Software Project Management

SOFE 3490U

Lab Instructor: Sifatul Mostafi

Lab #3 - Project Management

March 5th, 2024

Group 8

Rodney Stanislaus (100822918)
Alexy Pichette (100822470)
Calvin Reveredo (100825740)

Introduction

To get started on our project, we'll use formal approaches like Function Point Analysis and the COCOMO model to estimate the amount of effort necessary to complete it. Using these approaches, we hope to estimate the time and resources required for effective project delivery. Following the estimating phase, we will create an activity diagram to outline the many phases involved in the development process. From the requirements gathering to the developmental phases such as coding and testing the system. Each activity's duration is estimated in weeks, with the first activity starting at week 0. Finally, we will examine a vital part of risk management: detecting prospective issues and creating preventative steps to alleviate them. Risks such as team productivity concerns or unexpected technological malfunctions during deployment will be extensively examined, and appropriate solutions will be proposed.

Activity 1: Calculating Estimated Effort

To estimate the effort for our present project, a school management system, we shall use the Constructive Cost Model (COCOMO). COCOMO needs us to determine the effort, time, and persons necessary based on both effort and duration. These three calculations use the following equations:

Estimated Effort: $E = a(KLOC)^b$

Duration: $D = c * E^d, where\ E\ is\ the\ Estimated\ Effort$

Person Needed: $P = E/D$

With these equations, we can determine our expected effort. We will take a semi-detached approach as our project is fairly simple, but our organization lacks much expertise in this area. As a result, it gets more complex. We will utilize the values a,b for effort and c,d for duration from the chart below in our previous equations.

Software Projects	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

The only thing needed before we can start estimating our effort is our code lines. We anticipate that 6,700 lines of code will be required to complete our project. With this, we can begin calculating the effort required for our project. As previously said, our variables and their values will be:

$a = 3.0$

$b = 1.12$

$c = 2.5$

$d = 0.35$

$KLOC = 6.7$

As previously noted, our computations will employ the variables listed above.

$$E = 3.0 * (6.7)^{1.12}$$
$$E = 25.25\, person - months$$

$$D = 2.5(25.25)^{0.35}$$
$$D = 7.74\, months$$

$$Person\ Needed = 25.25/7.74$$
$$Person\ Needed = 3.26$$

As a result, our project will require around three people to complete within an eight-month period.

Activity 2: Activity Diagram

Project Activities:

A. **Requirements Gathering**
   a. Description: Gather and document detailed requirements from stakeholders.
   b. Duration: 1 week

B. **System Design and Architecture**
   a. Description: Design the overall system architecture including frontend interfaces, backend services, and database schema.
   b. Duration: 3 weeks
   c. Dependencies: A - Requirements Gathering

C. **Design User Interface (UI)**
   a. Description: Create wireframes and layouts for the mobile application.
   b. Duration: 5 weeks
   c. Dependencies: B - System Design and Architecture

D. **Develop Frontend**
   a. Description: Implement mobile application interfaces for iOS and Android platforms by using chosen frameworks.
   b. Duration: 12 weeks
   c. Dependencies: C - Design User Interface (UI)

E. **Develop Backend**
   a. Description: Develop backend services and APIs for user authentication, profile management, class creation, and messaging.
   b. Duration: 15 weeks
   c. Dependencies: B -System Design and Architecture

F. **Integrate Frontend and Backend**
   a. Description: Combine frontend and backend components to create a functional mobile application.
   b. Duration: 6 weeks
   c. Dependencies: D - Develop Frontend, E - Develop Backend

G. Perform Testing

a. Description: Conduct unit testing and user acceptance testing (UAT) to ensure quality.
b. Duration: 4 weeks
c. Dependencies: F - Integrate Frontend and Backend

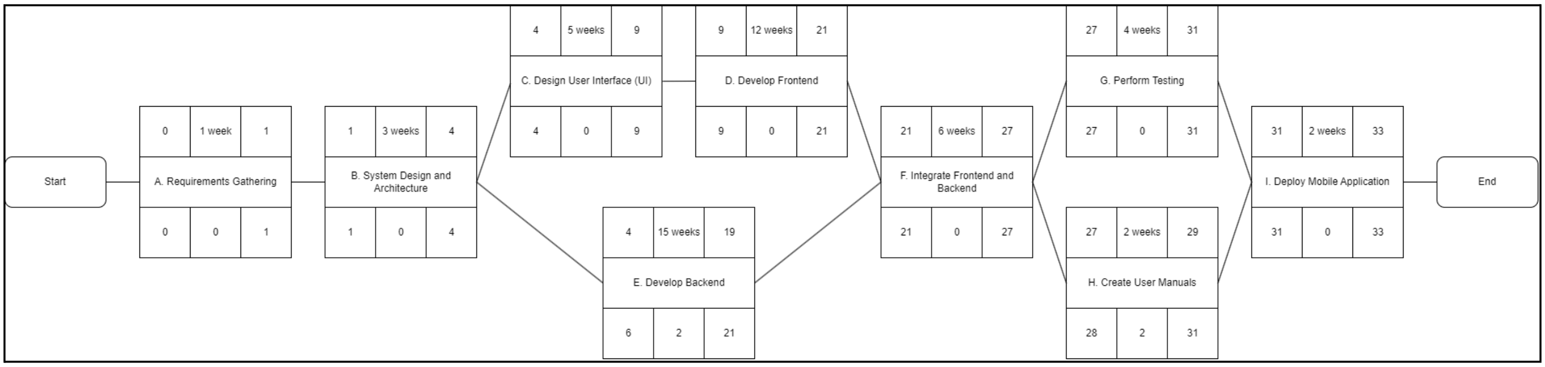
H. Create User Manuals

a. Description: Develop user documentation and guides to help users navigate and understand the application.
b. Duration: 2 weeks
c. Dependencies: F - Integrate Frontend and Backend (Best to start the user manual creation process after the majority of the implementation tasks are completed)

I. Deploy Mobile Application

a. Description: Deploy the mobile application to the app stores.
b. Duration: 2 weeks
c. Dependencies: G - Perform Testing, H - Create User Manuals

Activity Diagram:



Activity 3: Identifying Risks

During the creation and deployment of the school management system, there are several risks that are present within the process of each, that can lead to many challenges arising. It is important to have a mitigation plan that can be used in order to reduce the likelihood of these risks occurring.

- Stakeholders may not be satisfied with the way the school management system looks and operates.
    - ***Mitigation Plan***: Ensure regular meetings and consistent communication lines are established between the stakeholders, to ensure they are seeing the product before it is brought to completion, and are able to voice any input or concerns they have. That way, updates can be made with a lower cost compared to if it was later in the development process
- Hired team may have no issue with implementing functionality of the school management system, but may struggle in the area of user interface design, and may not have sufficient knowledge of what constitutes a good and appealing user interface.
    - ***Mitigation Plan***: There must be a strong and thorough hiring process, where potential candidates have demonstrated a tremendous understanding of effective user interface design, from past projects and experiences.
- Students, alumni, professors, and staff are still leaning towards other methods of communication that are not the school management system's messaging feature, in order to communicate with one another (i.e. email, social media apps, etc).
    - ***Mitigation Plan***: The messaging system must be made to stick out, compared to other forms of communication. A variety of features can be implemented in order to do this, such as…
        - the ability to attach different file types of varying sizes
        - have your message automatically typed out based on what you audibly say
        - send a message as audio itself
        - ability for large group chat creation, enabling communication between a lot of people at once
- Those that are using the school management system are content with its features and the way it operates, but the number of initial users of the system is tremendously lower compared to what was expected.
    - ***Mitigation Plan***: An effective advertising campaign should be created that lets everyone know about the system, and makes it look as appealing as possible. It is important to use a variety of advertising channels to ensure it is seen by everyone, such as…
        - Social media posts and emails
        - Posters on the school campus
        - Website
- The speed of the operations of the school management system is not as quick as was expected (i.e. the output from an alumni search takes longer than expected to be displayed)

    - ***Mitigation Plan***: Importance must be placed on the cloud servers that are chosen to be used for the system. It should be servers that are able to support the system’s needs by a large amount, and not simply servers that can just barely support it.
- Hackers attempt to break into the school management system, in order to gain information of students, alumni, professors, and staff.
    - ***Mitigation Plan***: Ensure the data of users is securely encrypted, and effective alerts are put in place to provide warning if any possible signs of a security attack being attempted is detected.
- Development of the school management system starts to fall behind schedule.
    - ***Mitigation Plan***: Ensure an appropriate amount of time is set for each task, including accounting for the latest possible date that it may be completed.