PowerEnjoy - Project Plan

January 13, 2017

A.Y. 2016/2017
Bolshakova Liubov, matr. 876911
Gao Xiao, matr. 876265
Kang Shuwen, matr. 876245

Contents

1 INTRODUCTION

1.1 Purpose

This document represents the Project Plan Document in order to specific the evaluation of effort and cost which need in developing the PowerEnJoy project. In this document ,we use COCOMO and Function Point algorithmic techniques to to clarify the total work we should do as well as the technique support. We also focus on the risks we have and enumerate basic strategies to avoid them.

1.2 Definitions and Abbreviations

- Car: the cars that supplied for the car-sharing service in the *PowerEnJoy* system.
- Car information: the basic information that helps guests and users to make decisions, include the dump energy, location information, distance to the setting location, the passenger capacity.
- Starting position: the current position of user or the positions user input to start a ride.
- Available car: the car has dump energy more than 50
- Available queue: a queue that maintains available cars
- Sensors: the GPS and power plug sensor, weight sensor, display screen, battery sensor, door state sensor, locks of door in the car, and the sensor on the power grid.
- System: the whole system, which include the electric devices and the *PowerEnJoy* system background.
- Ride: in this system, the ride process is started with ignite the engine and ended with all passengers leave the car.

2 FP ESTIMATION

2.1 Breve Introduction

Function Points is part of the algorithmic techniques for cost and estimation modelling. It bases on the assumption that the dimension of the software can be characterized by abstraction. The main function points are based on following aspects.

2.2 Internal Logic Files

The system include some Internal Logic Files (IFL) to store the related persistent information, which are the users account information, the car information about battery, state and location. We are now going to explain the ITF as following:

- Users account: four different tables are expected, listed below:
  - User information:ID(key), name, surname, driving license information, encrypted password, email address, credit card ID.
  - APP keys:ID(key),linked email address, encrypted password.
- Car information:ID(key), dump energy, car state, capacity of passenger.

ILF	Complexity	FP
User account	Avg	10
Car information	Low	7

Table 1: ITF

2.3 External Interface Files

To provide an efficient service, our system also connected to some external applications, the system need External Logic Files (EIF) to store those data:

- Google API: geographical data which is needed to locate the GPS of car and the user.
- Traffic system: data about the validation of users' driving licenses.
- Bank system: data about the validation of credit card which users supplied, also the data of payment information after the driving or reservation processes.

EIF	Complexity	FP
Google API	Low	5
Traffic system	Low	5
Bank system	Low	5

Table 2: EIF

2.3.1 External Input

The system allows users to maintain Internal Logical Files (ILFs) in order to add, change and delete the data.

According to the functional requests which are listed in the RASD document, there are the functionalities can be considered as External Input.

- Allows a guest executes the registration to become a user.
- Allows users to login into the system.
- Allows users to search the available cars near the starting position.
- Allow users to make reservation.
- Allow users to cancel the reservation.
- Allows the users to pick up the car.
- Allow users to access the *Saving Money Option*.

EI	Complexity	FP
Allows a guest executes the registration to become a user.	Low	3
Allows user to login into the system.	Low	3
Allows the users to search the available cars near the starting position.	Low	3
Allow users to make reservation.	Avg	4
Allow users to cancel the reservation.	Low	3
Allow the users to pick up the car.	Low	3
Allows the users to pick up the car.	Low	3

Table 3: EI

2.3.2 External Output

This Transactional function point can realize the production of outputs. For the aim of accomplishing services, system should response the result data to users, the data that responded is called External Output (EO).

According to the RASD document , the External Output functionalities are as follows:

- Produce the list of available cars near the starting position got by GPS signal.
- Produce the optimal parking station according to the destination user inputed.
- Produce the amount of money user should pay after the driving or missing the reservation.
- Produce an external programmatic interface to the external applications to communicate with the system and get the information they need for services.

EO	Complexity	FP
List of available cars	Low	4
Optimal parking position	Low	4
Money in the payment	Low	4
Programmatic interface	Low	4

Table 4: EO

2.3.3 External Inquiries

To accomplish the interaction between the user and system, there should have a function to accept the user selections from the displayed and listed data from files. In this process , the data was not be changed but only a direct retrieval of information contained on the files. These processes are referred to as External Inquiries (EQ).

In accordance to the RASD documents, we have identified these points:

- Display the location of user and the cars on the map.
- Display ride information includes recharge money, dump energy of cars. Visualize the list of available cars around the starting position.

- Display the optimal destination when users choose the saving money option.
- Display the information of cars.
- Display the countdown time in the screen of mobile phone after user made a reservation.

EQ	Complexity	FP
Display location of users and cars	Low	3
Display the ride information	Low	3
Display the list of available cars	Low	3
Display the optimal destination	Low	3
Display the car information	Low	3
Display the countdown of one hour	Low	3

Table 5: EQ

2.3.4 Final Estimation

Now we look back at the Function Points assigned in the previous part, we have a overview:

Function Type	Value
Internal Logic Files	17
External Interfaces Files	15
External Input	22
External Inquiries	16
External Output	18
Total	88

Table 6: Final Estimation

Lines of Code now we can estimate the number of SLOC (Source Line of Code). SLOC is the necessary number of code lines we need for our software. In the URL which is specified below, we can find the AVG

(Average number of LOC per FP for a given language). As we choose the java as the language for this program, we have:

$$\frac{SLOC}{FP} = 46 \tag{1}$$

so the expected number of SLOC is :

$$46 \times 88 = 4048 \tag{2}$$

3 EFFORT AND COST ESTIMATION

3.0.1 Introduction of COCOMO

COnstructive COost MOdel (COCOMO) is a cost estimation model which based on scale drivers and cost drivers, help to determine the general effort, duration and number of people needed for the project. The value of COCOMO are also involved by FP value we calculated in the previous part.

3.0.2 Scale Drivers

- Precedentedness

  This value is proportional to the experience we have for similar projects. Since this is our first time using this framework and method, we don't have many past example can be consulted , so this value must be very low.

- Development Flexibility

  It reflects the degree of flexibility that exists when implementing the project. This value is high, because we our work is based on single page goals document.

- Architecture/Risk Resolution

  Reflects the degree of uncertainty about requirement. Since the electric-car sharing applications have been used for many years, and we know clearly how this service works, the user requirements is quite certain, so the value is low.

- Team Cohesion

  A process where the group stays together in pursuit of a goal, it depends mostly on the member interact and interdependency. In our group, we have three member who are from different countries, so we have a bit difficulty in the communication and opinion, but it decreased gradually while we work together, so this value is high.

- Process maturity

  This was evaluated around the 18 Key Process Area (KPAs) in the SEI Capability Model. We have arrived the level 2 repeatable, so the value of this is normal.

3.0.3 Cost Drivers

- Required Software Reliability

  This is the measure the stability of the software in a period of time, and the reliability is connected directly with the consequences would

have if there are failures or errors exist in the software. In our case , the value is low because there would not induce heavy consequences even the software fails in the execution.

- Data Base Size

  The database is the base infrastructure of a software, the size of database is an important factor to the capability of a service. Since in this example we don't have a large amount of data,the value will be low.

- Product Complexity

  Complexity is divided into five areas: control operations, computational operations, device- dependent operations, data management operations, and user interface management operations. Select the area or combination of areas that characterize the product or a sub-system of the product. The complexity rating is the subjective weighted average of these areas.

  In accordance with the new COCOMO II CPLEX rating scale, this value will be high.

- Required Re-usability

  This cost driver accounts is defined as the capability to use in a project requirements that have already been used before in other projects. This allows optimizing resources during development and reduce errors.

  We have chosen a nominal value,for in this case the architecture and methods are quite relative to the specific requests.

- Documentation match to life-cycle needs

  The cost driver is evaluated in terms of the suitability of the project's documentation to its life-cycle needs. The value of this is high, because we followed the standard processing of documentation and write the document step by step.

- Analyst Capability

  Analysts are personnel that work on requirements, high level design and detailed design. The major attributes that should be considered in this rating are Analysis and Design ability, efficiency and thoroughness, and the ability to communicate and cooperate. We chosen high for this value.

- Programmer Capability

  Evaluation is based on the capability of the programmers as a team rather than as individuals. Major factors which should be considered

in the rating are ability, efficiency and thoroughness, and the ability to communicate and cooperate. This parameter is evaluated according to our degree of cooperation,so the value of this is high due to the excellent cooperation of our team.

- Application Experience

  This rating is dependent on the level of applications experience of the project team developing the software system or subsystem. The ratings are defined in terms of the project team's equivalent level of experience with this type of application. A very low rating is for application experience of less than 2 months. A very high rating is for experience of 6 years or more. From the developing time aspect , our value is low, because we spend less than 6 month on this project according to your document schedule.

- Platform Experience

  This rating is based on the capabilities of understanding the use of powerful platforms, including more graphic user interface, database and networking. The value of this scale is low for our team because of the lack of experience.

- Language and Tool Experience

  This is a measure of the level of programming language and software tool experience of the project team developing the software system or subsystem. Software development includes the use of tools that perform requirements and design representation and analysis, configuration management, document extraction, library management, program style and formatting, consistency checking, etc. In addition to experience in programming with a specific language the supporting tool set also effects development time. A low rating given for experience of less than 2 months. A very high rating is given for experience of 6 or more years. For this scale of factor our value is low.

- Personnel Continuity

  The rating scale for Continuity is in terms of the project's annual personnel turnover,We've selected a value very high because all members of our group are focus on the project from the beginning .

- Platform Volatility

  It represents the volatility of the platform. Because the principal users are mobile clients, we know this kind of technology is continually updating. Then the value is set as high.

- Use of Software Tools

In accordance with COCOMO table, the value of our team is very low because we only edit and debug the code.

- Multisite Development

  Given the increasing frequency of multisite developments, and indications that multisite development effects are significant, the SITE cost driver has been added in COCOMO II. Determining its cost driver rating involves the assessment and averaging of two factors: site collocation (from fully collocated to international distribution) and communication support (from surface mail and some phone access to full interactive multimedia).

  We used narrow-band email and phone for the communication with users, so the value is normal.

- Required Development Schedule

  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. Accelerated schedules tend to produce more effort in the later phases of development because more issues are left to be determined due to lack of time to resolve them earlier. A schedule compress of 74% is rated very low. A stretch-out of a schedule produces more effort in the earlier phases of development where there is more time for thorough planning, specification and validation. A stretch-out of 160% is rated very high. We have chosen the scale normal,because we follow the developing step by step on time.

3.0.4 Effort Estimation

Base on the COCOMO formula, we translate the level of cost driver and scale drivers into the numbers, the very low counts 5, low counts 4, normal counts 3, high counts 2,very high counts 1,extra high counts 0.

After the calculation we get the value of cost drivers is EAF= 1.00, and scale drivers is E=1.0997 in order to calculate the effort, inserting this values in the formula:

$$effort = 2.94 \times EAF \times (KSLOC)^{E} \tag{3}$$

$$effort = 2.94 \times 100 \times (4.048)^{1.0997} = 13.68 Person/Month \tag{4}$$

The duration of the project can be found with this formula, considering the exponent E = 0.3179:

$$Duration = 3.67 \times (effort)^{E} \tag{5}$$

$$Duration = 3.67 \times (1.368)^{0.3179} = 8.43 Month \tag{6}$$

Now it is possible to calculate the number of people needed for this project.

$$N.people = effort/Duration \tag{7}$$

$$N.people = 13.68/8.43 = 1.62 Person \tag{8}$$

4 RISKS AND RECOVERIES

Risk management is an important issue for the development of a project, thus it is impossible to develop software with zero risks, and risks can affect the project schedule as well as the quality of the product, it is necessary to estimate the risks before we start doing the project.

These are the risks that could be found:

4.0.1 Project Risk

This is a critical risk: it is possible that while going on the implement of system, there exists some request misunderstanding in the late phase in the project. In this situation, following the previous schedule would be a hard work and the product may finish after the estimated time.

To avoid this, we can take the strategy of building many basic prototypes with GUI, and investigate with some users, this way can avoid the heavy modifications when most of the system has already been implemented.

4.0.2 Market Risk

For developing a software, the main problem is if the market accept this software, in another words, if we can sell this software and make interests. The competitions between software who has similar functions is the main factor, in order to avoid this situation, the investigation face to the market is very important.

4.0.3 Staff Experience

It is possible that the lack of technical experience of the project developers affect the quality of the software and the schedule of the project.

In order to prevent this from happening, the team of software devolving should be guide by someone who has plenty experience of this kind of projects.

5 EFFORT SPENT

Gao Xiao	15 Hours
Kang Shuwen	12 Hours
Liubov Bolshakova	10 Hours

6 REFERENCES

6.1 Reference Documents

- Specification Document Assignments AA 2016-2017
- RASD
- Design Document
- Integration Test Plan Document
- sunset.usc.edu/research/COCOMOII/expert_cocomo/drivers.html
- http://www.qsm.com/resources/function-point-languages-table

6.2 Used Tools

The tools used to creat this document are:

- Github: for version control
- Latex: for typesetting