



Faculty of Engineering and Applied Science
SOFE - 3490U Software Project Management
Winter 2023

Lab 3

Vishan Patel - 100784201
Akshat Kapoor - 100781511
Steven Mai - 100781485
Sabeh Khalid - 100754735

Monday, March 6th 2023
CRN 74669, Group 2

Calculating Effort

Classifying The Project

This project has been classified as an embedded system. This was decided as the project consists of motion sensors, web applications, security protocols, and databases. Furthermore, the software team lacks experience in this area and thus the complexity of the project increases. In order to successfully build this project, extensive research needs to be conducted on sensors and similar projects. Due to these reasons, this project was classified as an embedded system [1].

COCOMO Basic Model

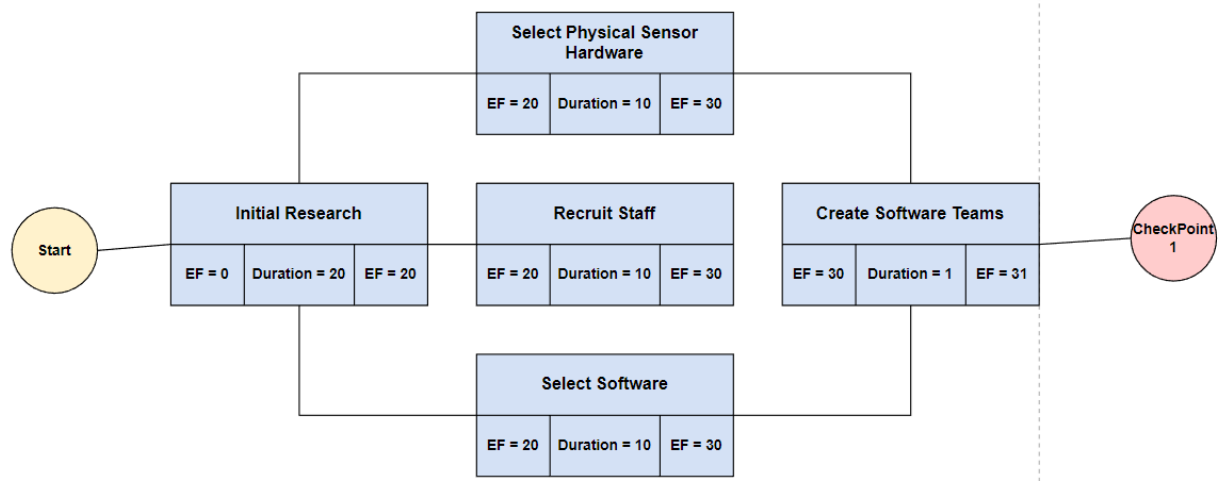
To calculate effort and time using COCOMO, variables a, b, c, and d were required. These values were obtained from the table below. The project is estimated to have 500 KLOC. This value originates from expert judgment with a software engineer familiar with this field [1].

	a	b	c	d
Organic	2.40	1.05	2.50	0.38
Semi-detached	3.00	1.12	2.50	0.35
Embedded	3.60	1.20	2.50	0.32

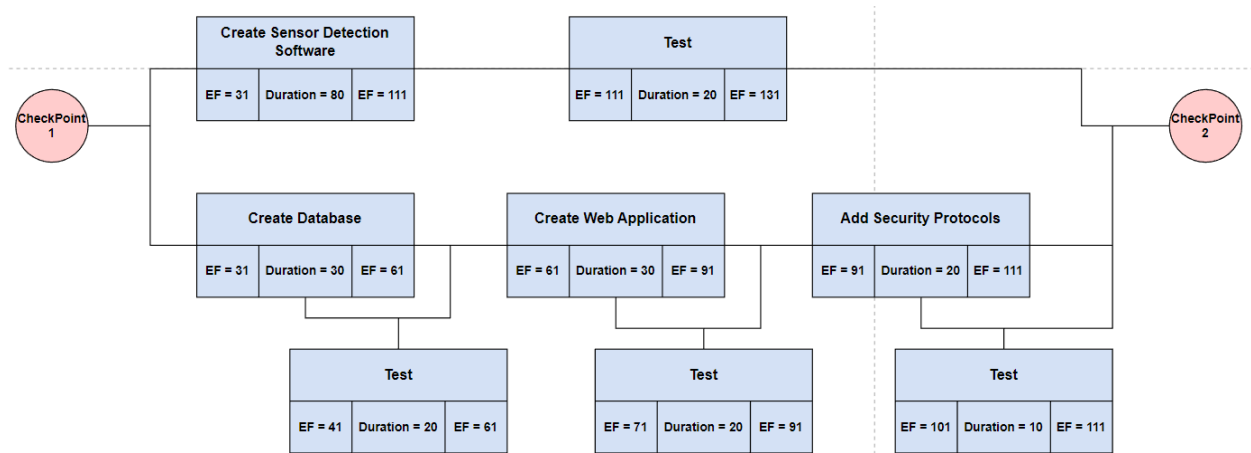
$Effort = a(KLOC)^b = 3.6(500)^{1.2}$	6238 person months
$Time = c(effort)^d = 2.5(6238)^{0.32}$	40.96 months
$PersonRequired = effort/time = 6238/40.96$	152 persons

Activity Diagram

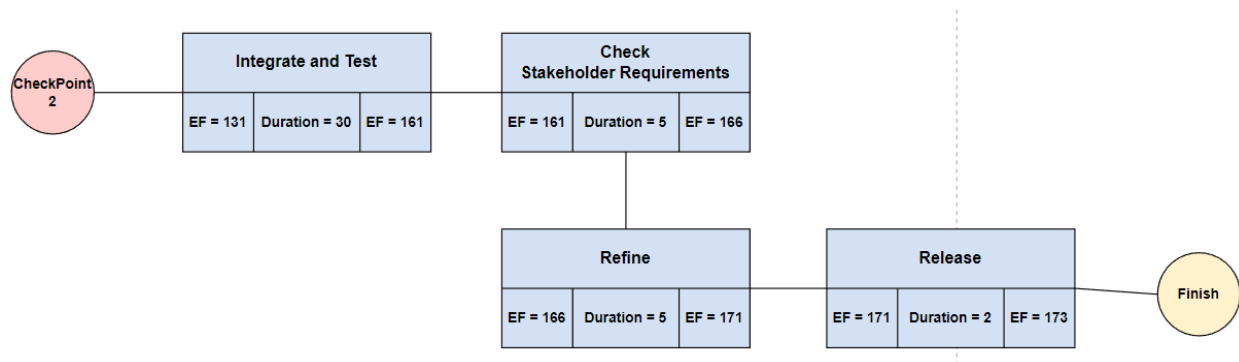
Planning Phase



Design Phase



Release Phase



Risk Management

As this is a complex project with strict requirements, there are many risks that may occur during the software design, development, and release phases.

Lack of Motivation

With this project taking approximately 41 months according to COCOMO, motivation is a probable issue. This will lead to low worker performance and delays. In order to combat this, incentives that recognize and celebrate good results will be introduced. Furthermore, a well-paced plan will be created to lower stress and each software team will be assigned a leader who is capable of motivating and managing the team [2].

Falling Behind Schedule

With a software development time of 41 months, development may fall behind schedule at any point leading to a huge time crunch for later activities. Weekly meetings will be conducted to confirm the progress. During the meetings, fairly strict deadlines will be set. If it is determined a software team is falling behind, resources will be allocated to ensure it stays on track. As long as the workers meet deadlines and communicate appropriately, the project will not fall behind [2].

Miscommunication

With this project requiring extensive research, there may be a lack of understanding leading to miscommunication. This may include failure to understand documentation. This problem will also be resolved during the previously mentioned meetings. There will be direct communication in person so that both parties understand the objectives of the project. Both parties will also go through the documentation together and figure out the details. As this is our first time dealing with sensors, there is also a very good chance that something does not work as expected [2].

Integration Issues

Along with not working as expected, there is a risk that modules may be missing. Similar to a conveyor belt system, our team will split up into multiple small teams when working on the project. Each small team is responsible for their own module. To ensure modules work together, the manager of each team will communicate with one another [3].

Budget Issues

Falling behind, miscommunication, and lack of motivation all require resources to solve whether it's time or money. When planning, risks will be taken into account and extra funds will be saved in case of an emergency. If a delay does occur, then funds will be available [2].

Functionality Issues

This project is built of multiple components; sensors, databases; security protocols, and a web application. With many components there are risks that the project does not function as required by the stakeholder requirements. To lower this risk, software testing will be conducted concurrently with the design stages. By consistently testing, it can be determined if the software has the required functionality.

References

- [1] Kalopsia. “Software Engineering: Cocomo Model”. GeeksforGeeks. Available: <https://www.geeksforgeeks.org/software-engineering-cocomo-model/>. (accessed Mar 6, 2023)
- [2] Team Asana, “7 Common Project Risks And How To Prevent Them [2023]”. Asana. Available: <https://asana.com/resources/project-risks>. (accessed Mar 6, 2023)
- [3] Simplilearn, “13 Common Project Risks And How To Tackle Them”. Simplilearn. Available: <https://www.simplilearn.com/common-project-risks-article>. (accessed Mar 6, 2023)