car as free and delete the booking instance, so we consider it with a low weight. **3 FPs**
- **Car Unlocking**
  It involves several entities and some conditions must be checked, we assign an high weight. **6 FPs**

2.1.4 External Outputs

- **Ride and Base Fare**
  The Base Fare value must be shown in real time, Ride Fare is calculated at the end of the ride from the Base Fare value, because of that we assign to them a low weight. 2x4=**8 FPs**
- **Successful Book Notification**
  It's a simple notification sent to the user, we consider it with a low weight. **4 FPs**

- **Penalty Notification**
  This notify the user a bill him for the penalty amount, since it involves also the payment service we assign an average weight. **5 FPs**

2.1.5 External Inquiries

- **Retrieve Booked Car**
  It's a quite simple query, we only need to look for the right Car ID, so we consider it with a low weight. **3 FPs**

- **SPA List**
  Retrieve a list of closest SPAs, it's quite simple we assign a low weight. **3 FPs**

- **Closest SPA**
  Retrieve the closest SPA, to a given position, that satisfy our criteria, this is a simple function and for that we assign a low weight. **3 FPs**

- **Discount Available**
  It retrieve the discount available and check if they are applicable, so we assign an average weight. **4 FPs**

- **Map with Cars Position**
  This needs to retrieve both map and Cars position, because of that we consider it with an average weight. **4 FPs**

2.1.6 Summary

Function Type	**Total FPs**
Internal Logic Files	7+7+10+7+7+7+7=**52**
External Interface Files	5+10+7=**22**
External Inputs	4+6+4+6+3+6=**29**
External Output	8+4+5=**17**
External Inquiries	3+3+3+4+4=**17**
Total UFPs	**137**

2.2 Cost and Effort Estimation: COCOMO II

2.2.1 Scale Drivers

- **Precedentedness**
  The product belongs to a category which is not newly to our company. We expect to reuse most techniques and methodologies from previous experiences or well known patterns. However some features regarding the automotive IoT will certainly result in some challenges to be overcome. Nominal(3.72)

- **Development Flexibility**
  The customer imposed some requirements over high-level end-user experience. So we have enough freedom in designing inner components but the expected distributed characterization of the solution will certainly impose lots of constraints regarding internal subsystems interactions as well as external components interfacing. Low(4.05)

- **Architecture / Risk Resolution**
  To prevent major compatibility problems across various parts of the solution we intend to give strict guidelines regarding its architecture perspective. Such a choice will give relevant robustness to the solution design against unexpected issues affecting development phase. High(2.83)

- **Team Cohesion**
  Most of members of the team have previous experience working together at the same product. We wont expect the addition of new members will disrupt inner interpersonal relationships. Moreover each member has reasonably no reasons to leave the team. Very High(1.10)

- **Process Maturity**
  As a software production company working with cutting-edge technologies we're continuously in search of new approaches to address new design scenarios, therefore major production phases are subject to processes for which no maturity rating is available. Low(6.24)

2.2.2 Cost Drivers

- Product
  - **Required Software Reliability**
    As stated in RASD, the customer has not expressed any reliability need so we expect to implement basics reliability oriented patterns in order to invest less effort as possible on this point even keeping reliability at a more than sufficient level. Low(0.88)
  - **Database Size**
    The solution will not require more than a middle-low amount of data to be managed as long as the customer does not intend to keep temporary data for other reasons like mining or selling. In any case the amount of real-time managed data does not grow over some predictable levels. Nominal(1.00)
  - **Product Complexity**
    The solution will manage a large amount of devices that need to be concurrently connected and synchronized by many different protocols. We expect to decompose the product into many different components to help managing that level of complexity. High(1.15)

  - **Developed for Reusability**
    The abstraction levels of many solution elements will be high to overcome the front-end volatility. Moreover, the application scenario at this historical moment is quite common so we expect much of our work may come handy in future if we plan to reuse some elements. Very High(1.29)
  - **Documentation Match to Life-Cycle Needs**
    We provide deeply detailed documentation for each step of solution life-cycle. The suitability of those documents should give a valuable speed-up in terms of effort. High(1.06)

- Personnel
  - **Analyst Capability**
    Analysts involved in this project have proven experience into providing very good solutions to various kind of problems this product could arise. High(0.83)
  - **Programmer Capability**
    We expect most of the team will be able to work without major problems on the coding of required features. Nominal(1.00)
  - **Personnel Continuity**
    We have very good reasons to grant for sure that each member of the team will remain part of this project until the end of the academic course this project relates to. High(0.92)
  - **Applications Experience**
    The whole team has an average experience in working onto this specific kind of applications. To give a more realistic and tolerant effort prediction we prefer to keep its rating at a low level. Low(1.10)
  - **Platform Experience**
    Given that the deployment platform of the solution is still undefined we prefer to keep this rating at a low level to better adapt to new platforms eventually which would eventually serve better the architecture choices. Low(1.10)
  - **Language and Toolset Experience**
    Most of the team members have a very good familiarity with any kind of software development framework and its related environments. We expect am high level of knowledge sharing across team members. High(0.88)

- Platform
  - **Time Constraint**
    No kind of requirements from the customer. However we prefer to keep the solution as usable as other similar competitors solutions. Nominal(1.00)
  - **Storage Constraint**
    No kind of requirements from the customer. Since no further decisions will be taken on this point, we keep its effort rating to a standard level. Nominal(1.00)

  - **Platform Volatility**
    We expect to base our solution on cutting-edge technologies which are subject to a well known rating of volatility. Also some subsystems will probably need to be reworked in future, especially in the set of user-premise elements subject to frequent upgrades. Very High(1.30)
- Project
  - **Use of Software Tools**
    The use of powerful enterprise frameworks will give a valuable advantage in terms of the effort taken to configure and setup both development and production environments High(0.86)
  - **Multi-site Development**
    The whole team is collocated in the same site. Any interaction with the customer will not take any effort in time since the customer location is the same of our company. Very High(0.84)
  - **Required Development Schedule**
    Since we had no release timing requirement from the customer we keep a standard rating for this point. Nominal(1.00)

2.2.3 Expected effort summary

Given the effort (ϵ) equation, which is:

$$\epsilon = A \times Size^{E} \times EAF$$

Where:

- A = 2.94 as in COCOMO II.2000
- Size: actual size of the product to be developed in terms of KSLOC (thousands of Source Lines Of Code).
  $$Size = \frac{UFP \times LF}{1000}$$
  Where LF = 46 SLOCs as for the avarage value of J2EE applications.
- E : Exponent derived from the five Scale Drivers, with
  $$E = B + 0.01 \times \sum_{j=1}^{5} SF_j$$
  With B = 0.91 as in COCOMO II.2000
- EAF : Effort Adjustment Factor derived from the Cost Drivers, with
  $$EAF = \prod_{i=1}^{N} EM_i$$

Therefore the estimated effort is:

$$\epsilon = 2.94 \times (\frac{137 \times 46}{1000})^{0.91+0.01\times 17.94} \times 1.15 = \textbf{29.4 person-month}$$

with an estimated duration of:

$$\delta = C \times \epsilon^{D+0.2\times(E-B)} = \textbf{11.2 months}$$

Note: C = 3.67 and D = 0.28 as for COCOMO II.2000

And a (minimum) required number of people of:

$$\pi = \lceil \epsilon/\delta \rceil = \textbf{4 persons}$$

3 Task and Schedule and Resource Allocation

This section contains the list of all tasks, their durations and their dependencies.

Please note that in dependencies column *Ta-b* means all the tasks from *Ta* to *Tb*.

3.0.1 Requirements Analysis and Specification

Task	Task Description	Duration (days)	Dependencies
T1	Scenarios identification	3	
T2	Assumptions and goals	5	T1
T3	Quality of service	2	T1
T4	Constraints	2	T1
T5	Functional requirements	4	T2
T6	External interface requirements	5	T5
T7	Use cases	5	T5
T8	Alloy modeling	6	T5
T9	Class diagram	2	T5
T10	Document introduction	2	T3, T4, T6-9
T11	Final document review	2	T10
M1	**Milestone: Requirements Analysis and Specification Document**		

3.0.2 Design and Architecture

Task	Task Description	Duration (days)	Dependencies
T12	High level components and architectural styles	10	M1
T13	Component view and component interfaces	12	T12
T14	Deployment view	6	T12
T15	Runtime view	8	T13
T16	Algorithmic design	6	T15
T17	User interface design	6	T15
T18	Requirements Traceability	4	T14, T16, T17
T19	Document introduction	2	T18
T20	Final document review	2	T19
M2	**Milestone: Design Document**		

3.0.3 Integration Test Plan

Task	Task Description	Duration (days)	Dependencies
T21	Integration strategy	6	M2
T22	Individual steps and test description	8	T21
T23	Tools and test equipment required	3	T22
T24	Program stubs and test data required	3	T23
T25	Document introduction	2	T24
T26	Final document review	2	T25
M3	**Milestone: Integration Test Plan Document**		

3.0.4 Implementation

Task	Task Description	Duration (days)	Dependencies
T27	Components development	90	M3
T28	Components testing	40	T27
T29	Components integration in subsystems and testing	30	T28
T30	Subsystems integration in system and testing	30	T29
T31	Full system on-field testing	30	T30
M4	**Milestone: First Release**		

3.1 Scheduling

This section contains Gantt charts of the entire project. Due to the large dimensions of the chart, it has been split to improve readability.

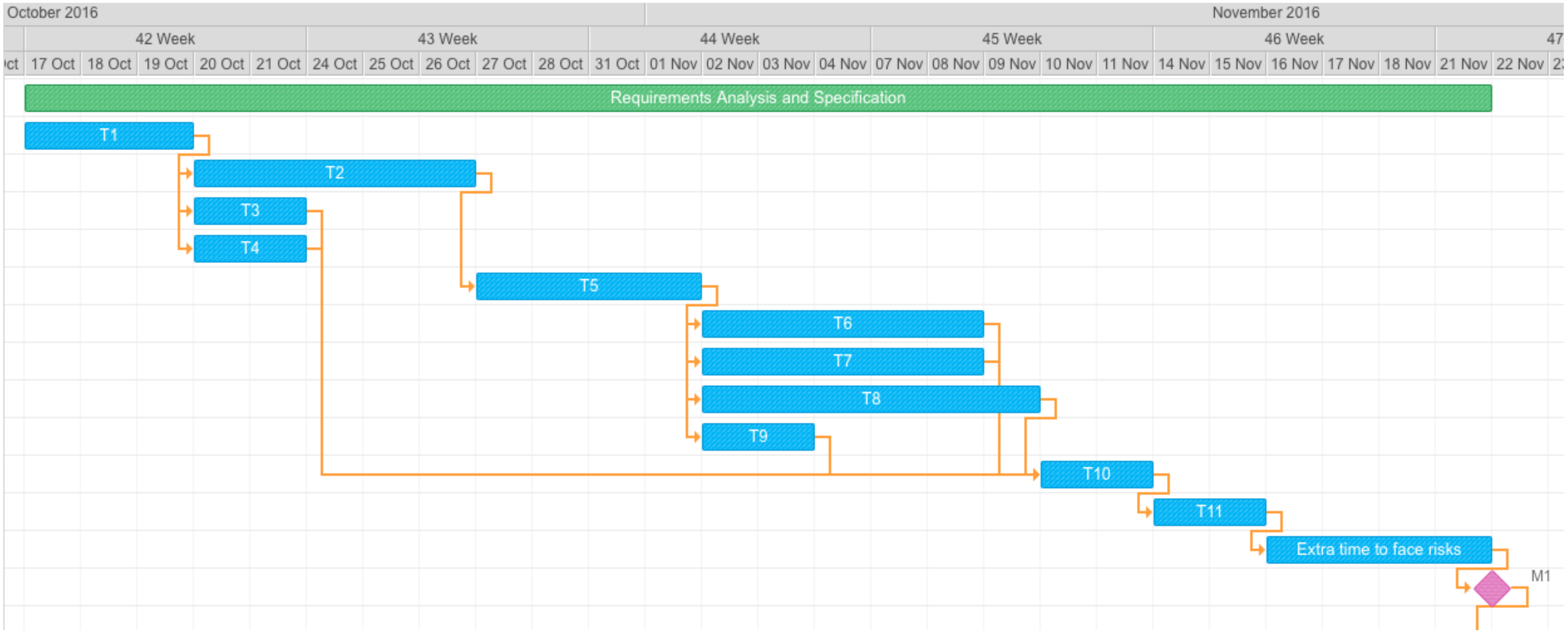

Figure 1: Gantt chart of Requirements Analysis and Specification

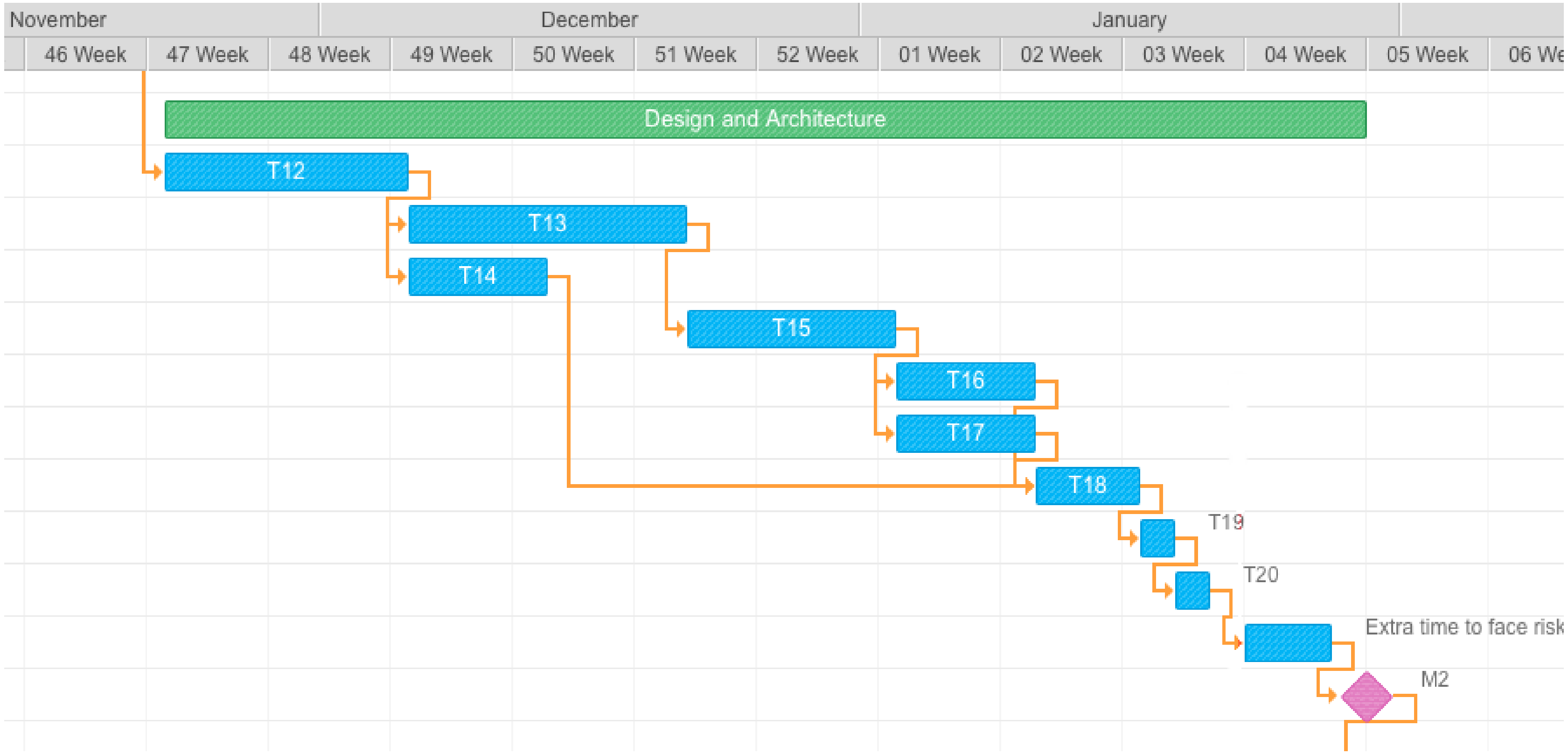

Figure 2: Gantt chart of Design and Architecture

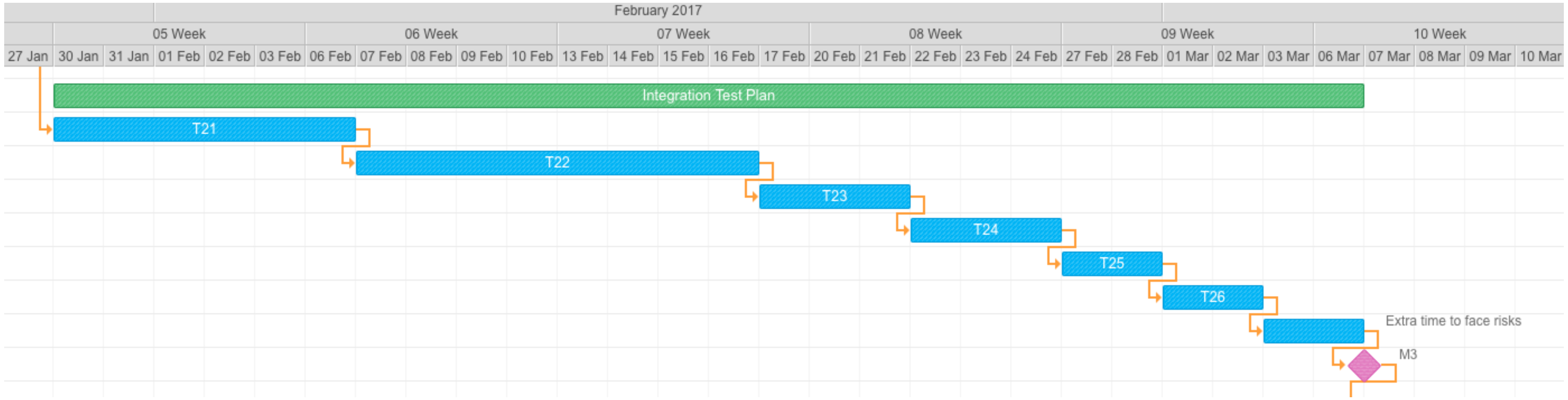

Figure 3: Gantt chart of Integration Test Plan

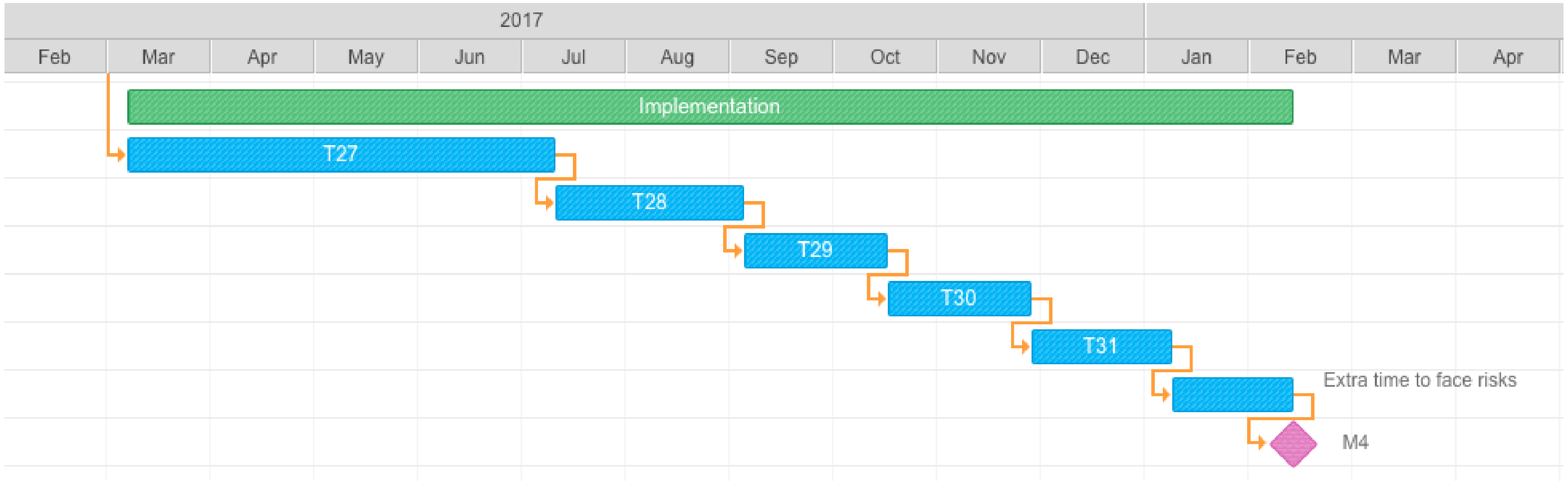

Figure 4: Gantt chart of Implementation

3.2 Resource Allocation

This section contains a table for each phase of the project where each of the previous tasks is assigned to at least one member of the group.

Member	**T1**	**T2**	**T3**	**T4**	**T5**	**T6**	**T7**	**T8**	**T9**	**T10**	**T11**
Piccirillo Luca		X	X	X	X		X		X	X	X
Zampogna Gian Luca	X	X	X	X				X		X	X
Zini Edoardo	X	X	X	X		X	X			X	X

Member	**T12**	**T13**	**T14**	**T15**	**T16**	**T17**	**T18**	**T19**	**T20**
Piccirillo Luca	X	X	X					X	X
Zampogna Gian Luca	X	X		X			X	X	X
Zini Edoardo	X	X			X	X		X	X

Member	**T21**	**T22**	**T23**	**T24**	**T25**	**T26**
Piccirillo Luca	X	X	X	X	X	X
Zampogna Gian Luca	X	X	X		X	X
Zini Edoardo	X	X			X	X

Member	**T27**	**T28**	**T29**	**T30**	**T31**
Piccirillo Luca	X	X	X	X	X
Zampogna Gian Luca	X	X	X	X	X
Zini Edoardo	X	X	X	X	X

4 Risk Management

This section contains the contingency plans for the identified risks. To help understating the classification, we add a brief explanation of probability coefficients:

- **Low Probability**: risk is unlikely to arise.
- **Moderate Probability**: risk should not arise, but it is possible it will.
- **High Probability**: risk is likely to arise.

and effect coefficients:

- **Moderate Effect**: impact on project schedule should be limited.
- **Serious Effect**: large impact on the project that may end up in a small delay.
- **Critical Effect**: project delayed to face the new problems.
- **Catastrophic Effect**: the entire project may be cancelled.

Risk	Risk Description	Probability	Effect
R1	Group member get ill and cannot work on the project for a short period of time.	Moderate	Moderate
R2	Requirements are modified by stakeholders or contains errors.	Moderate	Critical
R3	Milestones cannot be completed within the deadline.	Moderate	Critical
R4	Architecture cannot scale well enough to handle an unexpected load.	Moderate	Critical
R5	Failures in external components of the system.	Low	From Critical to Catastrophic
R6	Damage on system platform hardware.	Low	Catastrophic
R7	Bug on application.	Moderate	Moderate
R8	New competitors enters the market.	Moderate	Serious
R9	New laws impose unavoidable changing in the requirements.	Moderate	From Critical to Catastrophic
R10	New stakeholders are discovered during the development of the system.	Moderate	From Moderate to Catastrophic
R11	Agreement issues with one of the possible Internet provider that need to provide data connectivity to Cars.	Moderate	Critical
R12	Positioning system (GPS, GLONASS, Galileo, ...) availability is restricted and can no longer be used for users and Cars localization.	Low	Critical

Risk	**Mitigation Strategy**
R1	All members are aware of the main structure of the project, so in case of a missing member the others can take care of his work, and ensure work completion before deadlines; furthermore the schedule takes into account extra time to face unexpected events like this.
R2	All the already completed milestone must be updated to reflect the changes; this takes a lot of effort if done during the implementation phase. Keeping extra time at each step of the project can be useful to limit the impact; a flexible system design can help speed up the updating phase; legally binding the customer to the original requirements can ensure he can only update the requirements to make them clearer, and never to add new ones.
R3	Schedule should take into account extra time for each phase of the project in order to face unexpected events.
R4	During the design phase the possibility of a peak in requests submitted to the service must be taken into account in order to design an architecture flexible enough to properly scale when needed.
R5	If the failure is short a do not involve core functions the system may still be able to continue operating, and the impact of the failure would be seen only by a minor part of the user. If the failure last longer, most of the user might experience it, resulting in a loss of trust and the possibility of switching to a rival service. To prevent this the most fundamentals external components should be taken from a well-known and well-established provider, that can ensure its components will not face long outage.
R6	To reduce the probability of damage on system hardware it it possible to deploy the system on a cloud infrastructure, whose hardware is located in different places. However privacy laws must be taken into account when selecting the places.
R7	Any major bug should have been discovered during testing phase. Remaining bugs should not represent a big issue and can be address with a minor update on the application.
R8	Depending on the economic and marketing strength of the competitor, it might be used to consider its functions and then decided whether invest to add new functions to the system we are developing, or to speed up the release in order to reach the market before the competitor, or organize a marketing campaign to ensure our system will have a better coverage than the one of the competitor.
R9	Like *R2* but with the impossibility of binding the customers. If the extent of the required modifications are too big it is possible that the system will no longer fit the budget even with the extra time described in R2. In this case a feasibility study should be done to evaluate whether increase the budget or, if possible, cut out some functions from the system.
R10	Depending on the influence and the impact of interest of the new stakeholders we may end up in situation described in *R2* or *R9*.

R11	Since the system relies on Internet data connectivity to send data back and forth from Cars to the central system, an issue with the Internet provider would restrain the system from being deployed. Due to number of Internet provider it is unlikely that an agreement cannot be reach with any of them; however exploring different solutions requires more time than just one solution. To ensure the lowest delay possible agreement with Internet provider should be done as soon as possible.
R12	It is unlikely to happen due to the large impact that disabling the availability of a positioning system would have; however, especially after recent political events, it should be wise to rely on hardware that support at least a pair of different positioning systems.

5 Appendix

5.1 Effort Spent

Teamwork	$\sim$7h
Piccirillo Luca	$\sim$6h
Zampogna Gian Luca	$\sim$6h
Zini Edoardo	$\sim$7h

5.2 Revision History

Version	**Date**	**Changes**
1.0-RC1	22/01/2017	First deadline release.