United International University

Department of Computer Science and Engineering

Course Name: Project Management

Course Code: CSE 469/PMG 4101

Section: A

Project Name: Student Room - Online Learning Platform

Group Number: 4

Group Members:

Sumaiya Akhter - 011201069

K. M Shahriar Alam Adib - 011201073

Sanjida Jannat Anannaya - 011201440

Aura Shattika Ali - 011182143

Submitted to:

Dr. Suman Ahmed

Assistant Professor of CSE Department

at United International University

Index

Chapter

Link: https://github.com/anannayajannat/pm?search=1

1. Introduction

1.1 Project and Project Management

A project is a purposeful venture with a defined beginning and end, undertaken to achieve specific goals while sticking to constraints such as time, budget, and resources. Effective project management is the art of organizing these ventures. It involves meticulous planning, organized execution, and continuous monitoring to ensure that objectives are met efficiently and with the highest quality. Project management navigates the complexities of tasks, team dynamics, and unforeseen challenges, steering the project towards success while optimizing the utilization of resources and maintaining clear communication among stakeholders.

1.2 Vision

The Vision Statement is the initial part of the documentation that communicates the overall goal of the project, what it aims to achieve, and its benefits. It describes the desired outcome and sets the direction for the project. It is usually a high level statement that conveys the essence of the project.

The vision of the "Student Room" project is to create a comprehensive learning platform that enhances the academic experience of students at UIU. The platform aims to provide a seamless and collaborative environment where students can access and share study materials, engage in meaningful discussions, participate in mock tests, utilize simulations and learning resources, and stay updated with the latest research papers. By leveraging technology, the "Student Room" strives to empower students to excel in their academic pursuits, foster a culture of continuous learning, and cultivate a vibrant educational community.

1.3 Project Overview

Our project focuses on the development and implementation of an innovative university-based learning platform. This platform aims to revolutionize the academic experience by providing comprehensive tools and resources tailored to make students' lives easier.. Key features of the platform include the sharing of study materials, a discussion forum, interactive mock tests and simulations for practical understanding, and easy access to learning resources.

By creating “Student Room”, our project seeks to empower students with the tools they need to excel in their academic pursuits and thrive in a collaborative learning community.

1.4 Scope

The Scope Statement defines the boundaries of the project, outlining the deliverables and objectives It describes what will be included and excluded from the project, as well as the specific requirements, timelines, and resources needed to achieve the goals. The Scope Statement helps to ensure that everyone involved in the project understands the specific objectives and what is required to achieve them.

The "Student Room" project will encompass the following key features and functionalities:

Study Material Sharing:

- Students can upload and share study materials, including questions, notes, and relevant resources.
- Categorization and organization of study materials for easy accessibility.
- Version control and moderation mechanisms to ensure the quality and integrity of shared content.

Discussion Forum:

- Students can participate in topic-based discussion forums to engage in academic conversations.
- Threaded discussions, replies, and notifications to facilitate meaningful interactions.
- Moderation features to maintain a respectful and conducive environment for discussions.

Mock Tests:

- Platform provides a repository of mock tests covering various subjects and topics.
- Timed assessments with automatic grading and feedback.
- Performance analytics and progress tracking for students to evaluate their strengths and areas for improvement.

2. Project Management Methodologies

2.1 Introduction to Project Management Methodology

Project management methodology is a set of principles and practices that guide in organizing the projects to ensure the optimum performance. It’s a framework that helps to manage a project in the best way possible.

There are several types of methodologies and there are lots of factors that will impact which project management methodology is right for a project, team, and organization. According to our budget, team size, ability to take risk, timeline, etc, we can select one of the methodology for project management which are given below:

2.2 Incremental Method

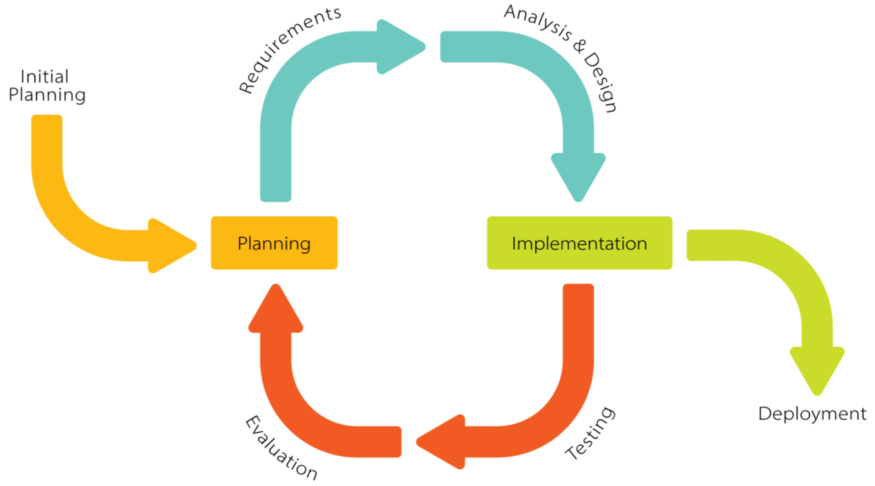


Fig: Incremental Model

Incremental Model, also known as the Iterative Method, is a process of software development where requirements are divided into multiple standalone modules of the software development cycle. In this model, each module goes through the requirements, design, implementation and testing phases. Every subsequent release of the module adds function to the previous release. The process continues until the complete system is achieved.

This can be achieved in the following steps:

1. Initial Planning: The overall goals, requirements, and priorities of the project are defined.
2. Requirements
3. Analysis & design
4. Implementation
5. Testing
6. Deployment

Incremental Method is best used when:

- Software with evolving requirements.
- Complex projects with high risks.
- Time-Critical projects.
- Projects with a need for continuous improvement.

2.3 Boehm’s Spiral Methodology

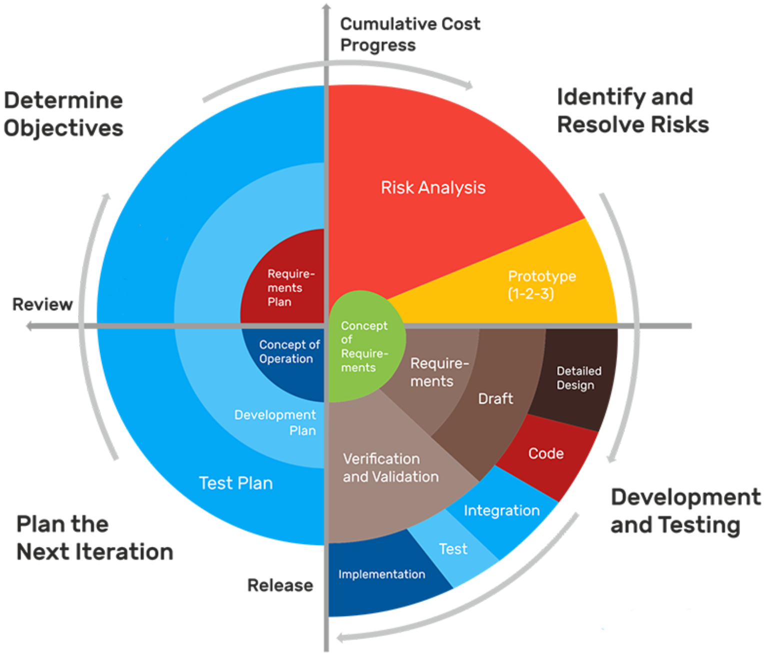


Fig: Boehm’s Spiral Methodology

Boehm's Spiral Model is an iterative software development methodology that combines elements of both waterfall and iterative models. It is based on the idea of managing risks and accommodating changes throughout the software development process. The Spiral Model emphasizes a systematic and controlled approach to building software systems.

The key concept of Boehm's Spiral Model is the iterative nature of the process. It consists of a series of iterations, or spirals, which represent different phases of the software development life cycle. Each spiral involves four main activities: planning, risk analysis, engineering, and evaluation.

This methodology works in four steps:

1. Determine Objectives: defines the specific objectives and requirements for the iteration.
2. Identify and Resolve Risks: potential risks and uncertainties are identified and analyzed.
3. Development and Tests: designed, developed, and tested according to the objectives and requirements
4. Determine the next cycle: based on the evaluation results and risk analysis, the next iteration objectives, requirements, and plans are defined.

Spiral Method is best used when:

- Frequent Updates or Changes Required
- Large and Complex Projects
- When Project Requirements are unclear and ambiguous

2.4 RUP (Rational Unified Process)

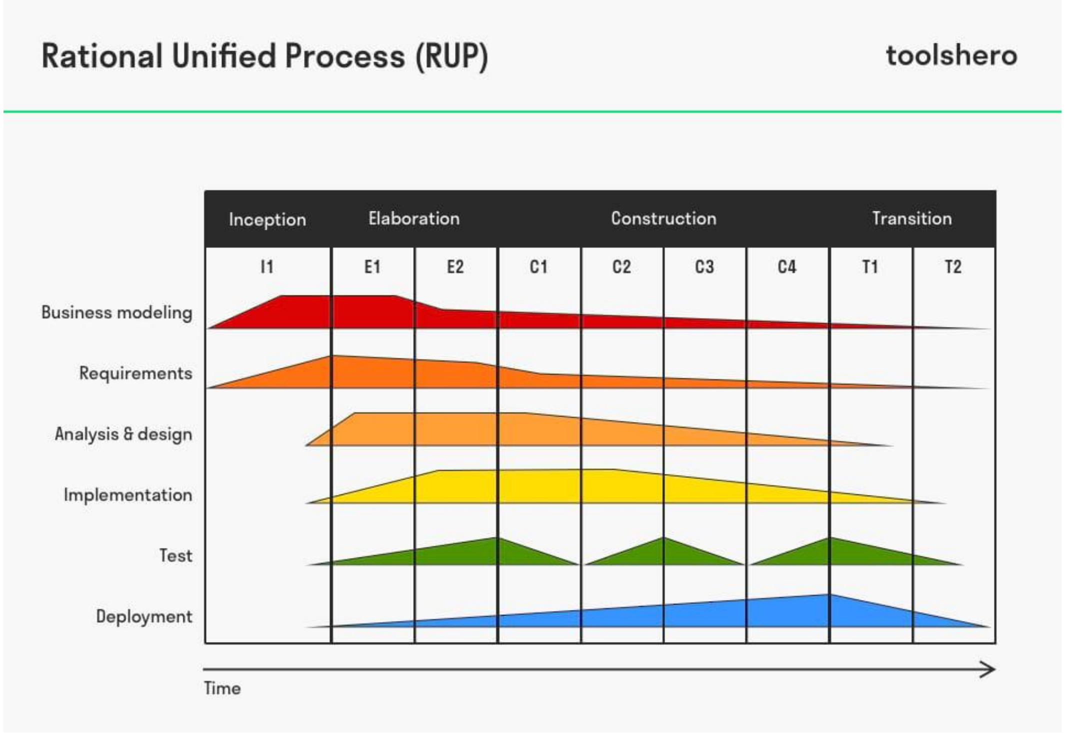


Fig: Rational Unified Process

The Rational Unified Process (RUP) is a structured software development methodology that emphasizes collaboration, iterative development, and continuous feedback. It divides the process into phases and provides guidelines for requirements, design, implementation, testing, and deployment. RUP promotes effective communication, risk management, and customization to fit project needs. It's a flexible framework for developing high-quality software systems.

There are 5 phases of RUP:

1. **Inception Phase**
    a. Communication and planning
    b. Identifies the scope of the project
    c. Project plan,goal,use case model and description
    d. Prototype development
2. **Elaboration Phase**
    a. Planning and modeling
    b. Detailed evaluation and development plan and diminishes the risks
    c. Executable architecture baseline

3. **Construction Phase**
    a. Project development and completion
    b. System or source code create and complete testing

4. **Transition Phase**
    a. Final project released and analysis
    b. Beta testing
    c. Collection of user feedback

5. **Production Phase**
    a. Project maintenance and update accordingly
    b. Final phase of the model
    c. User help and assistance platform availability

RUP is best used when:

- Projects with evolving requirements.
- Projects involving distributed teams.
- Projects with critical or high-risk components.

2.5 Scrum

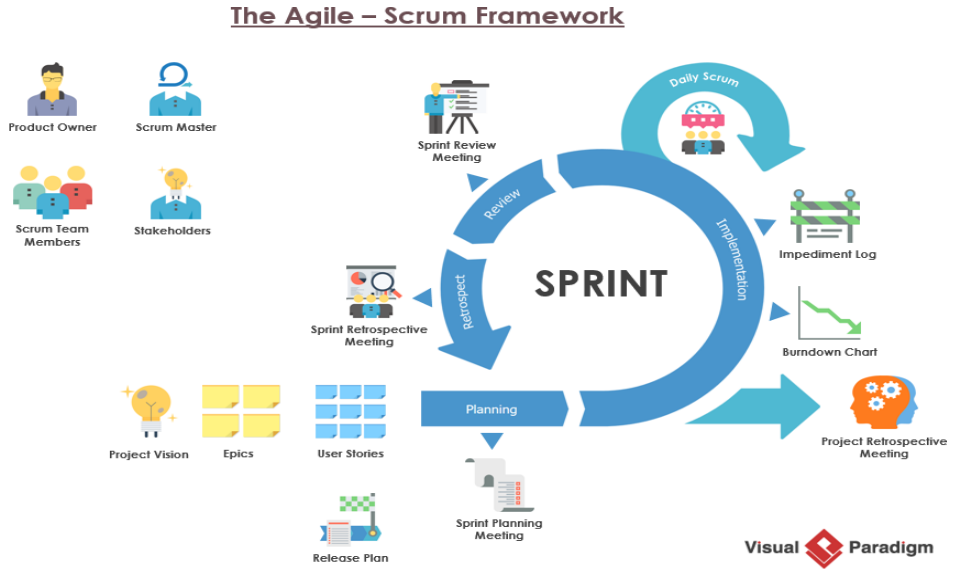


Fig: Scrum

Scrum is an agile project management framework for developing complex products. It is based on iterative and incremental practices, emphasizing flexibility, collaboration, and continuous improvement. Scrum is widely used in software development but can be applied to various industries.

To describe Scrum in the simplest terms, we can break it down into three parts:

Roles:

1. **Scrum Master:** Facilitates the Scrum process and helps the team stay on track.
2. **Product Owner**: Represents the stakeholders and defines product requirements.
3. **Development Team:** Cross-functional team responsible for delivering the product.

Artifacts:

1. **Product Backlog:** A prioritized list of all desired features and requirements.
2. **Sprint Backlog:** Subset of the Product Backlog containing items selected for the current iteration.
3. **Increment:** The sum of completed product backlog items at the end of a sprint.

Events:

1. **Sprint:** Time-boxed iteration (typically 1-4 weeks) where work is completed.
2. **Sprint Planning:** Collaborative session to determine what can be delivered in the upcoming sprint.
3. **Daily Scrum:** Daily stand-up meeting for the team to synchronize activities and discuss progress.

4. **Sprint Review:** Meeting at the end of a sprint to review the increment and gather feedback.

Scrum is used when:

- The project is complex (changing requirements)
- Requires collaboration
- Project needs adaptability
- Reduced risk (Iterative and incremental approach reduces the risk of project failure.)

2.6 Extreme Programming (XP)

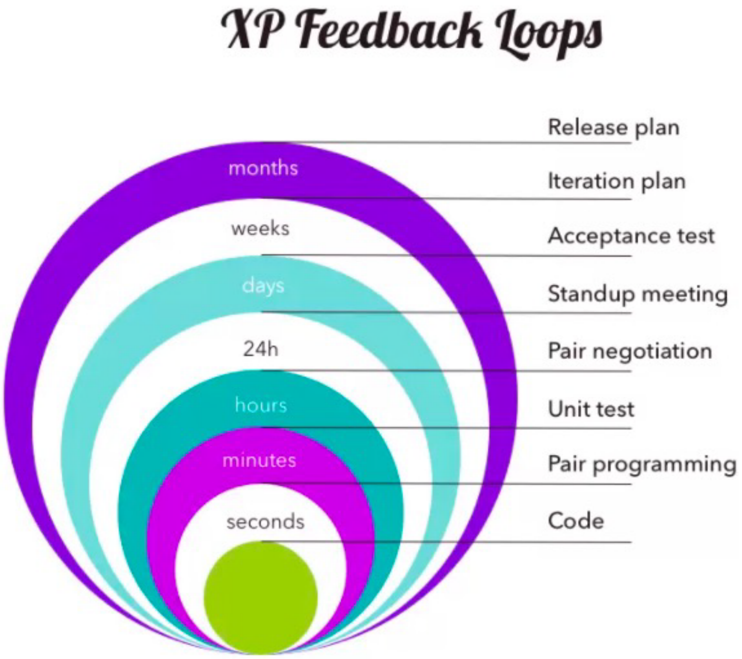


Fig: Extreme Programming Feedback

Extreme Programming (XP) is an agile software development methodology that aims to improve the quality and responsiveness of software development projects. It emphasizes collaboration, customer involvement, and iterative development. XP incorporates several practices and principles to enable teams to deliver high-quality software.

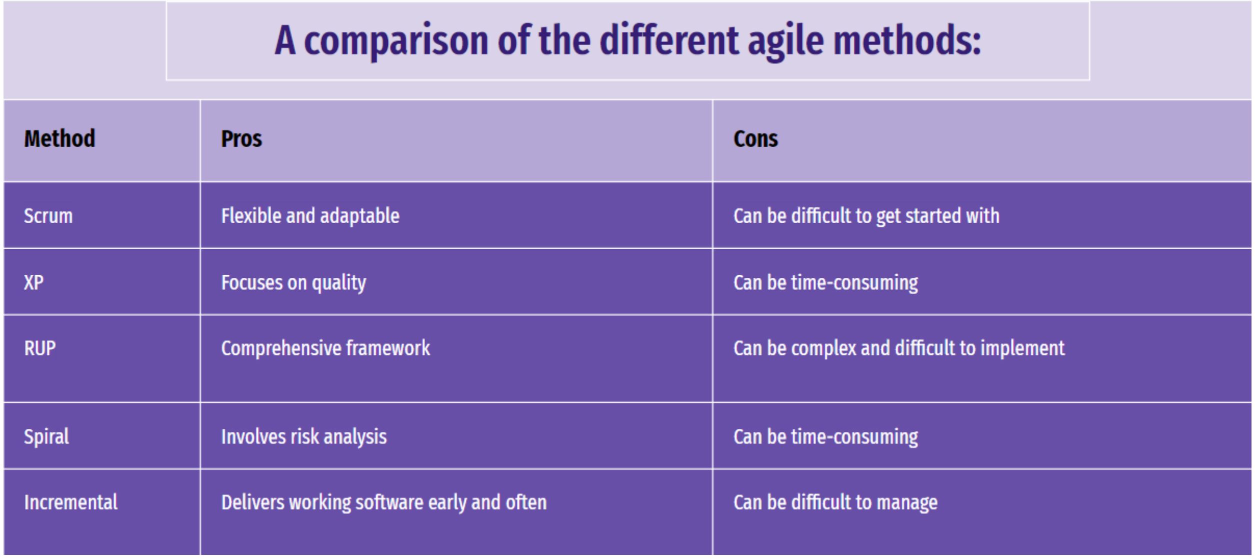
As we can see from the diagram provided, the core practices of XP are as follows:

1. Release Plan
2. Iteration Plan
3. Acceptance Test
4. Standup Meeting
5. Pair Negotiation
6. Unit Test
7. Pair Programming
8. Code

Extreme Programming is best utilized when:

- Project is complex and risky
- Require high quality.
- Have changing requirements
- Need to be delivered quickly

Here is a comparison table for all of the methodologies we have discussed:



A comparison of the different agile methods:

Method	Pros	Cons
Scrum	Flexible and adaptable	Can be difficult to get started with
XP	Focuses on quality	Can be time-consuming
RUP	Comprehensive framework	Can be complex and difficult to implement
Spiral	Involves risk analysis	Can be time-consuming
Incremental	Delivers working software early and often	Can be difficult to manage

For our project, an Online Learning Platform, based on UIU, we will be using Scrum Methodology. Since it is the most flexible and adaptable, it is the most suitable for our project.

2.7 Used Methodology for our project

Since our project is fairly simple and straightforward, we have decided to use Scrum methodology. Scrum is flexible and adaptable, so it will be suitable for the project we are trying to create.

3. System Documentation

3.1 Use Case

3.1.1 Use Case for Our Project

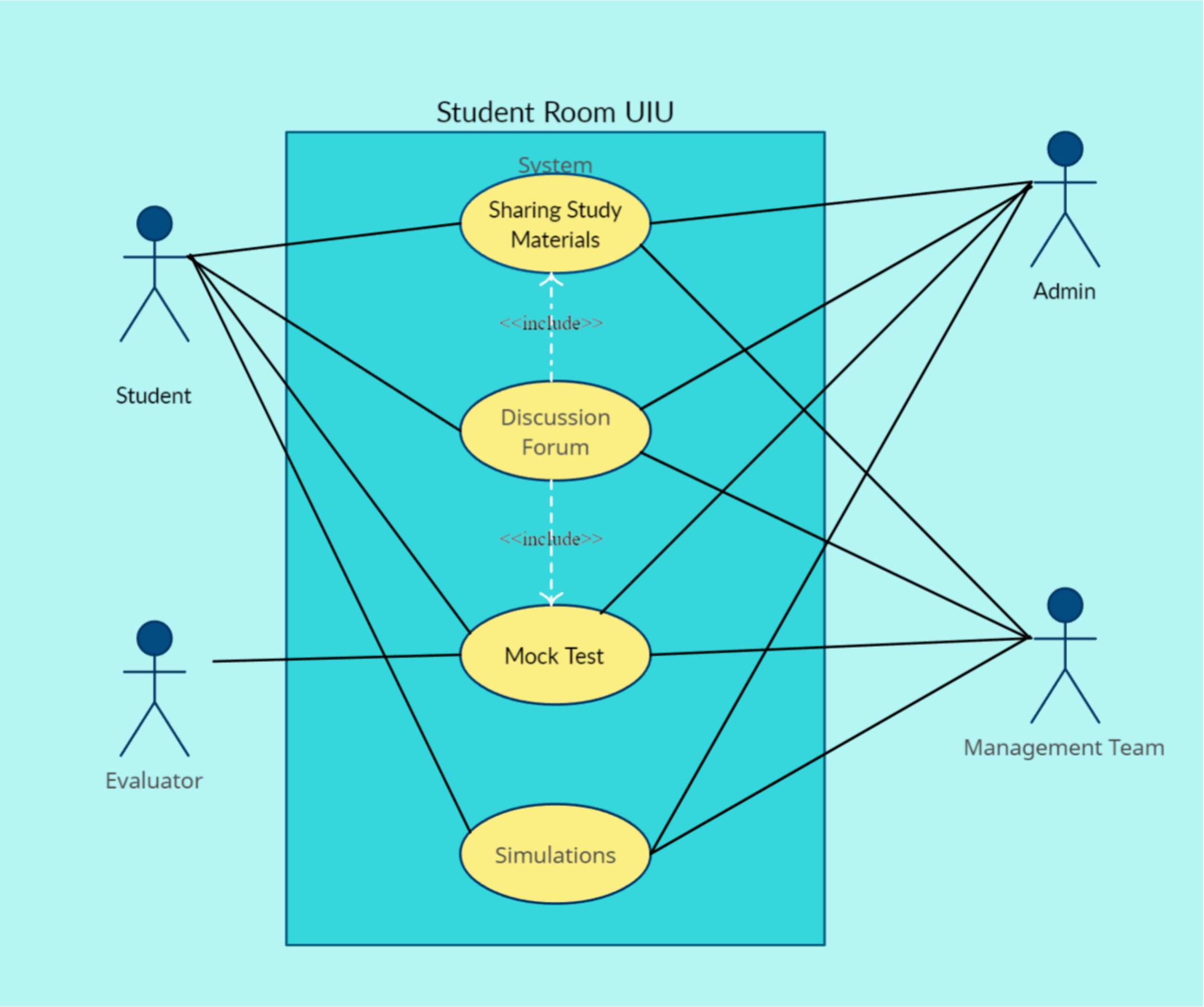


Fig: Use-case for Student Room

3.1.2 Descriptive form of Use Case

Name	**UC-1: Sharing Study Materials**
Summary	Outlines the process of how students share study materials with other students, collaborative learning and resource sharing.
Rationale	Sharing study materials enhances students' understanding of subjects and provides a diverse range of study resources. By enabling students to share study materials promotes knowledge exchange and supports a sense of community among the UIU students.
Users	Student, Admin, Management Team
Preconditions	Users must be logged in to their account.
Basic Course of Events	1. User logs into the "Student Room". 2. Students access the shared materials. 3. If the materials are available, students download the materials.
Alternative Paths	1. If the student doesn't have an account, they are prompted to create one before proceeding. 2. Students can request study materials by filling out a form. This allows the materials to be added to the database, enabling other students to access them as well.
Postconditions	● The shared study materials are available for other students to view, download, and interact with. ● The student who shared the materials receives feedback and comments from other users. ● The platform's study materials database is enriched with new resources.

Name	**UC-2: Taking Mock Tests**
Summary	Students can take mock tests simulating real exam conditions to assess their knowledge and readiness.
Rationale	Mock tests provide students with a realistic evaluation of their understanding, time management skills, and exam readiness. It helps students identify strengths and areas needing improvement before actual exams.
Users	Students, Management Team, Evaluator
Preconditions	The student must be registered and approved by administration panel to have access to the "Mock Tests" section of the platform
Basic Course of Events	1. A student selects a specific mock test from the available options. 2. The student begins the mock test, adhering to the allocated time limit and it is automatically graded after the time finishes.
Alternative Paths	1. This section displays a list of available mock tests, categorized by subjects or exam types. 2. It presents an overview of a mock test, including details such as the number of questions, time limit, and specific topics covered. 3. The student begins the mock test by clicking the "Start Test" button. 4. If the student completes the mock test before the time limit, they can review and revise their answers or they submit their answers after the time limit and they are provided with feedback on correct and incorrect answers, and highlights areas needing improvement.
Postconditions	● The student receives an assessment of their performance in the mock test, indicating strengths and weaknesses. ● The mock test results are saved in the student's account for future reference and comparison.

3.2 Functional Requirements

3.2.1 Introduction to Functional Requirements:

Functional requirements are product features or functions that developers must implement to enable users to accomplish their tasks

Our Projects Functional Requirements

FR-1: Sharing study materials
FR-2: Discussion Forum
FR-3: Mock Tests
FR-4: Simulations and Learning Resources
FR-5: Updates on Research Papers
FR-6: User Profiles and Personalization
FR-7: Notification System
FR-8: User Feedback and Reporting

3.2.2 Functional Requirements for our Project

Name	**FR-1: Sharing Study Materials**
Summary	Enable students to upload and share study materials, such as questions, notes, and resources.
Rationale	Promotes collaborative learning and resource sharing among students, enhancing their understanding and knowledge base.
Requirements	● Students can upload and categorize study materials. ● Students can comment on and rate shared materials. ● Search and filter options for easy access to shared resources.
References	UC-1

Name	**FR-2: Mock Tests**
Summary	Offer simulated exam experiences through mock tests to assess student readiness.
Rationale	Helps students evaluate their knowledge, practice time management, and identify areas for improvement.
Requirements	● Library of mock tests for various subjects and exams. ● Timed tests with automatic submission and result display. ● Detailed result analysis with correct/incorrect answers and explanations.
References	UC: 1, UC: 2, FR: 2

3.3 Non-Functional Requirements

3.3.1 Introduction to Non-Functional Requirement:

Non-functional requirements or NFRs are a set of specifications that describe the system's operation capabilities and constraints and attempt to improve its functionality.

Our projects Non-functional Requirements:

NF-1: Performance
NF-2: Security and Privacy
NF-3: Accessibility
NF-4: Scalability

3.3.2 Non-Functional Requirements for our Project

Name	**NF-1: Performance**
Summary	Ensure that the "Student Room" delivers consistent and acceptable performance to users under varying loads and usage patterns.
Rationale	Performance is crucial to maintaining user satisfaction and engagement, especially during peak usage times such as exams or project deadlines.
Requirements	● The platform should load pages and resources within 3 seconds on average. ● The system should handle concurrent user interactions without significant delays or slowdowns. ● Performance testing should be conducted to simulate different usage scenarios and ensure responsiveness.

Name	**NF-2: Security and Privacy**
Summary	Implement robust security measures to safeguard user data and maintain the privacy of sensitive information.
Rationale	Protecting user data and maintaining privacy instills trust and confidence in users to engage with the platform.
Requirements	● All user data, including personal information and study materials, must be encrypted during transmission and storage. ● User authentication and authorization mechanisms should prevent unauthorized access. ● Compliance with relevant data protection regulations and standards, such as GDPR or HIPAA, should be ensured.

4. Estimation and Scheduling

4.1 Work Breakdown Structure

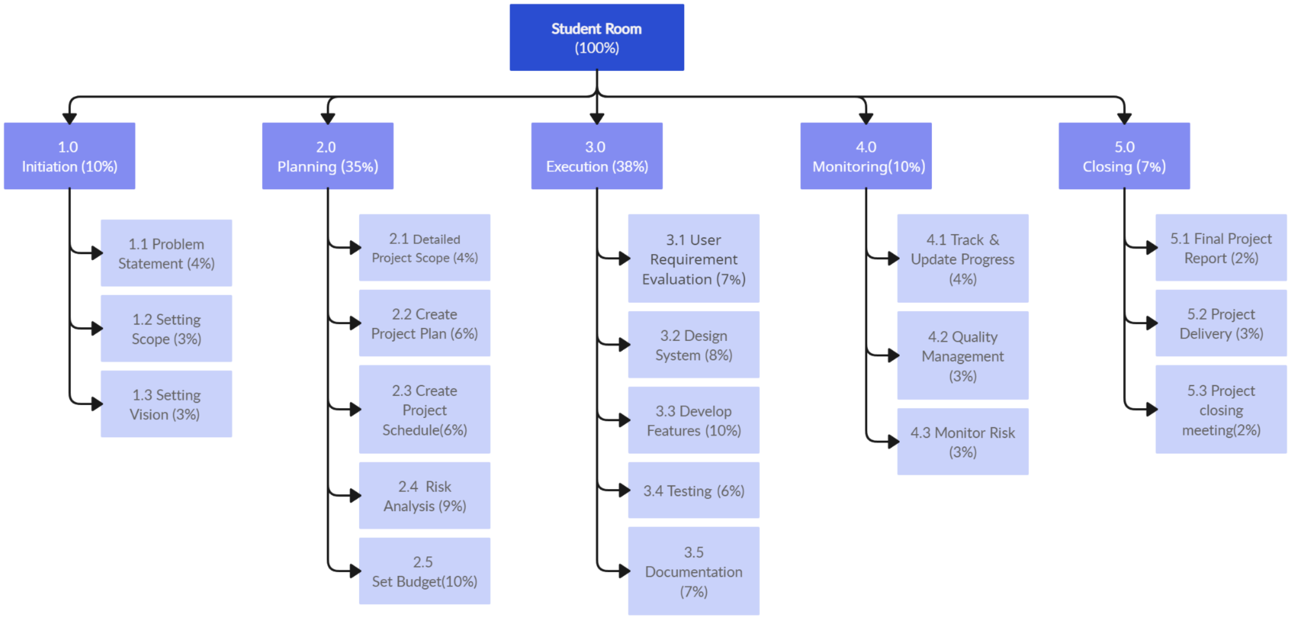
A project management tool that takes a step-by-step approach to complete large projects with several moving pieces. It is a visual representation of the project's scope, tasks, and deliverables. The purpose of a WBS is to organize and define the work required to complete a project and provide a framework for planning, scheduling, and controlling project activities.

Principles of WBS:

1. Hierarchy: Every child task has a relationship with the parent task.
   Adding all child tasks shows a clear picture of the parent.
2. 100% Rule: Every activity level must make up 100% of the parent level, and it must have at least two “child” elements.
3. Mutually-Exclusive Elements: Enforcing mutual exclusivity helps cut down miscommunication and avoid duplicate work.
4. Outcome Oriented: Focusing more on the outcomes rather than the actions.
5. Three Levels, Level 2 Most Important: If a WBS has about three levels, the scope of a project and the level of detail are about right. Level-2 determines how actual costs and schedule data are grouped for future project cost and schedule estimating which helps project managers to know how much it took to design (major work element) a product after it had been completed
6. Make Assignments: Every work package should be assigned to a
   specific team or individual.
7. 8/80 Rule: Less than eight hours of effort, but no more than 80. No more than ten days (which is the same as 80 hours if you work full time). A work package should take no more than a month to complete.

There are two approaches for WBS:

1. Top-down: Start with the largest items of the project and keep breaking them down
2. Bottom-up: Start with the detailed tasks and roll them up.


Student Room (100%)
1.0 Initiation (10%)
1.1 Problem Statement (4%)
1.2 Setting Scope (3%)
1.3 Setting Vision (3%)
2.0 Planning (35%)
2.1 Detailed Project Scope (4%)
2.2 Create Project Plan (6%)
2.3 Create Project Schedule(6%)
2.4 Risk Analysis (9%)
2.5 Set Budget(10%)
3.0 Execution (38%)
3.1 User Requirement Evaluation (7%)
3.2 Design System (8%)
3.3 Develop Features (10%)
3.4 Testing (6%)
3.5 Documentation (7%)
4.0 Monitoring(10%)
4.1 Track & Update Progress (4%)
4.2 Quality Management (3%)
4.3 Monitor Risk (3%)
5.0 Closing (7%)
5.1 Final Project Report (2%)
5.2 Project Delivery (3%)
5.3 Project closing meeting(2%)

WBS of Student Room

4.2. Delphi Wideband Method

Delphi WideBand Method is a structured method of problem-solving and decision-making that involves a group of experts providing feedback on a particular issue.
The goal is to arrive at a consensus on the best course of action based on the collective knowledge and expertise of the group.

Processes of Delphi Wideband Method -

- Choosing the team
- Kick-off meeting
- Individual preparation
- Estimation session
- Assembling TaskReview Result

Processes of Delphi Wideband Method for Student Room -

Choosing the team -

The project manager selects a panel of experts and a moderator as well as defines the scope of the project. For the student room project estimation, our estimation team consists of 4 members (Aura Shattika Ali, Sanjida Jannat Anannaya, Sumaiya Akhter, K.M Shahriar Alam Adib) and a moderator.

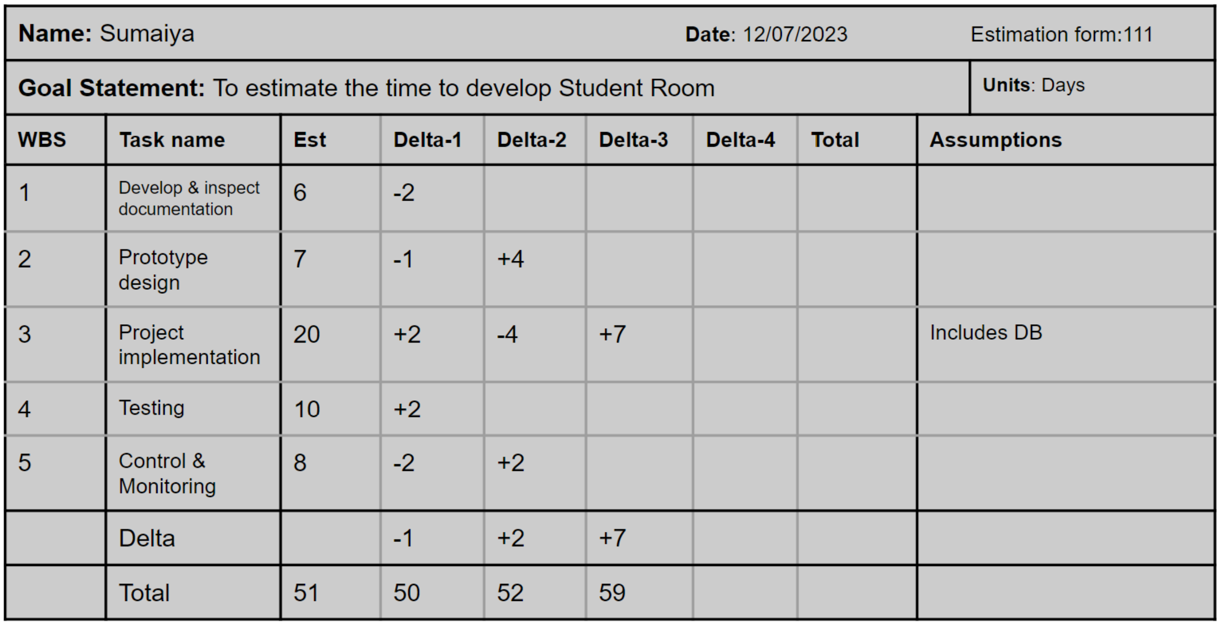
Kick off meeting -

Moderator will explain the delphi process. The estimators read the vision and scope through documentations and brainstorm the assumptions. Then the team will generate the initial WBS and decide on the units of estimations.

Individual preparation -

Initially estimated time for each task of the WBS of each member. Our team assumptions and tasks are Develop & inspect documentation, Prototype design, Project implementation, Testing, Control & Monitoring.

Estimation session -

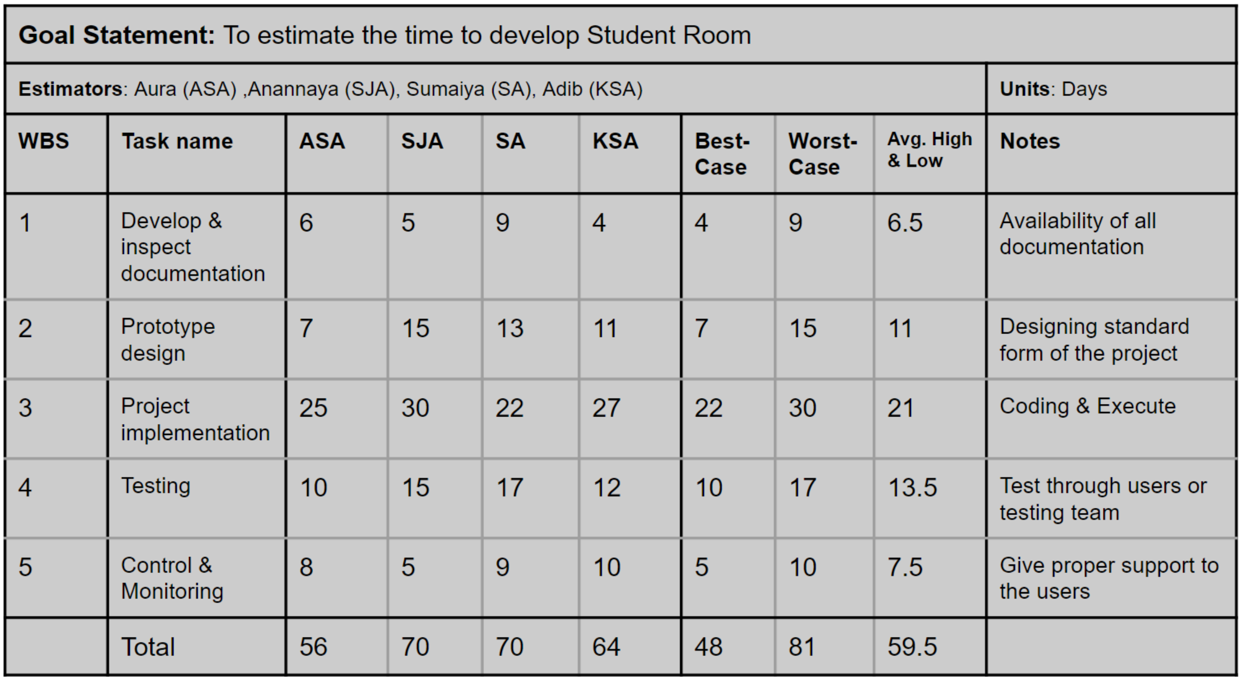


Name: Sumaiya **Date**: 12/07/2023 Estimation form:111

Goal Statement: To estimate the time to develop Student Room **Units**: Days

WBS	**Task name**	**Est**	**Delta-1**	**Delta-2**	**Delta-3**	**Delta-4**	**Total**	**Assumptions**
1	Develop & inspect documentation	6	-2					
2	Prototype design	7	-1	+4				
3	Project implementation	20	+2	-4	+7			Includes DB
4	Testing	10	+2					
5	Control & Monitoring	8	-2	+2				
	Delta		-1	+2	+7			
	Total	51	50	52	59			

Assembling Tasks -



Goal Statement: To estimate the time to develop Student Room

Estimators: Aura (ASA) ,Anannaya (SJA), Sumaiya (SA), Adib (KSA) | **Units**: Days

WBS	Task name	ASA	SJA	SA	KSA	Best-Case	Worst-Case	Avg. High & Low	Notes
1	Develop & inspect documentation	6	5	9	4	4	9	6.5	Availability of all documentation
2	Prototype design	7	15	13	11	7	15	11	Designing standard form of the project
3	Project implementation	25	30	22	27	22	30	21	Coding & Execute
4	Testing	10	15	17	12	10	17	13.5	Test through users or testing team
5	Control & Monitoring	8	5	9	10	5	10	7.5	Give proper support to the users
	Total	56	70	70	64	48	81	59.5	

Reviewing results -

After assembling all the tasks, the project manager decides to final estimate 60 days. The project manager will discuss and review the final task list, estimations and assumptions with the estimation team and moderator.

4.3. RET, DET, FP Estimations

Internal Logical Files (ILF) - Internal Logical File (ILF) is a user identifiable group of logically related data or control information that resides entirely within the application boundary. The primary intent of an ILF is to hold data maintained through one or more elementary processes of the application being counted.

Data Element Type (DET) – Data Element Type is a unique user recognizable, non-repetitive field.

Record Element Type (RET) – A Record Element Type (RET) is the largest user identifiable subgroup of elements within an ILF. It is best to look at logical groupings of data to help identify them.

RETs	**DETs**		
	1-19	20-50	51+
1	Low	Low	Average
2-5	Low	Average	High
6 or more	Average	High	High

Complexity	Points
Low	7
Average	10
High	15

Functional Point Estimation

Internal Logical File (ILF)	**RETs**	**DETs**	**Complexity**	**Function Points**
Student	1	5	Low	7
Admin	1	3	Low	7
Question Bank	2	8	Low	7
Discussion Forum	3	10	Low	7
Research Papers	1	4	Low	7
Total				**35**

4.4. COCOMO Model

The key outcomes of COCOMO Model are:

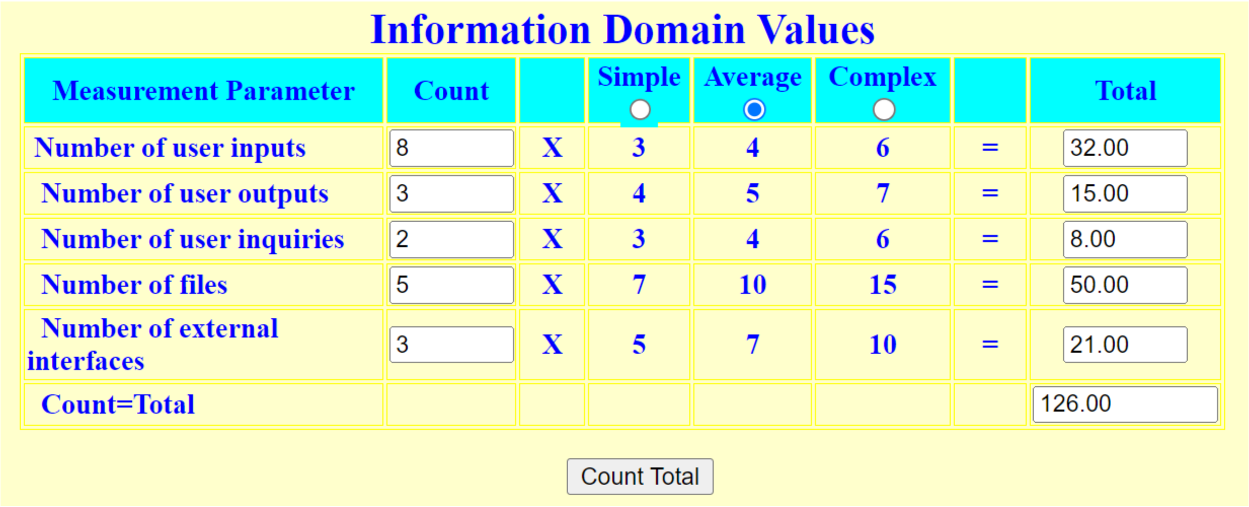
1. **Effort:** Amount of labor that will be required to complete a task.
2. **Schedule:** The amount of time required for the completion of the job

Types of COCOMO Model

There are three types of COCOMO model to predict the cost estimation at different levels, based on the amount of accuracy and correctness required.

1. **Organic** – The team size required is adequately small, the problem is well understood and has been solved in the past and also the team members have a nominal experience regarding the problem.
2. **Semi-detached** – The vital characteristics such as team size, experience, and knowledge of the various programming environments are comparatively less familiar and difficult to develop compared to the organic ones and require more experience and better guidance and creativity. Eg: Compilers or different Embedded Systems can be considered Semi-Detached types.
3. **Embedded** – A software project requiring the highest level of complexity, creativity, and experience requirement fall under this category. Such software requires a larger team size than the other two models and also the developers need to be sufficiently experienced and creative to develop such complex models.

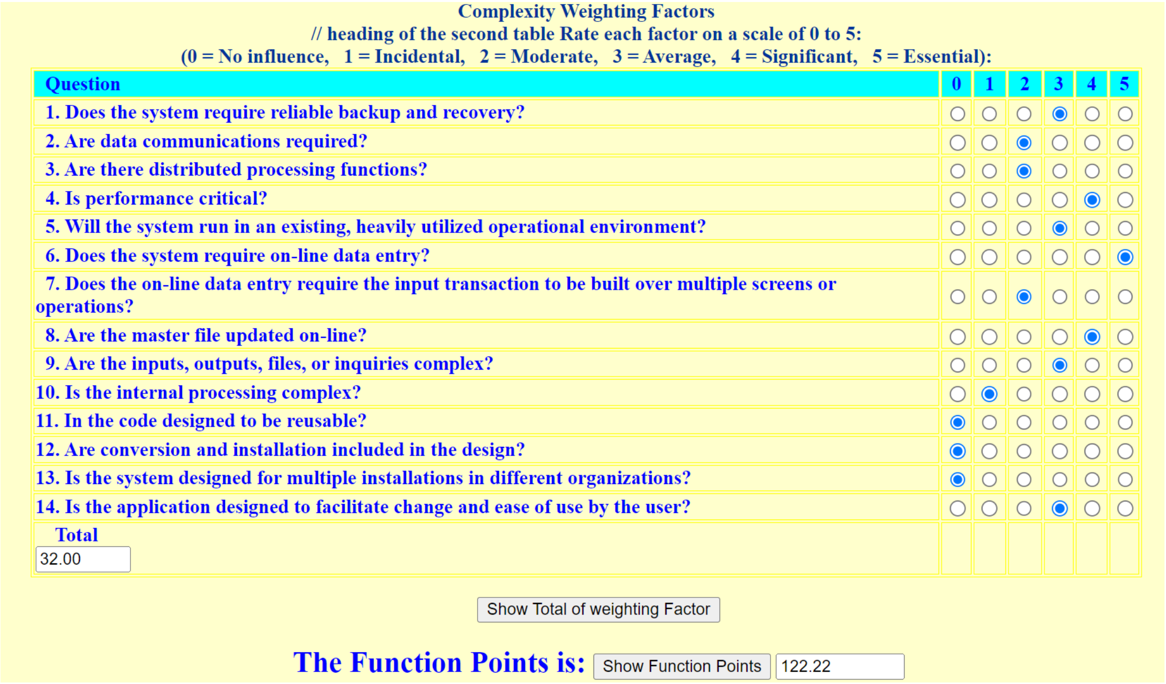
Step 1: Compute the count-total which will be used to define the complexity of a project.



Information Domain Values

Measurement Parameter	Count		Simple ○	Average ●	Complex ○		Total
Number of user inputs	8	X	3	4	6	=	32.00
Number of user outputs	3	X	4	5	7	=	15.00
Number of user inquiries	2	X	3	4	6	=	8.00
Number of files	5	X	7	10	15	=	50.00
Number of external interfaces	3	X	5	7	10	=	21.00
Count=Total							126.00

Step 2: Find the complexity adjustment values based on responses to the questions below:

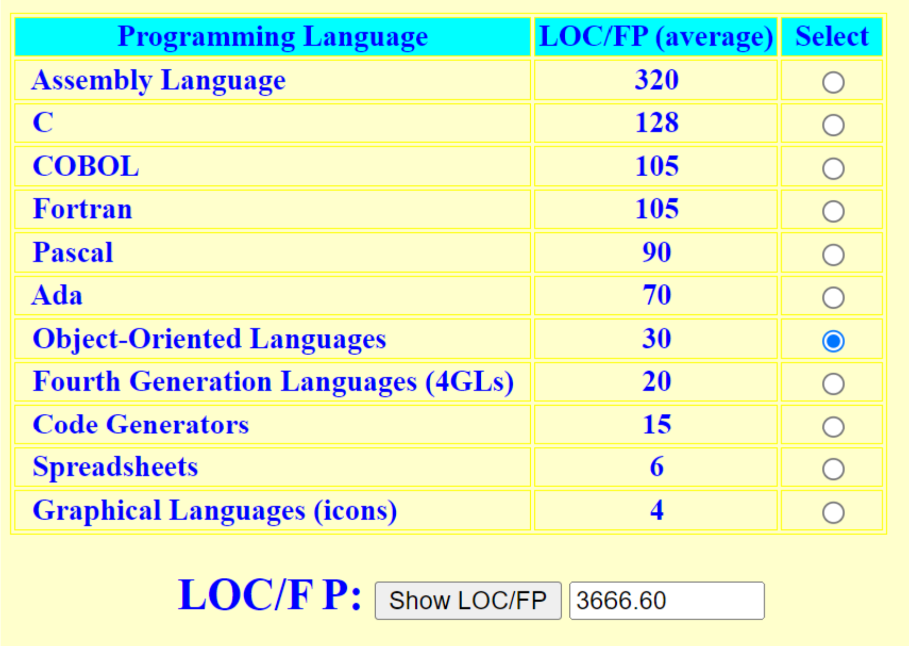


Complexity Weighting Factors
// heading of the second table Rate each factor on a scale of 0 to 5:
(0 = No influence, 1 = Incidental, 2 = Moderate, 3 = Average, 4 = Significant, 5 = Essential):

Question	0	1	2	3	4	5
1. Does the system require reliable backup and recovery?	○	○	○	●	○	○
2. Are data communications required?	○	○	●	○	○	○
3. Are there distributed processing functions?	○	○	●	○	○	○
4. Is performance critical?	○	○	○	○	●	○
5. Will the system run in an existing, heavily utilized operational environment?	○	○	○	●	○	○
6. Does the system require on-line data entry?	○	○	○	○	○	●
7. Does the on-line data entry require the input transaction to be built over multiple screens or operations?	○	○	●	○	○	○
8. Are the master file updated on-line?	○	○	○	○	●	○
9. Are the inputs, outputs, files, or inquiries complex?	○	○	○	●	○	○
10. Is the internal processing complex?	○	●	○	○	○	○
11. In the code designed to be reusable?	●	○	○	○	○	○
12. Are conversion and installation included in the design?	●	○	○	○	○	○
13. Is the system designed for multiple installations in different organizations?	●	○	○	○	○	○
14. Is the application designed to facilitate change and ease of use by the user?	○	○	○	●	○	○
Total 32.00						

The Function Points is: 122.22

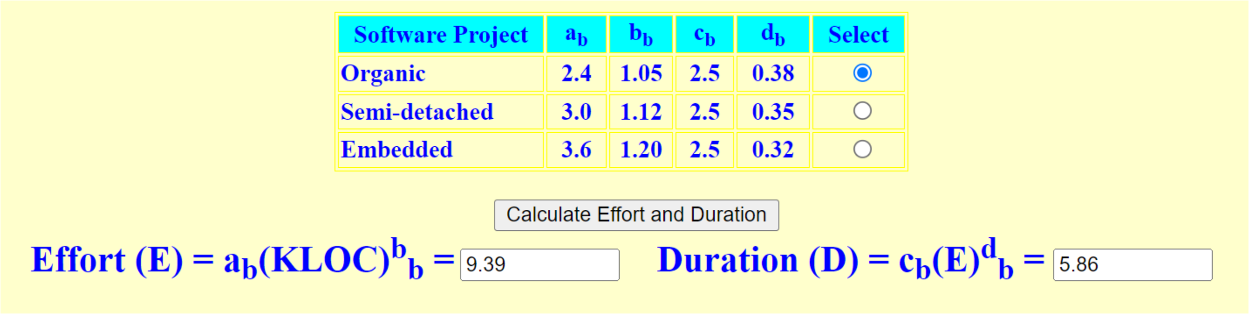
Step 3: Find LOC (Lines of Code) by choosing a programming language that we will using when developing a project:



Programming Language	LOC/FP (average)	Select
Assembly Language	320	○
C	128	○
COBOL	105	○
Fortran	105	○
Pascal	90	○
Ada	70	○
Object-Oriented Languages	30	●
Fourth Generation Languages (4GLs)	20	○
Code Generators	15	○
Spreadsheets	6	○
Graphical Languages (icons)	4	○

LOC/F P: 3666.60

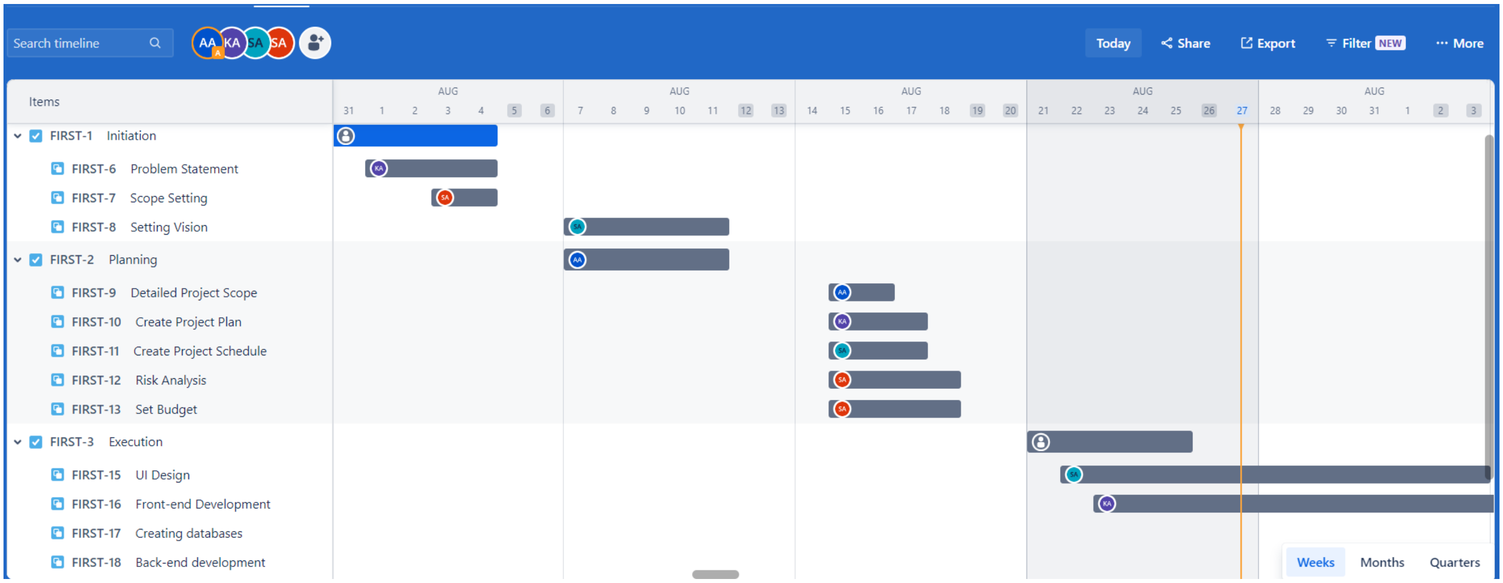
Step 4: Final Step is to select complexity of the software project:



Software Project	a_b	b_b	c_b	d_b	Select
Organic	2.4	1.05	2.5	0.38	●
Semi-detached	3.0	1.12	2.5	0.35	○
Embedded	3.6	1.20	2.5	0.32	○

Effort (E) = $a_b(KLOC)^{b_b}$ = 9.39 Duration (D) = $c_b(E)^{d_b}$ = 5.86

4.5. Gantt Chart



- FIRST-1 Initiation
  - FIRST-6 Problem Statement
  - FIRST-7 Scope Setting
  - FIRST-8 Setting Vision
- FIRST-2 Planning
  - FIRST-9 Detailed Project Scope
  - FIRST-10 Create Project Plan
  - FIRST-11 Create Project Schedule
  - FIRST-12 Risk Analysis
  - FIRST-13 Set Budget
- FIRST-3 Execution
  - FIRST-15 UI Design
  - FIRST-16 Front-end Development
  - FIRST-17 Creating databases
  - FIRST-18 Back-end development

5. Review and Risk Management

5.1 Inspection

An inspection is one of the most common sorts of reviews found in software projects. The goal of the inspection is for all of the inspectors to reach a consensus on a work product and approve it for use in the project.

Inspection Process Roles

There are various roles and responsibilities defined for an inspection process. Within an inspection team, five types of participants can be distinguished. They are:

- Author: The author is the person who has created the work product, that is to be inspected.
- Moderator: Moderator is the planner of the inspection. He is basically the coordinator and the leader.
- Reader: The person responsible for reading the documents.
- Reader/Scribe: This person is the one who records the defects that are being discovered.
- Inspector: Responsible for examining the product and identifying possible defects.

Inspection Process

An inspection process passes through the following stages:

- Planning: The moderator plans the activities to be performed during the inspection.
- Overview Meeting: An overview meeting intends to disseminate information regarding the background of the product under review.
- Preparation: Identify the possibility of the occurrence of a certain defect.
- Inspection Meeting: During this phase, the reader reads through the product and the inspector tries to point out the defects
- Rework: Based on the inspection meeting, the moderator makes necessary changes to the product.
- Follow up: The changes made are thus reviewed by the author.
- Approval: Once everyone on the team feels that the changes they identified are adequate, they can approve the updated work product without holding another inspection meeting.

5.2 Walkthrough

This is a process where the authors of the software as well as other associates are gathered in one place and discuss the software defects. Questions are made, comments are given, and answers are given to all the queries people have regarding the software. With all the members' satisfaction, conclusions are made.

Walkthrough Process Roles

- **Author–** Same as in the desk check, except when they act as the reader and scribe
- **Reader –** Reads or presents the deliverable being reviewed. Almost always the author.
- **Scribe –** Record comments and defects identified during the walk-through. Many times this role is filled by the author.
- **Reviewer –** Attends the walkthrough and provides feedback to the author. In more formal versions of a walkthrough, the reviewer may be asked to pre-read or pre-review the deliverable before the walkthrough.

Walkthrough Process

- The Author recruits reviewers.
- Send a meeting request for the walkthrough meeting. When remote participants are involved you will need to leverage collaborative tools that allow the deliverable to be shared and so that participants can hear each other.
- Send the deliverable or link the reviewers so they can prepare for the walkthrough. (Optional)
- During the walkthrough the reader presents or reads the deliverable to reviewers. The reviewers consider the material being presented and provide feedback that is recorded.
- After the walkthrough the author will need to follow up on the comments and defects generated during the walkthrough.

5.3 Deskcheck

The least formal and most common review technique is the desk check. A deskcheck is a simple review in which the author of a work product distributes it to one or more reviewers. In a deskcheck, the author sends a copy of the work product to selected project team members. The team members read it, and then write up defects and comments to send back to the author.

Deskchecks Process Roles

- **Author** – The author is the person that created the deliverable being reviewed. He or she sends the deliverable (or link) to the person doing the review, answers questions (if asked), and reacts to feedback.
- **Reviewer –** Reads the deliverable and provides feedback to the author.

Deskchecks Process

- The author recruits one or more reviewers. It is a common courtesy to always ask before sending a document to be reviewed. One has to make sure that they provide the reviewer with any relevant context.
- Send the deliverable or link to the reviewer.
- The reviewer reads and comments on the deliverable based on the agreed upon time box.

5.4 Risk Management Method - RMM

Risk Management

Risk management is the process of identifying, assessing and controlling threats to an organization's capital and earnings. Risk management is a process that allows individual risk events and overall risk to be understood and managed proactively, optimizing success by minimizing threats and maximizing opportunities and outcomes. Risk management is focused on anticipating what might not go to plan and putting in place actions to reduce uncertainty to a tolerable level.

RMM

RMMM Method can be broken down into three sections: Risk Mitigation, Monitoring and Management. All work is done as part of the risk analysis in this strategy. The project manager typically uses this RMMM plan as part of the overall project plan. Some development teams use a Risk Information Sheet (RIS) to document risk. For faster information handling, such as creation, priority sorting, searching, and other analyses, this RIS is controlled by a database system. Risk mitigation and monitoring will begin after the RMMM is documented and the project is launched.

Risk Mitigation

Risk Mitigation is a technique for avoiding risks. Risk Mitigation is the process of minimizing the likelihood of a risk being realized or the impact of risk.

The following are steps to take to reduce the risks:

- Risk mitigation (avoidance) is the primary strategy and is achieved through a plan
- Finding out the risk.
- Removing the factors that lead to risk occurrence.
- periodically examining the corresponding documents.
- carrying out timely reviews to speed up the work.

Risk Monitoring

It serves as a project tracking activity. In order to improve their strategy, businesses must monitor risk as a phase in the risk management process.

Its main goals are as follows:

- to check if predicted risks occur or not
- to make sure the risk-aversion measures are applied correctly.
- to gather information for next risk analysis.
- to identify which risks are responsible for which difficulties throughout the project

Risk Management

Risk management and planning are based on the assumption that the mitigation effort failed and the risk has become a reality. During that time it produces serious problems while the project manager is in charge of this responsibility. It is easier to manage risks if the project manager successfully implements project mitigation to eliminate risks. This demonstrates how a manager will respond to each risk. The risk register is the key objective of the risk management plan. This risk register identifies and prioritizes potential dangers to a software project.

5.5 Risk Plan Script

A risk plan is a list of all risks that threaten the project, along with a plan to mitigate some or all of those risks. Each of the risks in the plan must be assessed by the project manager and the team Risk assessment is an important part of planning a software project because it allows the project manager to predict potential problems that will threaten the project and take steps to mitigate those problems.

5.6 Risk Plan Script for our Project

Risk Plan for Project:	Online Learning Platform (UIU Based)			
Assessment Team Members:	Sumaiya, Adib, Anannaya, Aura			
Risk	**Prob.**	**Impact**	**Priority**	**Action**
Cyber Security Risks	3	5	15	Implement strong data security protocols, including encryption, access controls, and regular security audits. Adhere to industry best practices for data protection and privacy and train staff and users on data security practices.
Privacy and Data Security	2	4	12	Implementing robust data security measures, obtaining informed consent, providing privacy settings, and regularly updating privacy policies.
Scalability and performance	3	3	10	Design a scalable platform that adapts to user demand, ensuring smooth performance during peak usage by scaling infrastructure and resources accordingly.
Content Quality and Intellectual Property	1	1	5	Establish clear guidelines for content, ensuring accuracy, compliance with intellectual property laws, and up-to-date educational materials.
Aura may need to take a leave of absence to work on her thesis	1	1	3	1. Assign Annanaya to work with Aura to make sure she is cross-trained 2. If Aura is pulled off, she has to spend 10% of her time reviewing the project with Anannaya

6. Quality Control

6.1. Capability Maturity Model Integration - CMMI

The Capability Maturity Model Integration (CMMI) is a process improvement framework that was developed by the Software Engineering Institute (SEI) at Carnegie Mellon University in the United States.
CMMI provides a comprehensive set of guidelines for developing and managing products and services, with a focus on improving the quality, efficiency, and effectiveness of an organization's processes.

The CMMI model breaks down organizational maturity into five levels. For businesses that embrace CMMI, the goal is to raise the organization up to Level 5, the “optimizing” maturity level. Once businesses reach this level, they aren’t done with the CMMI. Instead, they focus on maintenance and regular improvements.

CMMI’s Maturity Levels are:

- **Maturity Level 0 – Incomplete:** At this stage work “may or may not get completed.” Goals have not been established at this point and processes are only partly formed or do not meet the organizational needs.
- **Maturity Level 1 – Initial:** Processes are viewed as unpredictable and reactive. At this stage, “work gets completed but it’s often delayed and over budget.” This is the worst stage a business can find itself in — an unpredictable environment that increases risk and inefficiency.
- **Maturity Level 2 – Managed:** There’s a level of project management achieved. Projects are “planned, performed, measured and controlled” at this level, but there are still a lot of issues to address.
- **Maturity Level 3 – Defined:** At this stage, organizations are more proactive than reactive. There’s a set of “organization-wide standards” to “provide guidance across projects, programs and portfolios.” Businesses understand their shortcomings, how to address them and what the goal is for improvement.
- **Maturity Level 4 – Quantitatively managed:** This stage is more measured and controlled. The organization is working off quantitative data to determine predictable processes that align with stakeholder needs. The business is ahead of risks, with more data-driven insight into process deficiencies.
- **Maturity Level 5 – Optimizing:** Here, an organization’s processes are stable and flexible. At this final stage, an organization will be in constant state of improving and responding to changes or other opportunities. The organization is

stable, which allows for more “agility and innovation,” in a predictable environment.

Once organizations hit Levels 4 and 5, they are considered high maturity, where they are continuously evolving, adapting and growing to meet the needs of stakeholders and customers.”

Goal of the CMMI: To create reliable environments, where products, services and departments are proactive, efficient and productive. Each maturity level includes a set of process areas that describe the key practices and activities necessary to achieve that level.

There is a total of 22 process areas in CMMI, grouped into four categories:

1. Process Management: This category includes process areas that focus on defining, implementing, and improving the organization's processes, such as Process and Product Quality Assurance (PPQA), Configuration Management (CM), and Organizational Process Definition (OPD).
2. Project Management: This category includes process areas that focus on managing individual projects, such as Project Planning (PP), Project Monitoring and Control (PMC), and Risk Management (RSKM).
3. Engineering: This category includes process areas that focus on the technical aspects of product development, such as Requirements Development (RD), Technical Solution (TS), and Product Integration (PI).
4. Support: This category includes process areas that support the other categories,
such as Measurement and Analysis (MA), Process and Product Quality Assurance (PPQA), and Configuration Management (CM). CMMI also provides a set of appraisal methods that organizations can use to assess their process maturity and capability. The most common appraisal method is the Standard CMMI Appraisal Method for Process Improvement (SCAMPI), which involves a formal assessment of the organization's processes against the CMMI model.

In summary, CMMI is a process improvement framework that provides a comprehensive set of guidelines for developing and managing products and services. It is organized into five maturity levels; each representing a different degree of process maturity and capability and includes a set of process areas that describe the key practices and activities necessary to achieve each level. CMMI also provides a set of appraisal methods that organizations can use to assess their process maturity and capability.

6.2. Six Sigma

Six Sigma ranks among the foremost methodologies for making business processes more effective and efficient. In addition to establishing a culture dedicated to continuous process improvement, Six Sigma offers tools and techniques that reduce variance, eliminate defects, and help identify the root causes of errors, allowing organizations to create better products and services for consumers. The Greek letter sigma refers to standard deviation—Six Sigma means "six standard deviations from the mean."

While most people associate Six Sigma with manufacturing, the methodology is applicable to every type of process in any industry. In all settings, organizations use Six Sigma to set up a management system that systematically identifies errors and provides methods for eliminating them.

The key sigma principles are the following:

- Customer focus
- Use data
- Improve continuously
- Involve people
- Be thorough

7. Project Management Tools

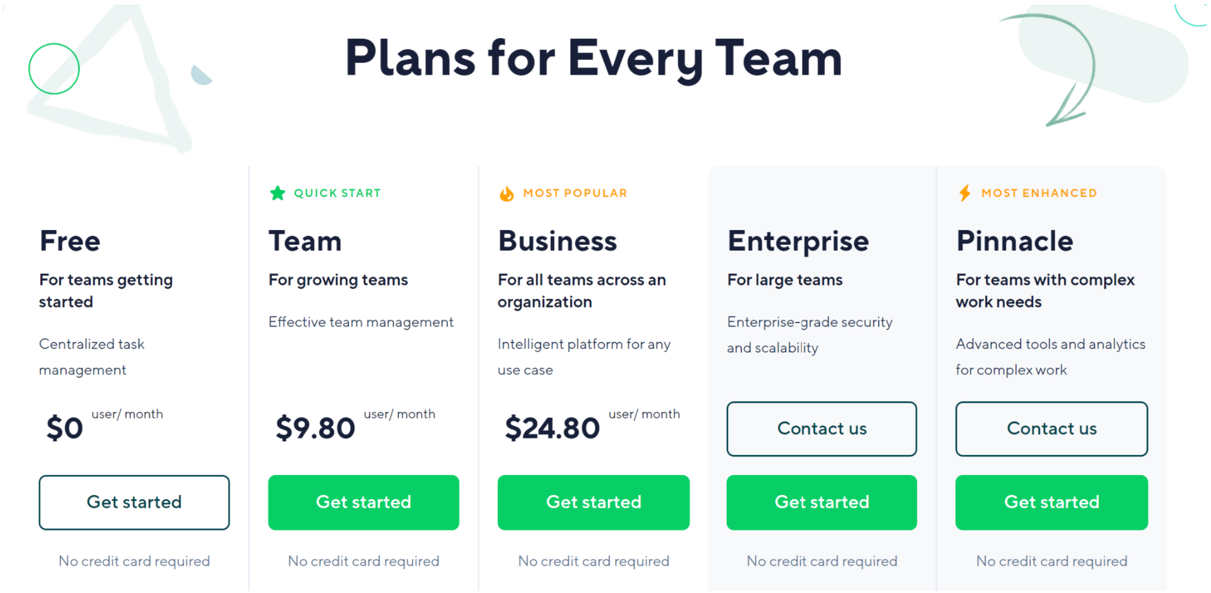
7.1 Introduction to Project Management Tools:

Project management software tools are digital platforms designed to
Facilitate and streamline the planning, Organizing, and execution of projects.
It helps project managers, team members, and stakeholders to collaborate effectively,
Manage tasks, track progress, and Achieve project goals more efficiently.

7.2 Wrike

A cloud-based project management and collaboration platform designed to help teams and organizations streamline their work processes, improve communication, and enhance productivity. It offers a suite of tools and features to manage tasks, projects, and workflows.

Pricing:



Plans for Every Team

Free	Team	Business	Enterprise	Pinnacle
	★ QUICK START	MOST POPULAR		MOST ENHANCED
For teams getting started	For growing teams	For all teams across an organization	For large teams	For teams with complex work needs
Centralized task management	Effective team management	Intelligent platform for any use case	Enterprise-grade security and scalability	Advanced tools and analytics for complex work
$0 user/ month	$9.80 user/ month	$24.80 user/ month	Contact us	Contact us
Get started	Get started	Get started	Get started	Get started
No credit card required	No credit card required	No credit card required	No credit card required	No credit card required

Pros:

- Collaborative
- Organization
- User Friendly
- User Interface
- Navigation

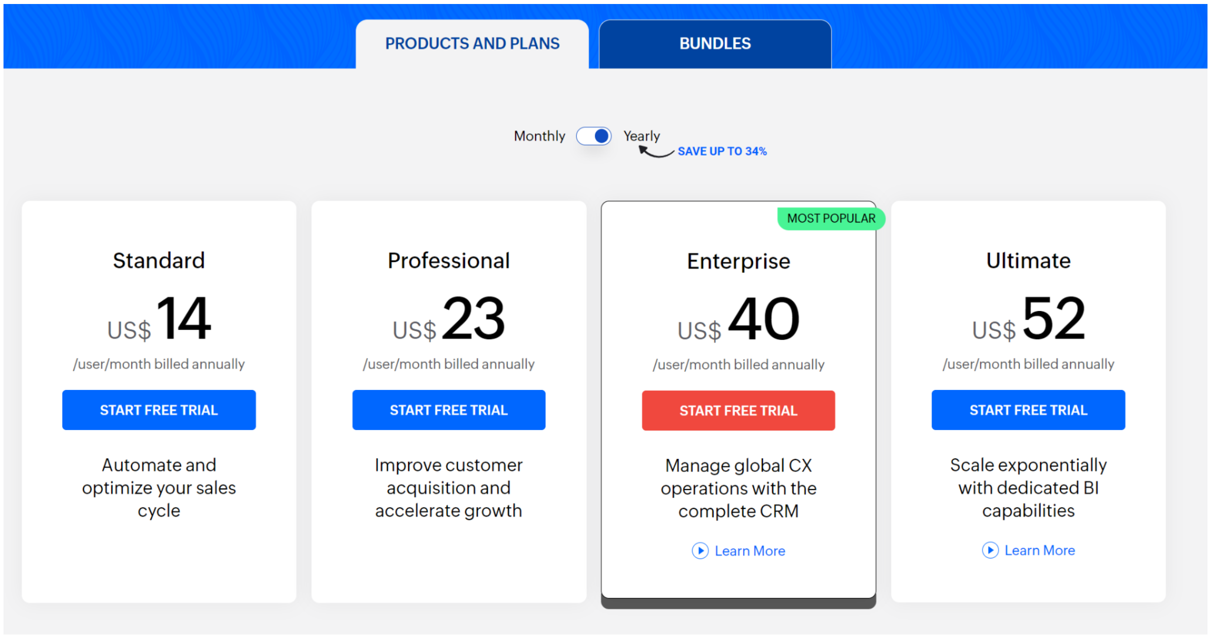
Cons:

- Takes Time
- Learning Curve
- Product Suggestion

7.3 Zoho

Zoho is a popular project management software used for managing sales, marketing campaigns, and customer support in a collaborative environment.

Pricing:



Standard	Professional	Enterprise	Ultimate
US$ 14	US$ 23	US$ 40	US$ 52
/user/month billed annually	/user/month billed annually	/user/month billed annually	/user/month billed annually
Automate and optimize your sales cycle	Improve customer acquisition and accelerate growth	Manage global CX operations with the complete CRM	Scale exponentially with dedicated BI capabilities

Pros:

- Affordable Pricing
- Customizable
- Integration Capabilities
- Scalability
- User-Friendly Interface

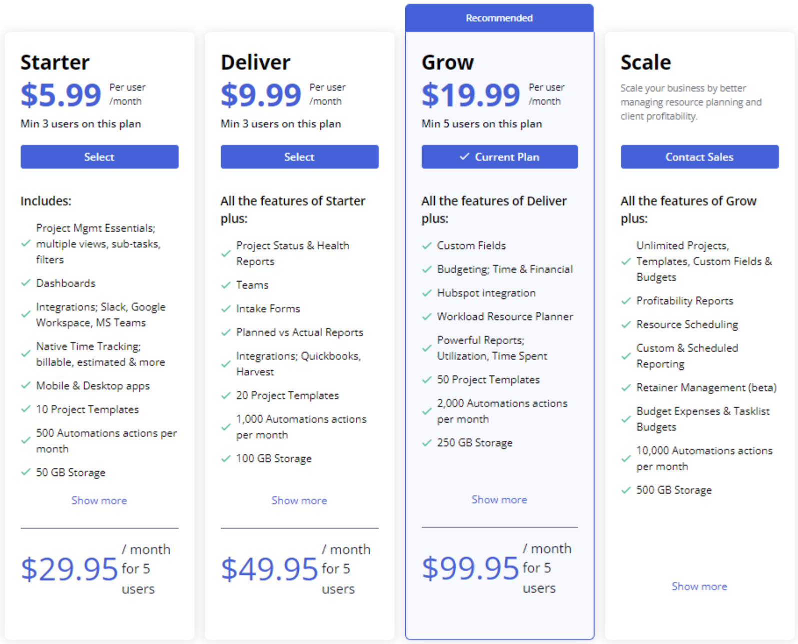
Cons:

- Limited custom fields
- Learning Curve for Advanced Features
- Low-tier users of Zoho CRM are limited by the number of users

7.4 Teamwork

Teamwork is one of the best cloud-based project management tools that lets you organize projects and tasks however you work best.

Pricing:



Starter
$5.99 Per user /month
Min 3 users on this plan

Includes:

- ✓ Project Mgmt Essentials; multiple views, sub-tasks, filters
- ✓ Dashboards
- ✓ Integrations; Slack, Google Workspace, MS Teams
- ✓ Native Time Tracking; billable, estimated & more
- ✓ Mobile & Desktop apps
- ✓ 10 Project Templates
- ✓ 500 Automations actions per month
- ✓ 50 GB Storage

$29.95 / month for 5 users

Deliver
$9.99 Per user /month
Min 3 users on this plan

All the features of Starter plus:

- ✓ Project Status & Health Reports
- ✓ Teams
- ✓ Intake Forms
- ✓ Planned vs Actual Reports
- ✓ Integrations; Quickbooks, Harvest
- ✓ 20 Project Templates
- ✓ 1,000 Automations actions per month
- ✓ 100 GB Storage

$49.95 / month for 5 users

Grow
$19.99 Per user /month
Min 5 users on this plan

All the features of Deliver plus:

- ✓ Custom Fields
- ✓ Budgeting; Time & Financial
- ✓ Hubspot integration
- ✓ Workload Resource Planner
- ✓ Powerful Reports; Utilization, Time Spent
- ✓ 50 Project Templates
- ✓ 2,000 Automations actions per month
- ✓ 250 GB Storage

$99.95 / month for 5 users

Scale
Scale your business by better managing resource planning and client profitability.

All the features of Grow plus:

- ✓ Unlimited Projects, Templates, Custom Fields & Budgets
- ✓ Profitability Reports
- ✓ Resource Scheduling
- ✓ Custom & Scheduled Reporting
- ✓ Retainer Management (beta)
- ✓ Budget Expenses & Tasklist Budgets
- ✓ 10,000 Automations actions per month
- ✓ 500 GB Storage

Pros:

- Comprehensive project management suite with various tools for collaboration and task management.
- Easy-to-use
- Easy to integration of third party apps

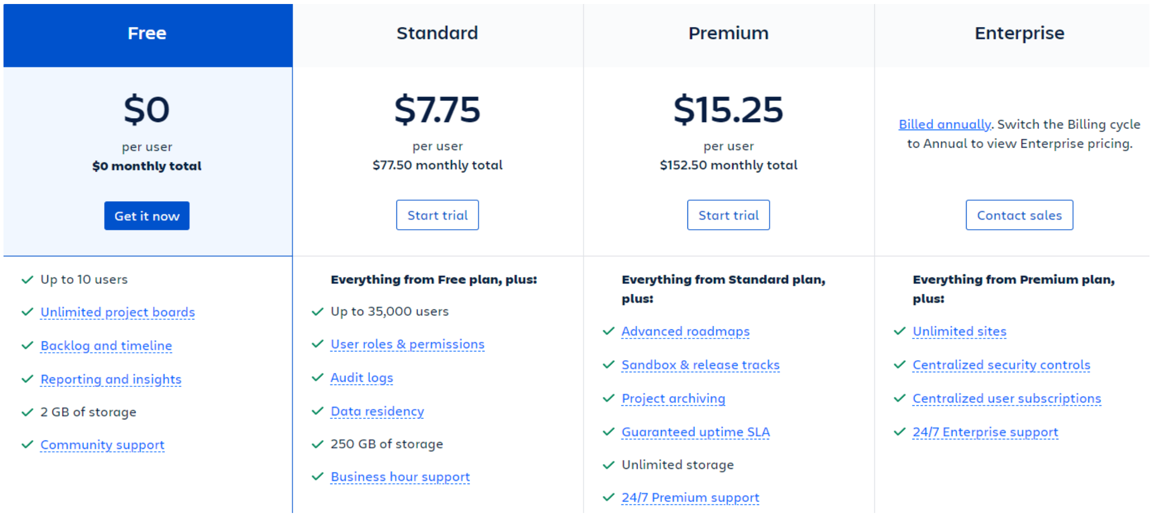
Cons:

- Relatively expensive - Need to buy paid version for bigger teams
- Learning curve for new users
- Customer service response slow

7.5 Jira

Jira is a popular project management software developed by Atlassian. It was originally released in 2002 and has since become one of the most widely used tools for managing software development and other types of projects.

Pricing:



Free	Standard	Premium	Enterprise
$0 per user **$0 monthly total**	$7.75 per user $77.50 monthly total	$15.25 per user $152.50 monthly total	Billed annually. Switch the Billing cycle to Annual to view Enterprise pricing.
Get it now	Start trial	Start trial	Contact sales
✓ Up to 10 users ✓ Unlimited project boards ✓ Backlog and timeline ✓ Reporting and insights ✓ 2 GB of storage ✓ Community support	**Everything from Free plan, plus:** ✓ Up to 35,000 users ✓ User roles & permissions ✓ Audit logs ✓ Data residency ✓ 250 GB of storage ✓ Business hour support	**Everything from Standard plan, plus:** ✓ Advanced roadmaps ✓ Sandbox & release tracks ✓ Project archiving ✓ Guaranteed uptime SLA ✓ Unlimited storage ✓ 24/7 Premium support	**Everything from Premium plan, plus:** ✓ Unlimited sites ✓ Centralized security controls ✓ Centralized user subscriptions ✓ 24/7 Enterprise support

Pros:

- Well-suited for Agile methodologies like Scrum and Kanban.
- Highly customizable.
- Integrates with numerous other tools and services such as version control systems.
- Provides built-in reporting features

Cons:

- Jira can be relatively expensive, especially for large teams or organizations.
- Configuring complex workflows in Jira might be time-consuming.

7. 6 Project management of our project with selected PM tool

We have used **Jira software** for our project, for the following reasons:

- Ease of Use
- Issue Tracking
- Agile Methodology
- Customizable Workflow
- Scalability
- Integration of External Tools