SOFE 3490U - Software Project Management

Lab 3

Project: UberRUSH

Name	ID
Aakash Patel	100616630
Karan Patel	100621178

Introduction

In this lab, we firstly had to calculate the estimated effort to complete the project. Secondly, we had to create an activity network diagram for our project which shows the overall activities which are to be done in the project. In the activity diagram we had to specify an estimation of the total time it will take in weeks to get the particular activity finished. Lastly, we documented the risks involved in the project with their description, the kind of risk it is and the countermeasures for the activity.

Software Project Estimation

Basic CoCoMo

In basic CoCoMo, the effort is the function of the number of lines of code and the constants associated with the type of software system to be built.

To estimate the effort of UberRUSH using the CoCoMo method we must define the type of system that UberRUSH is. Since Uber and UberEats have the same functionality and requirements as UberRUSH, it can be said that this problem has been solved before and this isn’t a big scale project like a electric car so this project can be defined as an organic system. The values of the constant ‘a’ for organic system is 2.4 and constant ‘b’ is 1.05. The KLOC has been agreed to be 10.

$$E = a(KLOC)^b$$
$$E = 2.4(10)^{1.05}$$
$$E = 27\ Person - months$$

The basic CoCoMo estimation method gives an effort estimation of 27 peron- months.

Intermediate CoCoMo

In intermediate CoCoMo, the lines of code, constants based on the type of software system to be built and various cost drivers are used for effort estimation. Since we determined that UberRUSH is a organic system, in the intermediate CoCoMo model, the value of ‘a’ is 3.2 and ‘b’ is 1.05. In the formula for intermediate CoCoMo, another factor is taken into account which is the Effort Adjustment Factor

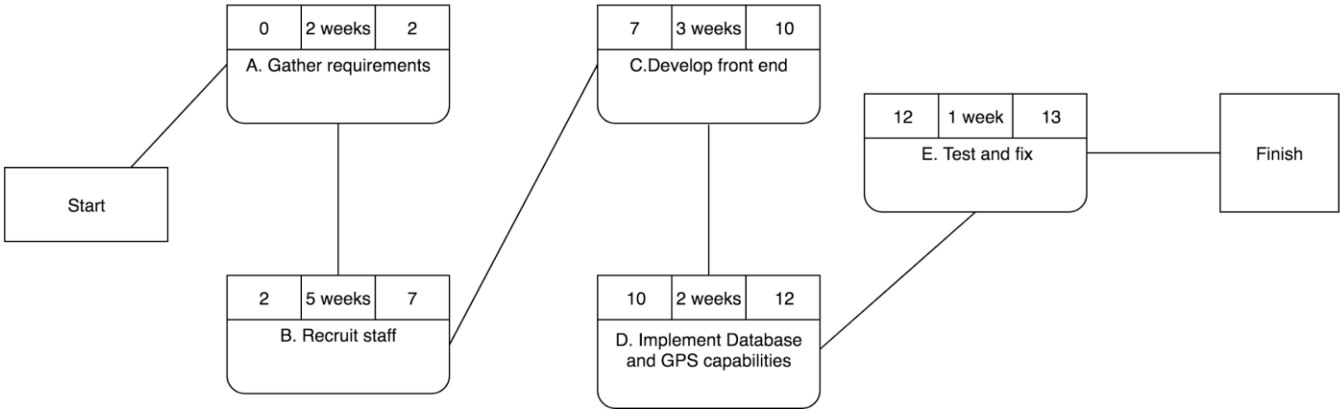
Classification of Cost Drivers and their attributes:

Cost Drivers	**Rating**	**Values**
Product Attributes		
Required Software Reliability	high	1.15
Size of Application Database	high	1.08
Complexity of the product	nominal	1.00
Hardware Attributes		
Runtime Performance Constraints	low	0.90
Memory Constraints	low	0.95
Volatility of the virtual machine environment	low	0.87
Require turnabout time	high	1.07
Personal attributes		
Analyst capability	nominal	1
Applications experience	high	1.03
Software engineer capability	nominal	1
Virtual machine experience	nominal	1
Programming language experience	high	0.95
Project Attributes		
Application software engineering methods	high	0.91
Use of software tools	nominal	1
Required development schedule	nominal	1
EAF = 0.88		

$$E = a(KLOC)^{b} * EAF$$
$$E = 3.2(10)^{1.05} * 0.88$$
$$E = 33.57 \text{ person - months}$$

The intermediate CoCoMo method gives an estimated effort of about 34 person-months. This is just above the effort estimated by the basis CoCoMo method.

Activity Diagram



Risks

There are many risks that would be associated with the project. The various risks are:

- Much longer time then the estimated 27 to 36 months to make the project which would cause a delay.
  - ❖ Countermeasure would be to train employees as much as possible before development of the project starts so that less time is wasted in the developing stage to train employees.
- Misunderstanding of requirements by the developers which may cause delays.
  - ❖ Clarify the requirements with stakeholders if developers are unsure of the specifications of the project.
- Unwanted behaviour may cause delays like if a developer is to confident and underestimates the duration that it will take to complete a task and he/she begins to waste time by leaving tasks for the end.
  - ❖ Countermeasures could be to have every developer show their progress either every day or every week to ensure that developers are on top of their tasks and focused.
- Because the estimated effort is 27 to 36 months, some functionality would have to left out of the software because of pressure to deploy the software.
  - ❖ Countermeasure should be to inform stakeholders about the amount of time the development may take and ask them to go over requirements and make sure every requirement is essential.

- Additional requirements might be asked to be added into the project which could cause more delay and more bugs in the software due to the pressure to deliver considering how long the estimated effort is (27 to 36 months). This could create last minute hurdles to meet requirements if the requirement is added at a late point in the development process.
    - ❖ Countermeasure could be inform stakeholders of the risks of adding more requirements in the later stages of the project as it would cause delays.

- Many problems may arise if there is more than anticipated traffic going to one server at once from the same region. This would cause a major problem for clients accessing the UberRUSH service and may even cause delay for the drivers using the application.
    - ❖ Countermeasure would be, to add additional servers for regions with heavy traffic and effectively manage them to balance the traffic for that region.