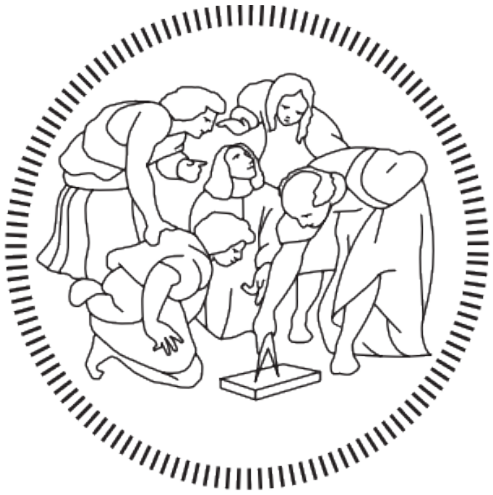


POLITECNICO
MILANO 1863

Politecnico di Milano
2016-2017 Software Engineering 2 Project:
PowerEnJoy
Project Plan Document

Version 1.0

Alessandro Polenghi 879111

Alessandro Terragni 879112

1 Introduction 4
1.1 Purpose and scope 4
1.2 List of definitions and abbreviations 4
1.3 List of reference documents 4
2 Project size, cost and effort estimation 5
2.1 Size estimation: Function points 5
2.1.1 Internal Logic Files 6
2.1.2 External Interfaces Files 7
2.1.3 External Inputs 8
2.1.4 External Outputs 9
2.1.5 Inquiries 9
2.1.6 Overall Estimation 11
2.2 Cost end effort estimation: COCOMO II 13
2.2.1 Scale Driver 13
2.2.2 Cost Drivers 16
2.2.2.1 Required Software Reliability 16
2.2.2.2 Database Size 16
2.2.2.3 Product Complexity 17
2.2.2.4 Product Reusability 18
2.2.2.5 Documentation match to life-cycle needs 18
2.2.2.6 Execution Time Constraint 19
2.2.2.7 Main Storage Constraint 19
2.2.2.8 Platform volatility 20
2.2.2.9 Analyst Capability 20
2.2.2.10 Programmer Capabilities 21
2.2.2.11 Personell Continuity 21
2.2.2.12 Application Experience 21
2.2.2.13 Platform Experience 22
2.2.2.14 Language and Tool Experience 22
2.2.2.15 Use of Software Tools 23
2.2.2.16 Multisite development 23
2.2.2.17 Required Development Schedule 24
2.2.2.18 Early design Cost Driver Calculation 24
2.2.3 Effort Equation 28
2.2.4 Schedule Estimation 28

3 Schedule Planning and Resource Allocation 29
3.1 Task Schedule 29
3.2 Resource Allocation 30
4 Risk Management 31
4.1 Project risks 31
4.2 Technical risks 32
4.3 Business risks 32
5 Appendix 33
5.1 Used tools 33
5.2 Hours of work 33

1 Introduction

1.1 Purpose and scope

This document is the Project Plan Document for PowerEnjoy.

Its main purpose is to analyze the expected complexity of the system and assist the project leader in the delicate phase of cost and effort estimation.

In the first section, we apply Function Points to estimate the size and then COCOMO II to estimate effort and cost of the system.

In the second section, we identify the tasks for your project and their schedule.

In the third section we allocate the resources (all members of the working group) to the various tasks.

Finally, we define the risks for the project, their relevance and the associated recovery actions.

1.2 List of definitions and abbreviations

- RET = Record Elements Type
- DET = Data Elements Type
- SLOC = Source Lines of Code
- PM = Person Months
- EM = Effort Multiplier

For the missing definitions, acronyms and abbreviations please refer to paragraph 1.4 of the RASD document and 1.3 of the DD document.

1.3 List of reference documents

- Design Document (DD)
- Requirements and Analysis Specification Document (RASD)
- Integration Test Plan Document (ITPD)
- COCOMO II Model Definition Manual

2 Project size, cost and effort estimation

2.1 Size estimation: Function points

The Function Points approach provides an estimation of the size of a project taking as inputs the amount of functionalities to be developed and their complexity.

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

- For Internal Logic Files and External Interfaces Files

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

- For External Output and External Inquiry

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

- For External Input

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

- UFP Complexity Weights

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

RET – Record Element Type is a user recognizable sub group of data elements within an Internal or External File.
DET – Data Element Type is a unique user recognizable, non-repetitive field.

2.1.1 Internal Logic Files

PowerEnjoy relies on a number of ILFs to store the data it needs to provide its functionalities:

- Reservation Data:
  - 7 RET: CurrentReservation, Ride, OPC, Timeout (PitStopTimeout, ReservationTimeout, PutOnChargeTimeout)
  - 5 DET: Car,Client, Payment,Duration

  Complexity: **Average**

- Payment Data
  - 1 RET: Payment
  - 3 DET: Price,Reservation, DOF

  Complexity: **Low**

- Client Data
  - 1 RET: Client
  - 9 DET: email, password, creditcardID, creditCardExpirationDate, creditCardCcv, DrivingLicenseID, DrivingLicenseExpirationDate, otherCredentials, Reservation

  Complexity: **Low**

- Parking slots Data
  - 5 RET: ParkingSlot, Position, SPA, RPA, ChargingGrid
  - 3 DET: longitude, latitude, plugged

  Complexity: **Low**

- Dof Data
  - ❖3 RET: description
  - ❖3 DET: amount(x2)

  Complexity: **Low**

ILF	Complexity	FPS
Reservation Data	Average	10
Payment Data	Low	7
Client Data	Low	7
Parking slots Data	Low	7
DOF Data	Low	7
Total		**38**

2.1.2 External Interfaces Files

- Car Data:
  - ❖9 RET : Car, Position, Car Status: To be Charged, Reserved, Available, Riding, Out Of Order, Position
  - ❖7 DET: battery level, number of passengers, engineON, doors Closed, Broken, Latitude, Longitude

  Complexity: **Average**

EIF	Complexity	FPS
Car Data	Average	7
Total		**7**

2.1.3 External Inputs

- Login: uses only a part of ILF Client Data
  - ❖ 1 RET: Client
  - ❖ 2 DET: email, password

  Complexity: **Low**

- Reservation: uses ILF ReservationData
  - ❖ 3 RET: CurrentReservation, Ride, OPC
  - ❖ 6 DET: Car,Client, Payment,Duration, ReservationTimeout

  Complexity: **Average**

- Registration: uses ILF Client Data
  - ❖ 1 RET
  - ❖ 9 DET

  Complexity: **Low**

- Edit Account: uses only a part of ILF Client Data
  - ❖ 1 RET: Client
  - ❖ 4 DET: email, password, creditcardID, creditCardExpirationDate, creditCardCcv

  Complexity: **Low**

EIs	Complexity	FPS
Login	Low	3
Registration	Low	3
Reservation	Average	4
Edit account	Low	3
Total		**13**

2.1.4 External Outputs

- Notification: payment notification and cancellation of a reservation

  - ❖ Payment notification uses the payment ILF
    Complexity : **Low**

  - ❖ Cancellation of a reservation notification uses reservation ILF
    Complexity : **Average**

EOs	Complexity	FPS
Payment notifications	Low	4
Cancellation of a reservation notification	Average	5
Total		**9**

2.1.5 Inquiries

- Check Current Reservation: it uses the reservation ILF
  - ❖ 2 RET: CurrentReservation, OPC
  - ❖ 3 DET: Car,Client, Payment

  Complexity: **Low**

- Check account: it uses the client ILF
  - ❖ 1 RET: Client
  - ❖ 8 DET: email, password, creditcardID, creditCardExpirationDate, creditCardCcv, DrivingLicenseID,DrivingLicenseExpirationDate, otherCredentials

  Complexity: **Low**

- Search a Car: it uses the Car ILF
  - 9 RET : Car, Position, Car Status: To be Charged, Reserved, Available, Riding, Out Of Order, Position
  - 4 DET: battery level, Broken, Latitude, Longitude

Complexity: **Average**

EQs	Complexity	FPS
Check current reservation	Low	3
Check account	Low	3
Search a car	Average	4
Total		**10**

2.1.6 Overall Estimation

Considering Java Enterprise Edition as a development platform we can estimate the total number of lines of code using the weights of the QSM function point languages table, attached below.

Language	QSM SLOC/FP Data			
	Avg	Median	Low	High
ABAP (SAP) *	28	18	16	60
ASP*	51	54	15	69
Assembler *	119	98	25	320
Brio +	14	14	13	16
C *	97	99	39	333
C++ *	50	53	25	80
C# *	54	59	29	70
COBOL *	61	55	23	297
Cognos Impromptu Scripts +	47	42	30	100
Cross System Products (CSP) +	20	18	10	38
Cool:Gen/IEF *	32	24	10	82
Datastage	71	65	31	157
Excel *	209	191	131	315
Focus *	43	45	45	45
FoxPro	36	35	34	38
HTML *	34	40	14	48
J2EE *	46	49	15	67
Java *	53	53	14	134
JavaScript *	47	53	31	63
JCL *	62	48	25	221
LINC II	29	30	22	38
Lotus Notes *	23	21	19	40
Natural *	40	34	34	53
.NET *	57	60	53	60
Oracle *	37	40	17	60
PACBASE *	35	32	22	60
Perl *	24	15	15	60
PL/1 *	64	80	16	80
PL/SQL *	37	35	13	60
Powerbuilder *	26	28	7	40
REXX *	77	80	50	80
Sabretalk *	70	66	45	109
SAS *	38	37	22	55
Siebel *	59	60	51	60
SLOGAN *	75	75	74	75
SQL *	21	21	13	37
VB.NET *	52	60	26	60
Visual Basic *	42	44	20	60

Function Type	Value
Internal Logic Files	38
External Logic Files	7
External Inputs	13
External Outputs	9
External Inquiries	10
Totale	**77**

Depending on the conversion rate, we have an **average** Source Lines of Code of:

SLOC = 77 * 46 = 3542

An **upper bound** of :

SLOC = 77 * 67 = 5159

2.2 Cost end effort estimation: COCOMO II

In this section we're going to use the COCOMO II approach to estimate the cost and effort needed to develop the system.

2.2.1 Scale Driver

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

Scale Factors	Very Low	Low	Nominal	High	Very High	Extra High
PREC	thoroughly unprecedented	largely unprecedented	somewhat unprecedented	generally familiar	largely familiar	thoroughly familiar
SF_j:	6.20	4.96	3.72	2.48	1.24	0.00
FLEX	rigorous	occasional relaxation	some relaxation	general conformity	some conformity	general goals
SF_j:	5.07	4.05	3.04	2.03	1.01	0.00
RESL	little (20%)	some (40%)	often (60%)	generally (75%)	mostly (90%)	full (100%)
SF_j:	7.07	5.65	4.24	2.83	1.41	0.00
TEAM	very difficult interactions	some difficult interactions	basically cooperative interactions	largely cooperative	highly cooperative	seamless interactions
SF_j:	5.48	4.38	3.29	2.19	1.10	0.00
PMAT	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

With the following meaning:

- *Precedentedness*: High if a product is similar to several previously developed projects

- *Development Flexibility*: High if there are no specific constraints to conform to pre-established requirements and external interface specs

- *Architecture / Risk Resolution*: High if we have a good risk management plan, clear definition of budget and schedule, focus on architectural definition

- *Team Cohesion*: High if all stakeholders are able to work in a team and share the same vision and commitment.

- *Process Maturity*: Refers to a well known method for assessing the maturity of a software organization, CMM, now evolved into CMMI (see additional material)

Considering our project:

- **Precedentedness**: since the team members are not expert in the field, this value will be low

- **Development flexibility**: Nominal because the developers have to face the integration of the existing CMS system for the car management.

- **Architecture\ Risk Resolution**: High because we did a good risk analysis.

- **Team Cohesion:** since the team worked together of several projects, we consider this value as very high.

- **Process maturity :** respecting to the Ratings for Estimated Project Maturity Level (EPML), as we considered our software company as an established but young reality conformed to modern software engineering standards, we decided to choose a 3 CMM Level (Maturity Level)

Scale Driver	Factor	Value
Precedentedness (PREC)	Low	2,48
Development flexibility (FLEX)	Nominal	3,04
Risk resolution (RESL)	High	2,83
Team cohesion (TEAM)	Very High	1,10
Process maturity (PMAT)	Level 3	3,12
Total		**12,57**

Now we calculate E as an aggregation of the five Scale Factors:

$$\mathbf{E} = 0{,}91 + 0{,}01 * 12{,}57 = 1{,}03$$

2.2.2 Cost Drivers

We decided to follow the **Early Design method** because we are at an early stage of the software project, thus we do not have clear information on the architecture of the system, the size of the product to be developed, the nature of the target platform and the nature of the personnel to be involved in the project.

Firstly we will calculate each cost driver and then we will combinate them for the Early design cost Driver calculation.

2.2.2.1 Required Software Reliability

RELY Descriptors:	slight inconven-ience	low, easily recoverable losses	moderate, easily recoverable losses	high financial loss	risk to human life	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.82	0.92	1.00	1.10	1.26	n/a

Given that the risk to human life is not endangered by our system and the risk of financial losses in case of malfunction is moderate and easily recoverable, we consider the RELY as **Nominal**.

2.2.2.2 Database Size

DATA* Descriptors		Testing DB bytes/Pgm SLOC < 10	10 ≤ D/P < 100	100 ≤ D/P < 1000	D/P ≥ 1000	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.90	1.00	1.14	1.28	n/a

* DATA is rated as Low if D/P is less than 10 and it is very high if it is greater than 1000. P is measured in equivalent source lines of code (SLOC), which may involve function point or reuse conversions.

At this stage of the project development we don't know yet the size of database and we can just make an estimation:
with an average SLOC of 4000, an approximate 500 mb database, the relative database size to be developed (where size refers to the amount of data to be assembled and stored in non-main storage) is :

D/P = (Database size in bytes or characters) / (Program size in SLOC)
D\P = 500 mb \ 4000 = 125000

Referring to the table above, we result with a Data Cost Driver that is **Very High**

2.2.2.3 Product Complexity

Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.73	0.87	1.00	1.17	1.34	1.74

Referring to the Component Complexity Rating Level of the COCOMO II Model Definition Manual

Our system can be placed in the range of to this characteristics of the **Nominal** Rating Level:

Complexity is divided into five areas:

- Control operation: Mostly simple nesting. Some intermodule control. Decision tables. Simple callbacks or message passing, including middleware supported distributed processing

- Computational operation: Use of standard math and statistical routines. Basic matrix/vector operations.

- Device-dependent operations: I/O processing includes device selection, status checking and error processing.

- Data management operations: Multi-file input and single file output. Simple structural changes, simple edits. Complex COTS-DB queries, updates.

- User interface management operations: Simple use of widget set.

2.2.2.4 Product Reusability

RUSE Descriptors:		none	across project	across program	across product line	across multiple product lines
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.95	1.00	1.07	1.15	1.24

This cost driver accounts for the additional effort needed to construct components
intended for reuse on current or future projects.
This effort is consumed with creating more generic design of software, more elaborate documentation, and more extensive testing to ensure components are ready for use in other applications.

In this case, We consider the Product Reusability Rating Level as **Nominal** because, even if this project is not connected to other future project, we want to develop it following SOLID principles design in a way that some modules of our system can be easily adapted to future integrations.

2.2.2.5 Documentation match to life-cycle needs

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 Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.81	0.91	1.00	1.11	1.23	n/a

We consider the DOCU as **Nominal** because we prefer to spend time to provide a right size documentation in order to avoid future extra cost during the maintenance portion of the lifecycle.

2.2.2.6 Execution Time Constraint

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 Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.11	1.29	1.63

As Power Enjoy is a quite complex system, referring to the table above, we consider a **High** Time Range Level.
In fact, the system should manage a lot of requests at the same time so the execution time usage should be considered.

2.2.2.7 Main Storage Constraint

STOR Descriptors:			≤ 50% use of available storage	70% use of available storage	85% use of available storage	95% use of available storage
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	n/a	1.00	1.05	1.17	1.46

This rating represents the degree of main storage constraint imposed on a software system or subsystem.
Given that with the new technologies is possible to have terabytes of storage for very low cost and our data are not really big, we consider the STOR Rating Level as **Low**.

2.2.2.8 Platform volatility

PVOL Descriptors:		Major change every 12 mo.; Minor change every 1 mo.	Major: 6 mo.; Minor: 2 wk.	Major: 2 mo.;Minor: 1 wk.	Major: 2 wk.;Minor: 2 days	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	n/a	0.87	1.00	1.15	1.30	n/a

For our system, especially in the first months after the release, we expect some minor changes to fix bugs and improve the system due to the clients feedbacks every two weeks.
After that, we expect to release updates on the client application every three months and eventually some updates of the system every 6 months.
Consider all this things, we take a **Nominal** PVOL Rating Level

2.2.2.9 Analyst Capability

ACAP Descriptors:	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.42	1.19	1.00	0.85	0.71	n/a

We consider the ACAP rating level as **High** because the requirements and design analysis has been done in a complete, precise and efficient way.

2.2.2.10 Programmer Capabilities

PCAP Descriptors	15th percentile	35th percentile	55th percentile	75th percentile	90th percentile	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.34	1.15	1.00	0.88	0.76	n/a

We consider the PCAP rating level as **High** because our programmer team as skillful, efficient and able to communicate and cooperate .

2.2.2.11 Personell Continuity

PCON Descriptors:	48% / year	24% / year	12% / year	6% / year	3% / year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.29	1.12	1.00	0.90	0.81	

We expect that the personnel turnover for this project will be very low, thus the PCON rating level is **very high**.

2.2.2.12 Application Experience

APEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 years	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.10	1.00	0.88	0.81	n/a

Our team works with similar technologies to the ones we are going to use in these project since last year, so we consider the APEX Rating Level as **Nominal**.

2.2.2.13 Platform Experience

PLEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.19	1.09	1.00	0.91	0.85	n/a

Our team works with similar platforms to the ones we are going to use in these project since last year, so we consider the APEX Rating Level as **Nominal.**

2.2.2.14 Language and Tool Experience

LTEX Descriptors:	≤ 2 months	6 months	1 year	3 years	6 year	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.20	1.09	1.00	0.91	0.84	

Our team has some experience with Java and , but is relatively new to Java EE, so we consider the LTEX Rating Level as **Low.**

2.2.2.15 Use of Software Tools

TOOL Descriptors	edit, code, debug	simple, frontend, backend CASE, little integration	basic life-cycle tools, moderately integrated	strong, mature life-cycle tools, moderately integrated	strong, mature, proactive life-cycle tools, well integrated with processes, methods, reuse	
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.17	1.09	1.00	0.90	0.78	n/a

This represents the degree to which software tools are used in developing the software product.
We expect a general use of tools such as virtual operating systems, database design, program design language, performance measurement and analysis, and program flow and test analyzer.
Thus we consider this parameter as **high**.

2.2.2.16 Multisite development

SITE: Collocation Descriptors:	Inter-national	Multi-city and Multi-company	Multi-city or Multi-company	Same city or metro. area	Same building or complex	Fully collocated
SITE: Communications Descriptors:	Some phone, mail	Individual phone, FAX	Narrow band email	Wideband electronic communicat ion.	Wideband elect. comm., occasional video conf.	Interactive multimedia
Rating Levels	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.22	1.09	1.00	0.93	0.86	0.80

Our team live in the same city and will work together for the majority of the time, so we consider the SITE Rating Level as **High**.

2.2.2.17 Required Development Schedule

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 Multiplier	1.43	1.14	1.00	1.00	1.00	n/a

This rating measures the schedule constraint imposed on the project team developing the software.
The ratings are defined in terms of the percentage of schedule stretch-out or acceleration with respect to a nominal schedule for a project requiring a given amount of effort.
In our case the SCED Rating Level is set to **Nominal**, because we expect to meet the deadlines in time, without stretching or accelerating too much the nominal schedule.

2.2.2.18 Early design Cost Driver Calculation

The following approach is used for mapping the full set of Post-Architecture cost drivers and rating scales onto their Early Design model counterparts. It involves the use and combination of numerical equivalents of the rating levels.

Early Design Cost Driver	Counterpart Combined Post-Architecture Cost Drivers
PERS	ACAP, PCAP, PCON
RCPX	RELY, DATA, CPLX, DOCU
RUSE	RUSE
PDIF	TIME, STOR, PVOL
PREX	APEX, PLEX, LTEX
FCIL	TOOL, SITE
SCED	SCED

PERS: Personnel Capability

PERS Descriptors:							
• Sum of ACAP, PCAP, PCON Ratings	3, 4	5, 6	7, 8	9	10, 11	12, 13	14, 15
• Combined ACAP and PCAP Percentile	20%	35%	45%	55%	65%	75%	85%
PERS Descriptors:							
• Annual Personnel Turnover	45%	30%	20%	12%	9%	6%	4%
Rating Levels	Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	2.12	1.62	1.26	1.00	0.83	0.63	0.50

ACAP + PCAP + PCON = HIGH + HIGH + VERY HIGH = 4 + 4 + 5 = 13

PERS rating level = **Very High**

RCPX: Product Reliably and Complexity

RCPX Descriptors:							
• Sum of RELY, DATA, CPLX, DOCU Ratings	5, 6	7, 8	9 - 11	12	13 - 15	16 - 18	19 - 21
• Emphasis on reliability, documentation	Very Little	Little	Some	Basic	Strong	Very Strong	Extreme
• Product complexity	Very simple	Simple	Some	Moderate	Complex	Very complex	Extremely complex
• Database size	Small	Small	Small	Moderate	Large	Very Large	Very Large
RCPX Descriptors:							
Rating Levels	Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.49	0.60	0.83	1.00	1.33	1.91	2.72

RELY + DATA + CPLX + DOCU = NOMINAL + VERY HIGH + NOMINAL + NOMINAL = 3 + 5 + 3 + 3 = 14

RCPX rating level = **High**

RUSE: Reusability

RUSE rating level = **Nominal**

PDIF: Platform Difficulty

PDIF Descriptors:					
• Sum of TIME, STOR, and PVOL ratings	8	9	10 - 12	13 - 15	16, 17
• Time and storage constraint	≤ 50%	≤ 50%	65%	80%	90%
• Platform volatility	Very stable	Stable	Somewhat volatile	Volatile	Highly volatile
Rating Levels	Low	Nominal	High	Very High	Extra High
Effort Multipliers	0.87	1.00	1.29	1.81	2.61

TIME + STOR + PVOL = High + Low + Nominal = 4 + 2 + 3 = 9

PDIF rating level = **Nominal**

PREX: Personnel Experience

PREX Descriptors:							
• Sum of APEX, PLEX, and LTEX ratings	3, 4	5, 6	7, 8	9	10, 11	12, 13	14, 15
• Applications, Platform, Language and Tool Experience	≤ 3 mo.	5 months	9 months	1 year	2 years	4 years	6 years

PREX Descriptors:							
Rating Levels	Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.59	1.33	1.22	1.00	0.87	0.74	0.62

APEX + PLEX + LTEX = Nominal + Nominal + Low = 3 + 3 + 2 = 8

PREX rating level = **Low**

FCIL: Facilities

FCIL Descriptors:							
• Sum of TOOL and SITE ratings	2	3	4, 5	6	7, 8	9, 10	11
• TOOL support	Minimal	Some	Simple CASE tool collection	Basic life-cycle tools	Good; moderately integrated	Strong; moderately integrated	Strong; well integrated
• Multisite conditions	Weak support of complex multisite development	Some support of complex M/S devel.	Some support of moderately complex M/S devel.	Basic support of moderately complex M/S devel.	Strong support of moderately complex M/S devel.	Strong support of simple M/S devel.	Very strong support of collocated or simple M/S devel.
Rating Levels	Extra Low	Very Low	Low	Nominal	High	Very High	Extra High
Effort Multipliers	1.43	1.30	1.10	1.0	0.87	0.73	0.62

FCIL = TOOL + SITE = HIGH + HIGH = 4 + 4 = 8

FCIL rating level = **High**

SCED: Required Development Schedule

SCED rating level = **Nominal**

Effort Multiplier Calculation

Cost Driver	Factor	Value
PERS	Very High	0,63
RCPX	High	1,33
RUSE	Nominal	1
PDIF	Nominal	1
PREX	Low	1,22
FCIL	High	0.87
SCED	Nominal	1
Total (product)		1,022238

2.2.3 Effort Equation

$$PM = A \times Size^{E} \times \prod_{i=1}^{n} EM_i$$

$$\text{where } A = 2.94 \text{ (for COCOMO II.2000)}$$

Average :
PM = 2,94 * 3,542 ^(1,03) * 1,022238 = 11,057 person months

Upper bound:
PM = 2,94 * 5,159 ^(1,03) * 1,022238 = 16,287 person months

2.2.4 Schedule Estimation

$$TDEV = [C \times (PM_{NS})^{(D+0.2\times(E-B))}] \times \frac{SCED\%}{100}$$

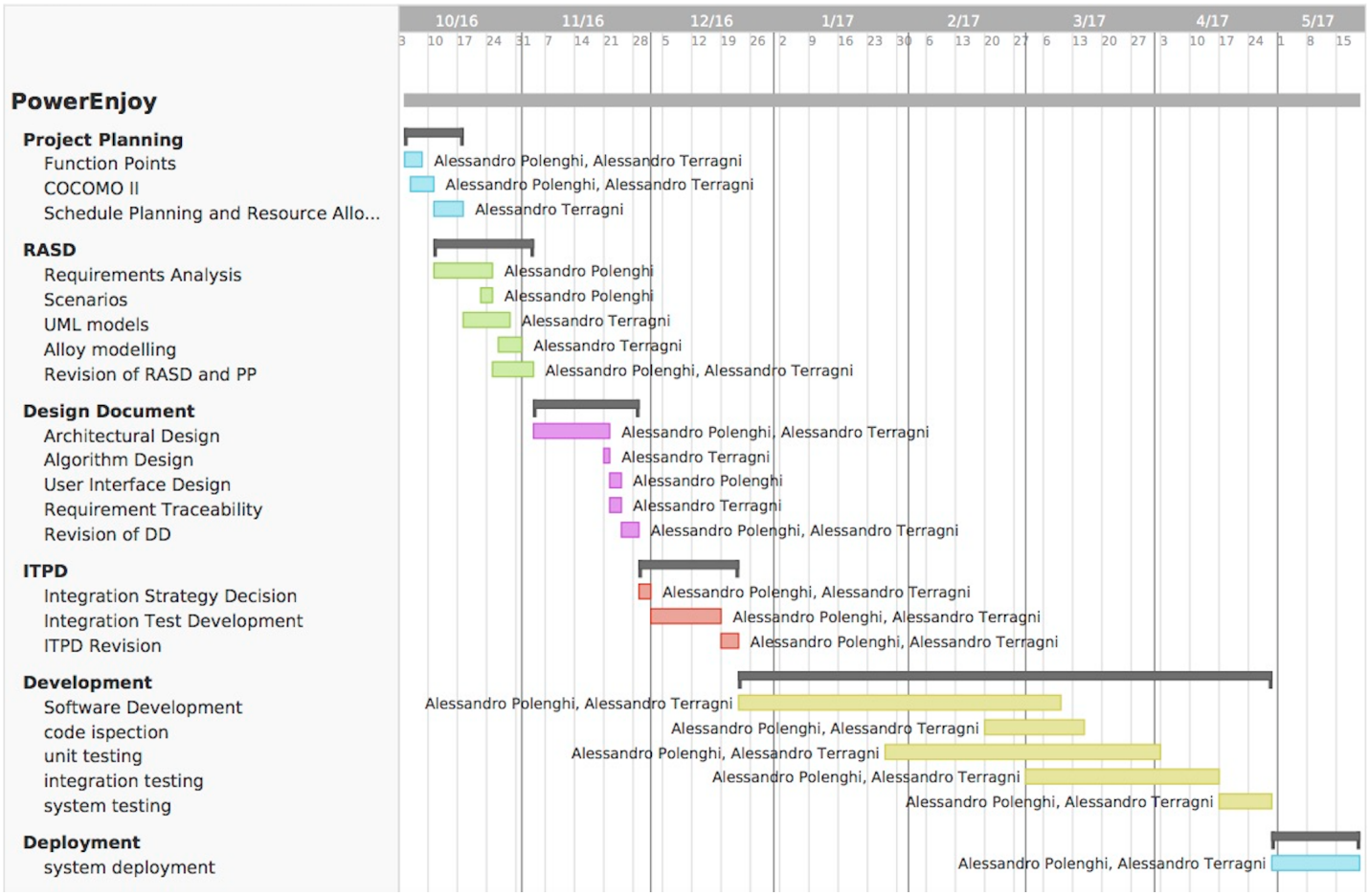
$$\text{where } C = 3.67, D = 0.28, B = 0.91$$

Average:
TDEV = [3,67 * (11,057)^(0,28 + 0,2 * (1,03 - 0,91))] * 100\100 = 7,6195 months

Upper bound:
TDEV = [3,67 * (16,287)^(0,28 + 0,2 * (1,03 - 0,91))] * 100\100 = 8,5716 months

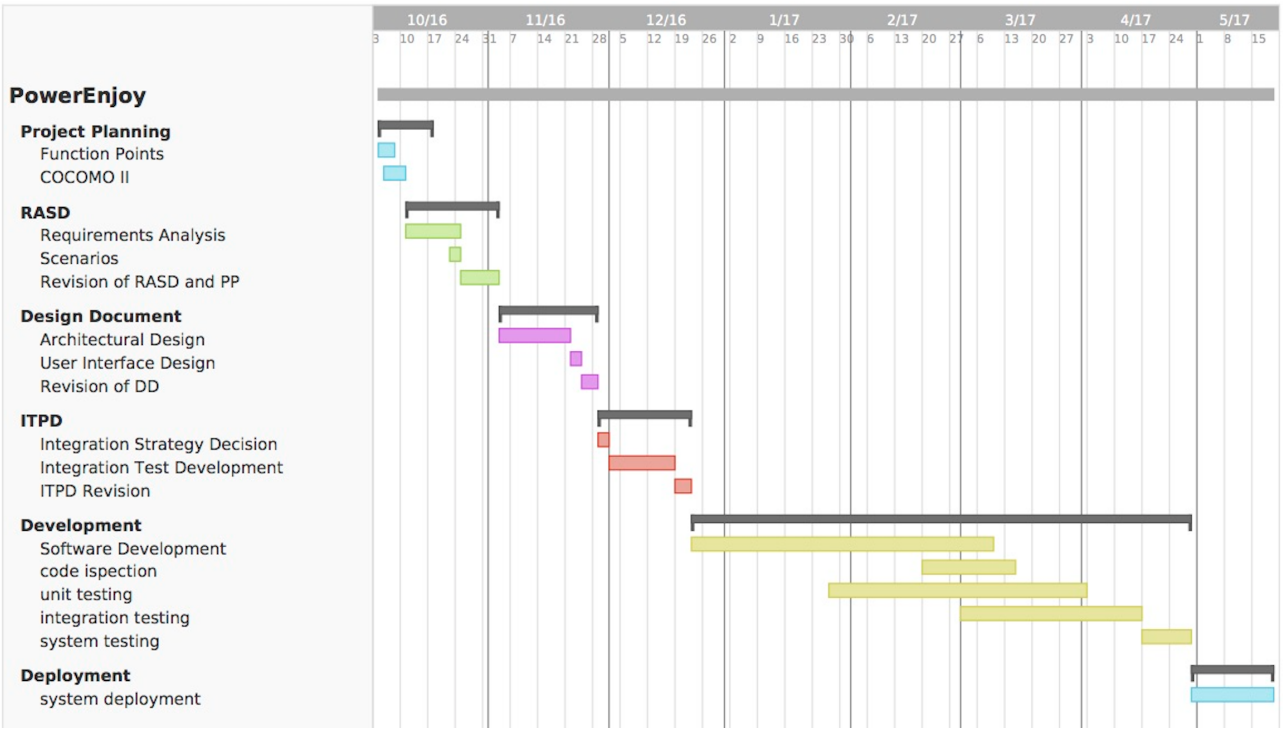
3 Schedule Planning and Resource Allocation

3.1 Task Schedule

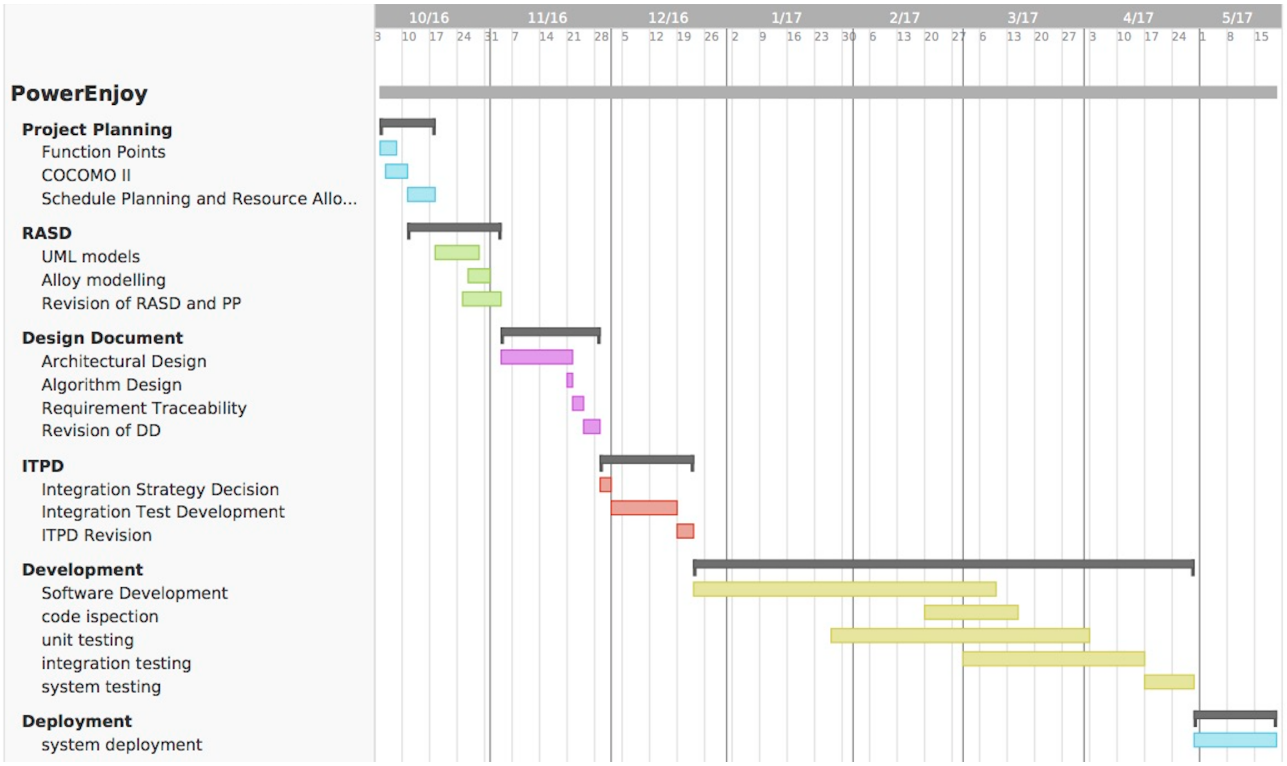


3.2 Resource Allocation

- Alessandro Polenghi



- Alessandro Terragni



4 Risk Management

During the project we have to consider the possibility of facing some potential problems.
In particular, we can divide the risks in three categories:

- Project risks: they threaten the project plan. If they become real, it is likely that the project schedule will slip and that costs will increase

- Technical risks: they threaten the quality and timeliness of the software to be produced
  If they become real, implementation may become difficult or impossible

- Business risks: they threaten the viability of the software to be built. If they become real, they jeopardize the project or the product

4.1 Project risks

Risk	Probability	Strategy	Effects
Team communications and integration problems	Moderate	Alert customer to potential difficulties and the possibility of delays and improve team working through team building training.	Serious
Key staff are ill at critical times in the project.	Moderate	Alert customer to potential difficulties and the possibility of delays; investigate buying-in components.	Moderate
Changes to requirements that require major design rework are proposed	Moderate	Delay the project schedules, derive traceability information to assess requirements change impact; maximize information hiding in the design.	Serious
The organization is restructured so that different management are responsible for the project	Very Low	Reschedule the schedule and resource allocation	Serious

4.2 Technical risks

Risk	Probability	Strategy	Effects
Problem in the CMS integration	Moderate	Ask the client to redesign the CMS	Catastrophic
Failures during Testing	Moderate	Alert customer to potential difficulties and the possibility of delays; solve and fix the bugs found	Moderate
Insufficient Performance	Low	Redesign the architecture or evaluating the purchase of powerful machines	Catastrophic

4.3 Business risks

Risk	Probability	Strategy	Effects
Organizational financial problems force reductions in the project budget.	Low	Prepare a briefing document for senior management showing how the project is making a very important contribution to the goals of the business and presenting reasons why cuts to the project budget would not be cost- effective.	Catastrophic
New Governmental constraints	Low	Rediscuss the requirements according to the new constraints. Prepare a briefing document for senior management showing how the project has changed and inform them about the possibility of delays	Serious

5 Appendix

5.1 Used tools

- Pages
- TeamGantt.com

5.2 Hours of work

• Alessandro Polenghi
13\01 1h
16\01 1.5 h
17\01 2h
18\01 7.5 h

• Alessandro Terragni
13\01 1h
16\01 1.5 h
17\01 2h
18\01 7.5 h