Software Project Management Lab 3

Sunny Patel (100620076)

Ethan Wallace (100632495)

Gabriel Gelgor (100614547)

Software Effort Estimation using COCOMO

Step 1: Estimation of software size

We will estimate the software size of this project to be 13500 KLoC.

For estimation purposes, we have taken into consideration the past software we have designed and the respective application sizes in that respect.

Project 1:

Project	Lines of Code
Inventory management and distribution center control web-application:	12081
Portfolio with blogging capabilities	8093 (ignored, because the complexity level is far lower than the project at hand)
Sports equipment borrowing application	14021

*lines of code was found using an automated script that ignored npm-installed files (as we didn't write those)

We have arrived at this estimate based on the fact that we will have to write a fully-featured and tested web application. This is within our team's experience, and something we have done many times. Our team will not have to write code to interface with the worn dongle hardware or bluetooth code as this will be done through an API. This work is new to our team, despite that we do not expect it to be an major integration roadblock.

Step 2: Determining Effort Adjustment Factor (EAF)

Run through the following table and redo some shit:

Cost Drivers	Ratings					
	Very Low	Low	Nominal	High	Very High	Extra High
Product attributes						
Required software reliability	0.75	0.88	1.00	1.15	1.40	
Size of application database		0.94	1.00	1.08	1.16	
Complexity of the product	0.70	0.85	1.00	1.15	1.30	1.65
Hardware attributes						
Run-time performance constraints			1.00	1.11	1.30	1.66
Memory constraints			1.00	1.06	1.21	1.56
Volatility of the virtual machine environment		0.87	1.00	1.15	1.30	
Required turnabout time		0.87	1.00	1.07	1.15	
Personnel attributes						
Analyst capability	1.46	1.19	1.00	0.86	0.71	
Applications experience	1.29	1.13	1.00	0.91	0.82	
Software engineer capability	1.42	1.17	1.00	0.86	0.70	
Virtual machine experience	1.21	1.10	1.00	0.90		
Programming language experience	1.14	1.07	1.00	0.95		
Project attributes						
Application of software engineering methods	1.24	1.10	1.00	0.91	0.82	
Use of software tools	1.24	1.10	1.00	0.91	0.83	
Required development schedule	1.23	1.08	1.00	1.04	1.10	

The total EAF, equal to the sum of all weights is: 0.384513

Step 3: Justification of Software Project Type

In our project, we are assuming a semi-detached project in terms of the a and b coefficient. Our justification for this is that while a fall-detection app is non-trivial, a number of the key systems are already created and we are merely using the API. The bluetooth layer, and any layer that touches the hardware on the worn dongle (i.e., the system accelerometer) will be through a third party API that ou team just has to integrate, not implement from scratch.

All in all, our software team only has the business application logic involved in sending notifications to authorities/assigned persons. This sort of application is well within our wheelhouse. We want to take a conservative view however, and just in case there are integration issues between our in-house application and the off-the-shelf components we will not assume the project will be organic.

Step 4: Using COCOMO for effort estimation and stuff

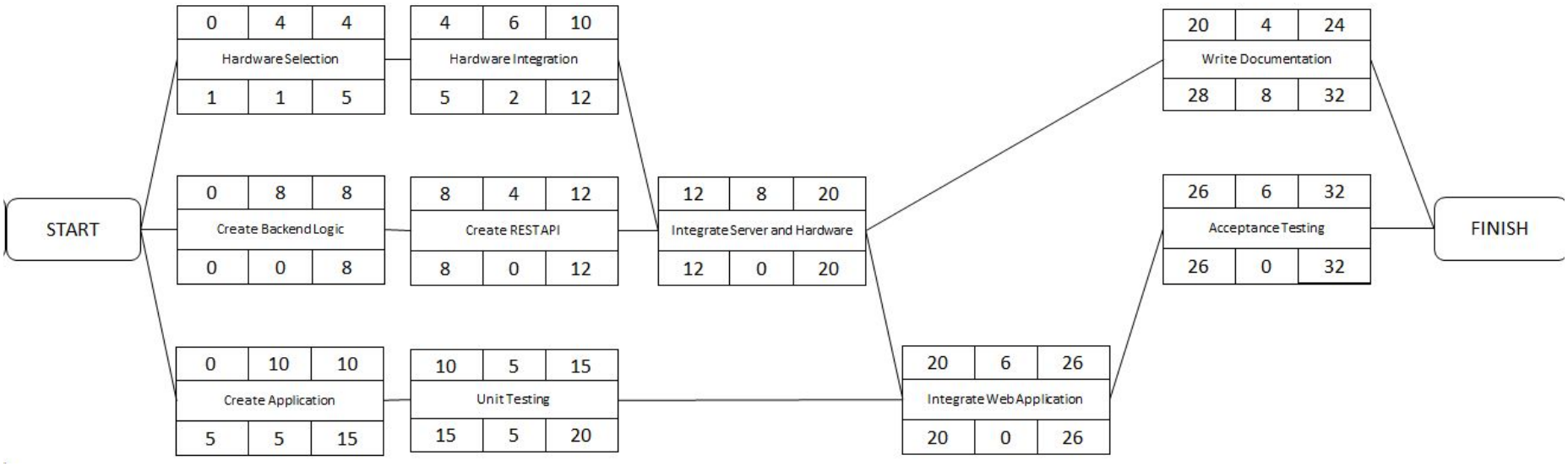
By using KLOC = 13.5, a = 3.0, b = 1.12 and EAF = 0.3845, our expected time is
E = 3.0*((7)^1.12) * (0.3845) = 21.281

By using a = 2.5 and b = 0.35:

D = 2.5*(21.28^0.35) = 7.29 months

Therefore, the project should take us 7.29 months and has an effort estimation of 21 work-months. However, an MVP with basic functionality can be released far sooner.

Activity Diagram for completion of the project:


START
0 4 4
Hardware Selection
1 1 5
4 6 10
Hardware Integration
5 2 12
0 8 8
Create Backend Logic
0 0 8
8 4 12
Create RESTAPI
8 0 12
0 10 10
Create Application
5 5 15
10 5 15
Unit Testing
15 5 20
12 8 20
Integrate Server and Hardware
12 0 20
20 6 26
Integrate Web Application
20 0 26
20 4 24
Write Documentation
28 8 32
26 6 32
Acceptance Testing
26 0 32
FINISH