Software Project Management Lab 3

UberRUSH

February 20, 2019

Mohamed Ibrahim 100626201
Amalnnath Parameswaran 100585138
Mohtasim Siddiqui 100635463

Project estimation:

COCOMO Model:

- Since our organization has created Uber, a similar application using a similar language and development style, we are considering this project to be an organic project. When looking at the lines of code, Android's Uber application equates to approximately 400,000 LOC and the iOS version has 600,000 LOC. We can also estimate that the web application will have 200,000 LOC. Therefore, when using these values, we can safely estimate that the initial effort for this project is calculated to be 1,727 + 2,644 + 834 = 5205 person-months. This may look daunting, however, all this means that we should hire more developers in order to keep delivery times reasonable. Approximately 200 developers will result in 26 months (around 2 years) for delivery. Many of the code can also be reused in different stages of the development so it can also help reduce the effort needed for the application.

Android app calculation

$$E = a(KLOC)^b$$
$$E = 3.2\ (400^{1.05})$$
$$E = 1727$$

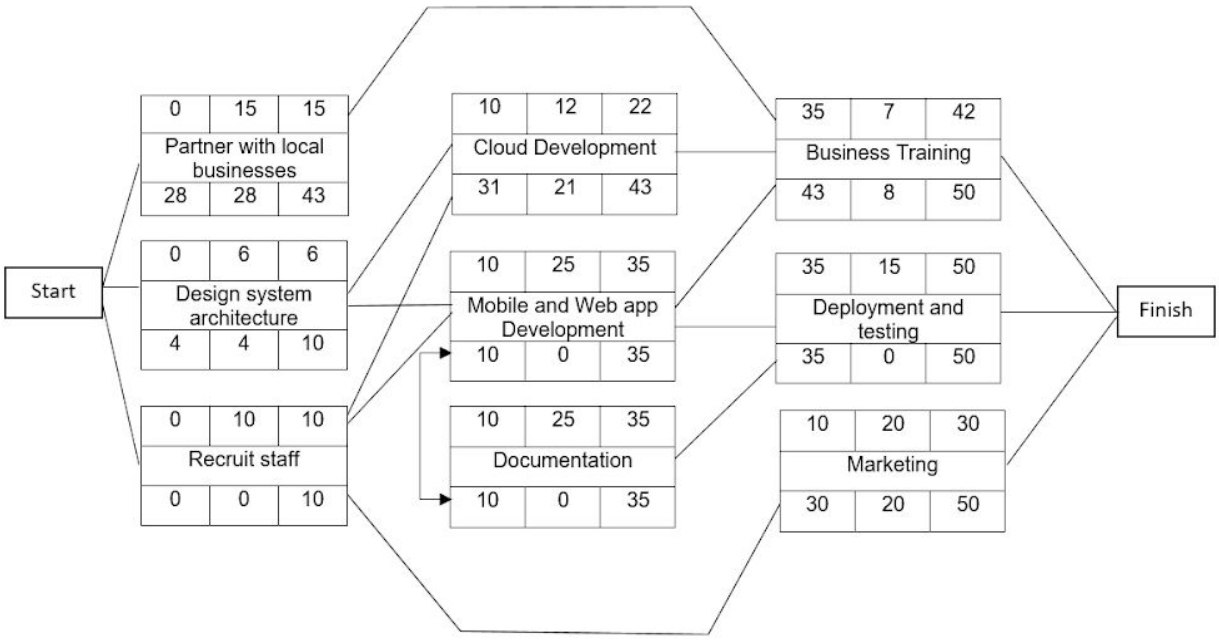
iOS app calculation

$$E = a(KLOC)^b$$
$$E = 3.2\ (600^{1.05})$$
$$E = 2644$$

Web app calculation

$$E = a(KLOC)^b$$
$$E = 3.2\ (200^{1.05})$$
$$E = 834$$

Activity diagram:



Project risks:

Actors	**Technology**	**Structure**	**Tasks**
Staff Shortage: We may not have enough people for deliveries. *Untrained Staff:* Drivers may not be the best food handlers.	*Real Time:* Constant and real-time update may not be available in remote locations. Not all the drivers may have iphone or android. *Resource:* Cloud server may limit to certain number of request per hour.	*Troubleshooting:* Development team may be required on customer service to solve challenging technical problem.	*Business partnership:* Building a partnership with other business may take several tries. *AD campaign:* may take more attention than development. Our assumptions about the potential users could be wrong

Risk	Risk reduction and countermeasures
Real-time performance problem *Real Time & Resource*	● Choose scalable framework *Ex: React* ● Rent server space on different locations to get wider coverage.
Unrealistic time estimates *Business partnership & AD Campaign*	● We may not have as much market for the product as we think. ● Prioritize the features and build a modular software ● Choose Incremental style of delivery to get the product to market quickly. Additional and fully featured software can be released according to market response ● Spend more time on advertising and getting market response to the product. ● Have time bounded contractual agreement with other companies. ● Base contract on objective standards which are fair to both parties ● Double the time estimate to stay out of trouble.
Personnel shortfalls *Staff Shortage & Untrained Staff*	● Hire minimum staff that may be necessary to carry the job ● Provide Drivers with baggage for keeping food safe ● Provide special training for each of the staff hired.
Shortfalls in externally performed tasks *Troubleshooting*	● Ticketing System to handle customer related inquiries and dedicate some development staff to handle complex technical issues.