



**United International University**

**Course Name: Project Management**

**Course Code: PMG 4101**

**Section: C**

**Course Teacher**

**Dr. Suman Ahmmed**

**Director- CDIP & Associate Professor, UIU**

**United International University**

**Team Information**

**Team Name: Team AOS**

**Rezoan Ahmed 011193110**

**Mehedi Hasan 011191078**

**Md. Abu Bakar Siddik 011201048**

**Rukaiya Rahim Rumi 011202047**

**Rabbi Islam Yeasin 011201067**

# Table of Contents

<b>1.1. Project Description.....</b>	<b>3</b>
<b>1.2. Motivation.....</b>	<b>3</b>
<b>1.3. Stakeholders.....</b>	<b>4</b>
<b>2.1. Vision:.....</b>	<b>4</b>
<b>2.2. Scope:.....</b>	<b>5</b>
<b>2.3 Features List.....</b>	<b>6</b>
<b>3.1. What is Methodology:.....</b>	<b>7</b>
<b>3.2. Waterfall Methodology:.....</b>	<b>7</b>
3.2.1 - Waterfall Model Design:.....	7
3.2.2 Advantages of Waterfall Model :.....	8
3.2.3 Disadvantages of Waterfall Model :.....	9
<b>3.3. Agile Development Methodology:.....</b>	<b>9</b>
3.3.1. Agile Model - Design.....	10
3.3.2. Advantages of the Agile Model.....	12
3.3.3. Disadvantages of the Agile Model.....	13
<b>3.4. Scrum Methodology:.....</b>	<b>14</b>
3.4.1. Scrum Model - Design.....	14
3.4.2. Advantages of Scrum in the context of CSE BUDDY:.....	16
3.4.3. Disadvantages of Scrum in the context of CSE BUDDY:.....	17
<b>3.5. Spiral Model.....</b>	<b>18</b>
3.5.1. Spiral Model - Design.....	18
3.5.2. Advantages of the Spiral Model.....	21
3.5.3. Disadvantages of the Spiral Model:.....	21
<b>3.6. Extreme Programming.....</b>	<b>22</b>
3.6.1. Extreme Programming – Design.....	22
3.6.2. Advantages of the Extreme Programming.....	24
3.6.3. Disadvantages of the Extreme Programming.....	24
<b>3.7. Our Project Methodology.....</b>	<b>25</b>
Benchmark Analysis:.....	25
3.7.1. Selected Methodology for CSE BUDDY:.....	26
3.7.2. Phases of Agile (Scrum) Methodology throughout SDLC for CSE BUDDY:.....	26
<b>4.1 What is Work Breakdown Structure (WBS).....</b>	<b>27</b>
<b>4.2 WBS Design Principles:.....</b>	<b>28</b>
4.2.1 The 100% Rule.....	28
4.2.2 Planned Outcomes, Not Planned Actions.....	28
4.2.3 Level 2 is the Most Important.....	28
4.2.4 The Four Elements in Each WBS Element.....	28
4.2.5 Mutually-exclusive Elements.....	28
4.2.6 The 40-Hour Rule of Decomposition.....	28

4.2.7 The 4% Rule of Decomposition.....	29
<b>4.3. WBS of Our Project.....</b>	<b>29</b>
<b>While doing our WBS for our project, we mainly focused on these parts:.....</b>	<b>29</b>
<b>5.1 What is Wideband Delphi.....</b>	<b>30</b>
<b>5. 2 Wideband Delphi Process.....</b>	<b>31</b>
5.2.1 Choose the team.....	31
5.2.2 Kickoff-Meeting:.....	31
5.2.3 Individual Preparation.....	32
5.2.4 Estimation Session.....	33
5.2.5 Assemble Tasks.....	33
5.2.6 Review Result.....	34
<b>5.3 Individual Estimation Form.....</b>	<b>35</b>
<b>5.4 Estimation session.....</b>	<b>48</b>
<b>5.5 Assemble Table.....</b>	<b>49</b>
<b>6.1 What is Risk Management?.....</b>	<b>52</b>
<b>6.2 Risk Mitigation, Monitoring, and Management (RMMM).....</b>	<b>52</b>
6.2.1 Risk Mitigation.....	52
6.2.2 Risk Monitoring.....	52
6.2.3 Risk Management and Planning.....	53
<b>6.3 Risk Plan.....</b>	<b>54</b>
<b>6.4 Individual Risk Information Sheet.....</b>	<b>55</b>
Risk 1 : Server crashes (SC-1).....	55
Risk 2 : Financial constraints (FC-2).....	60
Risk 3 : Security Breaches (SB - 3).....	64
Risk 4 : Plagiarism of content (PC-4).....	67
Risk 5: Copyright Infringement ( CI - 5 ).....	71
<b>7.1 Importance of Knowledge Sharing.....</b>	<b>76</b>
<b>7.2 Methods of Knowledge Sharing.....</b>	<b>76</b>
<b>7.3 Benefits of Knowledge Sharing.....</b>	<b>77</b>
<b>8.1 Overview of CMMI.....</b>	<b>78</b>
<b>8.2 Maturity Levels of CMMI.....</b>	<b>78</b>
<b>8.3 Benefits of CMMI Implementation.....</b>	<b>78</b>
<b>9.1 Role at Level 3 (Defined).....</b>	<b>79</b>
Key Responsibilities:.....	79
<b>9.2 Role at Level 4 (Quantitatively Managed).....</b>	<b>79</b>
Key Responsibilities:.....	79
<b>9.3 Role at Level 5 (Optimizing).....</b>	<b>79</b>
Key Responsibilities:.....	79

# Chapter - 1: Introduction

## 1.1. Project Description

**CSE BUDDY** is an online platform dedicated to facilitating the learning journey of undergraduate students pursuing degrees in Computer Science and Engineering (CSE) in Bangladesh. The platform offers a comprehensive suite of educational resources, including eBooks, video courses, live classes, and a dynamic question-and-answer community, all presented exclusively in the Bangla language. **CSE BUDDY** aims to address critical challenges faced by CSE students, such as language barriers, financial constraints, and limited access to live online classes, by providing a secure, convenient, and user-friendly learning environment.

## 1.2. Motivation

The motivation behind **CSE BUDDY** stems from the recognition of significant hurdles faced by CSE students in Bangladesh. Traditional learning resources often fall short in addressing the diverse needs of students, particularly in terms of language accessibility and affordability. Additionally, the lack of live online classes and adequate support systems further exacerbate these challenges, hindering students' ability to excel in their studies and pursue successful careers in the field.

**CSE BUDDY** seeks to empower students by providing a dynamic, user-friendly platform that offers comprehensive learning materials, real-time interaction with instructors, and a supportive community environment. By addressing these key issues, **CSE BUDDY** aims to democratize access

to quality CSE education, enabling students from all backgrounds to thrive academically and professionally.

### 1.3. Stakeholders

The stakeholders for this project are:

1. Students
2. Educators and Instructors
3. Parents
4. Employers and Industry Professionals
5. Educational institutions.

## Chapter - 2: Vision Scope & Features

### 2.1. Vision:

Our vision is to make **CSE BUDDY** a secure, convenient, and user-friendly platform for students to learn, explore, and share knowledge with ease and broadly. Nowadays so many of our students suffer from anxiety about financial barriers in terms of accessing a helpful course as well as a quality full course. And CSE BUDDY, having an interactive class environment, will make sure that the platform is free of financial stress, an easy pace with language uses with good playlists on different subjects.

Below are some of our most significant benefits that one student can get from this platform:

- Help students to learn things more effectively because of having the contents in Bangla language.
- Financially weak students will be benefited due to the free of cost learning

experience.

- Live classes help to solve real time problems
- Availability of essential course materials help students to learn efficiently and faster.
- Emphasize hands-on projects that allow learners to apply theoretical knowledge to real-world problems.

## 2.2. Scope:

### **Scope:**

Implementation of an online learning platform dedicated to supporting undergraduate students pursuing degrees in Computer Science and Engineering (CSE) in Bangladesh, this platform aims to address critical challenges faced by CSE students, such as language barriers, limited access to educational resources, and financial constraints, by providing a comprehensive suite of learning tools and resources exclusively in the Bangla language.

The scope of building an educational platform related to Computer Science and Engineering (CSE) is vast and encompasses various aspects:

- Global Reach
- Lifelong Learning
- Academic Support
- Skill Development for Industry
- Entrepreneurship and Innovation
- Continuing Professional Development (CPD)
- Specialized Domains
- Community Building and Networking
- Social Impact

## 2.3 Features List

- Bangla CSE based contents
- Live Online classes
- eBook Library
- Problem Discussion portal / FAQ
- Group Studying system for collaborative learning • Free Online courses and materials
- Creating personalized profiles ( Both for Students and Instructors)
- Content Request System
- Creating a personalized course list with videos (For Instructors)
- Courses Managing and Tracking System

# Chapter - 3: Methodology

## 3.1. What is Methodology:

The development of software systems requires the use of software development approaches. Custom software development companies use various methods for software development in their daily work. Every one of them has benefits and drawbacks. These approaches primarily focus on providing effective software development in compliance with project specifications.

Below is a list of the benefits and drawbacks of five distinct software development methodologies:

- Waterfall Model
- Agile Development Model
- Scrum Development Methodology
- Spiral Model
- eXtreme Programming Methodology

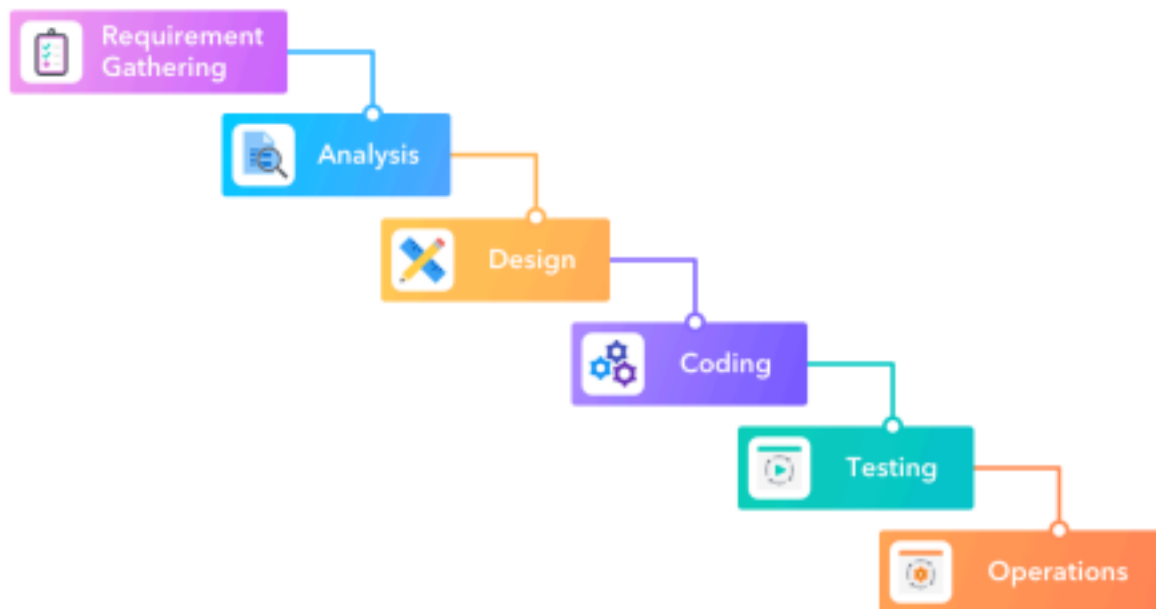
## 3.2. Waterfall Methodology:

The Waterfall methodology is a linear and sequential approach to software development. It consists of distinct phases that must be completed sequentially, with each phase serving as a foundation for the next. The phases typically include Requirements, Design, Implementation, Testing, Deployment, and Maintenance.

### 3.2.1 - Waterfall Model Design:

**The Waterfall Model's several phases are seen in the figure below.**





### Characteristics:

- **Sequential Process:** Sequential approach with distinct phases: Requirements, Design, Implementation, Testing, Deployment, and Maintenance.
- **Fixed Requirements:** Each phase must be completed before moving to the next.
- **Document-Driven:** Each phase produces specific documentation (e.g. requirements document, design document) that serves as a basis for subsequent phases.
- **Quality Emphasis:** Testing is typically conducted at the end of the development process to ensure that the final product meets quality standards.

### 3.2.2 Advantages of Waterfall Model :

- **Clarity:** Clear and well-defined phases provide structure and clarity to the development process.

- **Milestone-based Progress:** Milestones are clearly defined, making it easy to track progress and measure success.
- **Documentation:** Extensive documentation ensures that project requirements and design decisions are well-documented and understood by stakeholders.
- **Suitability for Stable Requirements:** Well-suited for projects with stable and clearly defined requirements.

### 3.2.3 Disadvantages of Waterfall Model :

- **Limited Flexibility:** Lack of flexibility for accommodating changes once the project has progressed beyond the requirements phase.
- **Late-stage Changes:** Changes requested late in the development process can be costly and time-consuming to implement.
- **Risk of Incomplete Requirements:** Requirements may not be fully understood or accurately captured at the beginning of the project, leading to potential gaps or misunderstandings.
- **Limited Stakeholder Involvement:** Limited opportunities for stakeholder involvement and feedback until the later stages of the project.

### 3.3. Agile Development Methodology:

Agile is a design framework that was originally used for software development but is now used in other areas where there is a need to complete large tasks, including project management tasks. It is an alternative to typical project management frameworks, such as waterfall, which is a linear process.

### 3.3.1. Agile Model - Design

Agile design states that product design and development are iterative and incremental processes. In other words, large product releases (known as milestones) should be broken down into smaller, more manageable chunks. Here is a pictorial view of Agile Model:



### Characteristics of Agile Development Model:

The Agile Model, particularly in the context of the CSE Buddy project, exhibits several key characteristics that align with Agile principles and methodologies. Here are the characteristics of the Agile Model as applied to the CSE Buddy project:

- Iterative and Incremental Development:** Agile methodologies, such as Scrum or Kanban, emphasize iterative and incremental development. The CSE Buddy project likely follows this approach by breaking down the development process into smaller, manageable iterations or sprints, with each iteration delivering a potentially shippable product increment.
- Customer Collaboration and Feedback:** Agile methodologies prioritize close

collaboration with customers, stakeholders, and end-users throughout the project lifecycle. In the case of **CSE Buddy**, stakeholders, including undergraduate students pursuing degrees in Computer Science and Engineering, educators, and content creators, are actively involved in providing feedback, validating requirements, and shaping the direction of the project.

- **Adaptability to Change:** Agile methodologies are highly adaptable and responsive to change. The CSE Buddy project embraces this characteristic by welcoming changes to requirements, priorities, and features even late in the development process. Agile teams recognize that change is inevitable and leverage it as a competitive advantage to deliver value more effectively.
- **Emphasis on Delivering Value:** Agile methodologies prioritize delivering value to customers and stakeholders early and continuously. The CSE Buddy project focuses on delivering tangible results and meaningful features in short iterations, allowing stakeholders to see progress and provide feedback iteratively.
- **Self-Organizing and Cross-Functional Teams:** Agile teams are self-organizing and cross-functional, meaning they have the autonomy to organize and manage their work and possess the diverse skills necessary to deliver the product increment. In the context of CSE Buddy, project teams comprise developers, designers, educators, and other stakeholders collaborating to achieve project goals.
- **Continuous Improvement:** Agile methodologies promote a culture of continuous improvement and reflection. Teams regularly reflect on their processes, seek opportunities for optimization, and adapt their practices to improve productivity, quality, and customer satisfaction. The CSE Buddy project likely conducts regular retrospectives to identify areas for improvement and implement changes accordingly.

- **Transparent Communication:** Agile methodologies emphasize transparent and open communication among team members, stakeholders, and customers. The CSE Buddy project fosters transparent communication channels to facilitate collaboration, share progress updates, address challenges, and ensure alignment with project goals and objectives.
- **Focus on Working Software:** Agile methodologies prioritize delivering working software that meets customer needs and adds value to the business. The CSE Buddy project focuses on developing functional components and features that enhance the learning experience for undergraduate students pursuing degrees in Computer Science and Engineering in Bangladesh.

In summary, the characteristics of the Agile Model in the context of the CSE Buddy project include iterative and incremental development, customer collaboration and feedback, adaptability to change, emphasis on delivering value, self-organizing and cross-functional teams, continuous improvement, transparent communication, and focus on working software. These characteristics collectively contribute to the project's success in delivering a high-quality, user-centric learning platform tailored to the needs of CSE students in Bangladesh.

### 3.3.2. Advantages of the Agile Model

- **Flexibility and Adaptability:** Agile methodologies, such as Scrum and Kanban, allow for flexibility in responding to changes in requirements, priorities, and market conditions. Teams can adjust their approach and priorities based on ongoing feedback and evolving project needs.
- **Customer Collaboration:** Agile encourages close collaboration between development teams and customers or stakeholders throughout the project lifecycle. This collaboration ensures that the product meets the customer's expectations and delivers business value.
- **Early and Continuous Delivery:** Agile emphasizes delivering working software in short iterations, typically ranging from one to four weeks. This allows

stakeholders to see tangible progress early and provides opportunities for feedback and validation.

- **Iterative Improvement:** Agile promotes continuous improvement through regular retrospectives and feedback loops. Teams reflect on their processes, identify areas for improvement, and make adjustments to enhance productivity, quality, and customer satisfaction.
- **Reduced Risk:** By breaking the project into small, manageable iterations and delivering incremental value, Agile mitigates project risks associated with changing requirements, technology uncertainties, and market dynamics.

### 3.3.3. Disadvantages of the Agile Model

- **Lack of Predictability:** Agile projects may lack predictability in terms of scope, schedule, and budget due to their adaptive and iterative nature. This can make it challenging for stakeholders to estimate project timelines and make long-term plans.
- **Resource Intensive:** Agile methodologies require active involvement and collaboration from team members, stakeholders, and customers throughout the project lifecycle. This can be resource-intensive and may require dedicated time and effort from all involved parties.
- **Dependency on Team Dynamics:** The success of Agile projects heavily depends on the effectiveness of team collaboration, communication, and self-organization. Team dynamics, including trust, accountability, and shared understanding, play a crucial role in achieving project goals.
- **Complexity in Scaling:** While Agile methodologies work well for small to medium-sized projects and teams, they can become complex to scale for larger projects or organizations. Managing dependencies, coordination, and alignment across multiple Agile teams can pose challenges.

- **Documentation Challenges:** Agile methodologies prioritize working software over comprehensive documentation. While this fosters collaboration and flexibility, it can lead to challenges in maintaining documentation, especially for regulatory compliance or knowledge transfer purposes.

### **3.4. Scrum Methodology:**

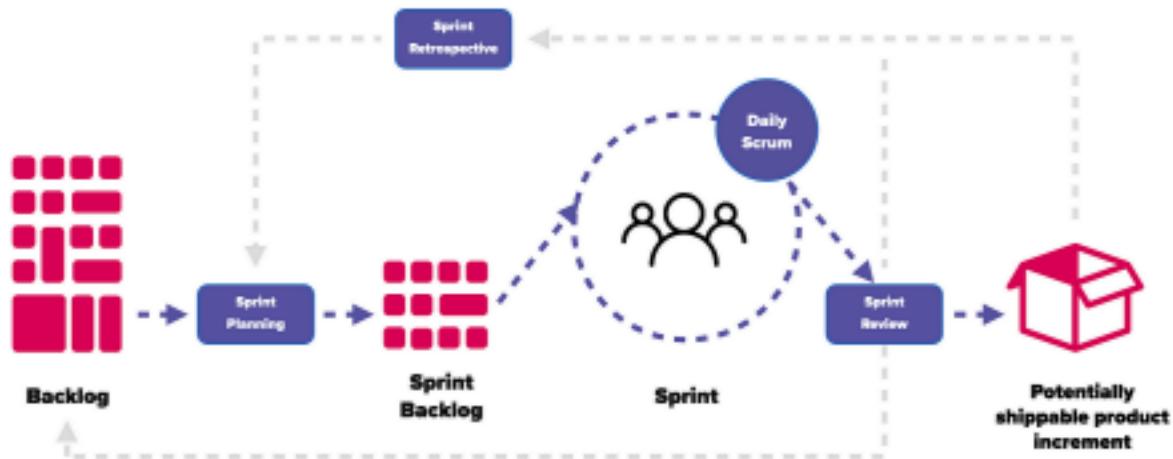
Scrum is an agile framework designed to help teams work together to efficiently address complex adaptive problems, while productively and creatively delivering products of the highest possible value. It emphasizes collaboration, flexibility, and speed in response to changing requirements.

Scrum facilitates this through regular, iterative progress on a product or project, divided into short work cycles known as sprints. Within these sprints, teams work to complete tasks. Scrum roles include the Scrum Master, who guides the team in following Scrum practices; the Product Owner, who represents the stakeholders' interests; and the Development Team, who are the individuals working on the project. Through its structured yet flexible approach, Scrum helps teams manage their work efficiently.

#### **3.4.1. Scrum Model - Design**

This approach allows for adjustments based on feedback and self-reflection on the team's work processes, aiming to continuously improve both the product and the way the team functions.

**The Scrum Model's several phases are seen in the figure below.**



Certainly, let's analyze the Scrum methodology in the context of the project "CSE BUDDY" an online platform for Computer Science and Engineering students with content in Bangla.

### Characteristics of Scrum Model:

#### • Roles and Responsibilities:

**Product Owner:** Someone deeply familiar with Bangla language and the CSE curriculum could hold this role, ensuring that the platform meets the educational needs of the students.

**Scrum Master:** This person would be responsible for facilitating Scrum practices, removing obstacles, and ensuring that the development team can work effectively.

**Development Team:** A cross-functional group that includes course creators, software developers, testers, and UX/UI designers who work collaboratively.

#### • Sprints:

The platform development is divided into short phases, focusing initially on the most critical features like the Q&A forum or the eBook library.

#### • Daily Stand-Up Meetings:

Quick, daily check-ins to ensure the content creation is on track and the development aligns with student learning patterns.



- **Product Backlog:**

- A prioritized list that might include features like video course integration, interactive exercises, and community engagement tools.

- **Sprint Planning:**

Deciding on the features to be developed in the next sprint, possibly based on the urgency of students' needs, like preparing for exams or submission deadlines.

- **Sprint Review:**

- Evaluating newly developed features, such as the user interface for the eBook library or the functionality of the practice exercises.

- **Sprint Retrospective:**

- Reflecting on the sprint to improve the content delivery and platform functionality for the next sprint.

- **Incremental Delivery:**

- Regular updates to the platform ensure that students receive new and improved features, such as additional courses or enhanced community features.

### 3.4.2. Advantages of Scrum in the context of CSE BUDDY:

- **Flexibility and Adaptivity:**

- As the needs of the students evolve, the platform can adapt by incorporating new learning materials and tools.

- **Client Involvement:**

- Regular feedback from students and educators can be incorporated swiftly to make the platform more effective.

- **Transparency:**

- The development process is visible to all stakeholders, including students, educators, and investors.

- **Early and Predictable Delivery:**

- Students can expect new features and content regularly, which can help with their academic planning.

- **Improved Quality:** - Iterative cycles allow for continuous refinement of the platform based on user engagement and feedback.

### 3.4.3. Disadvantages of Scrum in the context of CSE BUDDY:

- **Scope Creep:**
  - Without careful management, the desire to constantly add new features could delay the platform's progress or overwhelm students with too many updates.
- **Requires Experienced Team Members:**
  - The team must be adept not just in software development but also in understanding educational content and the specific needs of the student demographic.
- **Not Suitable for All Projects:**
  - If there are very rigid deadlines tied to academic calendars, the flexibility of Scrum may be less beneficial.
- **Daily Meetings Can Be Overhead:**
  - If not managed efficiently, daily meetings might take time away from content creation and development work.
- **Relies Heavily on Teamwork:**
  - The success of the platform depends on the collaborative efforts of team members who may have varying degrees of familiarity with Scrum methodology.

In the case of **CSE BUDDY**, the Scrum framework can offer significant benefits due to its iterative nature and focus on user feedback. This allows the platform to evolve with the needs of its users. However, it will require careful planning and experienced team members who are capable of navigating the challenges

associated with Scrum.

### **3.5. Spiral Model**

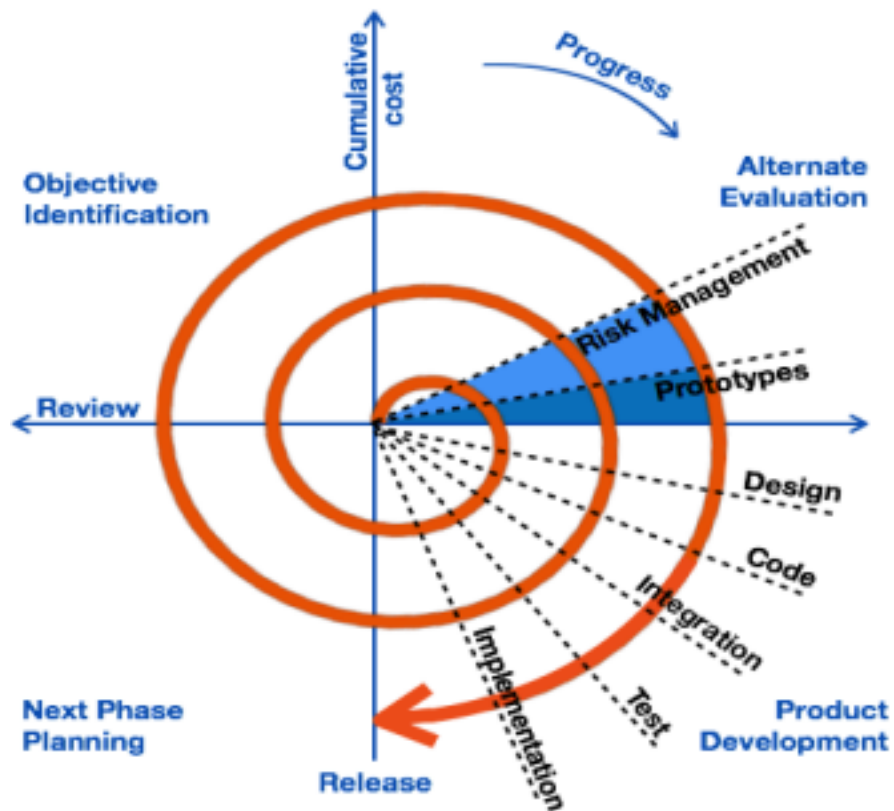
The spiral model is similar to the incremental development of a system, with more emphasis placed on risk analysis. The spiral model has four phases: Planning, Design, Construction, and Evaluation. A software project repeatedly passes through these phases in iterations (called Spirals in this model).

#### **3.5.1. Spiral Model - Design**

The design phase starts with the design in the baseline spiral and involves architectural, logical design of modules, physical product design, and final design in the successive spirals.

#### **Characteristics Of Spiral Development Model:**

The characteristics of the Spiral Model in project management, as applied to the CSE Buddy project, can be described based on its iterative and risk-driven approach to software development. Here are the characteristics of the Spiral Model in the context of the CSE Buddy project:



- Iterative Development:** The Spiral Model emphasizes iterative development, with the project progressing through multiple cycles or iterations. Each iteration involves planning, implementing, evaluating, and refining project components, allowing for continuous improvement and evolution of the software product.
- Risk Management:** A key characteristic of the Spiral Model is its focus on risk management. The model incorporates risk analysis and mitigation activities into each iteration, allowing project teams to identify, assess, and address potential risks early in the development process. This proactive approach to risk management helps minimize project uncertainties and enhances the overall success probability of the project.
- Phases of the Spiral:** The Spiral Model consists of four main phases: identification of objectives, risk analysis and evaluation, development and

testing, and planning for the next iteration. Each phase represents a cycle of activities aimed at addressing specific project objectives and managing associated risks.

- **Flexibility:** The Spiral Model offers flexibility in accommodating changes and adaptations throughout the project lifecycle. As project requirements evolve or new risks emerge, the model allows for adjustments to be made to the project plan and implementation strategy. This flexibility enables the project team to respond effectively to changing stakeholder needs and market dynamics.
- **Client Involvement:** The Spiral Model encourages active involvement and collaboration with clients or stakeholders throughout the development process. Stakeholder feedback and input are solicited and incorporated into each iteration, ensuring that the final product aligns closely with user expectations and requirements.
- **Progressive Elaboration:** The Spiral Model supports progressive elaboration of project requirements and design. Initial iterations focus on high-level requirements and conceptual design, with subsequent iterations adding more detail and refinement based on stakeholder feedback and validation.
- **Continuous Evaluation and Improvement:** A fundamental characteristic of the Spiral Model is its emphasis on continuous evaluation and improvement. Project teams regularly assess the outcomes of each iteration, identify lessons learned, and incorporate feedback into subsequent iterations to enhance the quality and effectiveness of the software product.

In summary, the characteristics of the Spiral Model in the context of the CSE Buddy project include iterative development, risk management, flexibility, client involvement, progressive elaboration, and continuous evaluation and

improvement. These characteristics collectively contribute to the successful planning, execution, and delivery of the project's objectives while effectively managing project risks and uncertainties.

### 3.5.2. Advantages of the Spiral Model

- **Risk Management:** One of the primary advantages of the Spiral Model is its focus on risk management. The model allows for early identification and mitigation of risks through iterative prototyping and evaluation. This helps in reducing the overall project risk.
- **Flexibility:** The Spiral Model is highly flexible and can accommodate changes in requirements, design, and even project scope more effectively compared to traditional waterfall models. It allows for incremental development and refinement based on feedback obtained during each cycle.
- **Client Involvement:** The model promotes continuous client involvement and feedback throughout the development process. This ensures that the final product aligns closely with client expectations and requirements.
- **Early Prototyping:** The Spiral Model encourages the development of early prototypes or versions of the software, which can be reviewed and evaluated by stakeholders. This early feedback helps in refining the requirements and design, leading to a higher-quality end product.

### 3.5.3. Disadvantages of the Spiral Model:

- **Complexity:** The Spiral Model can be more complex compared to traditional linear models like the waterfall model. It requires careful planning, coordination, and management of multiple iterations, which can increase project overhead and complexity.
- **Resource Intensive:** The iterative nature of the Spiral Model can be

resource-intensive, requiring additional time, effort, and resources compared to other development models. This can result in higher development costs and longer time-to-market.

- **Uncertainty in Estimation:** Estimating project timelines and costs can be challenging in the Spiral Model due to its iterative and incremental nature. The uncertainty associated with the number and duration of iterations can make it difficult to accurately predict project milestones and deliverables.
- **Dependency on Risk Analysis:** The effectiveness of the Spiral Model heavily depends on the accuracy and thoroughness of risk analysis conducted at each iteration. If risks are not properly identified, analyzed, and managed, it can lead to project delays, cost overruns, and quality issues.

### 3.6. Extreme Programming

Extreme Programming (XP) is an agile software development framework that focuses on making higher-quality software and higher quality of life for the development team. XP is one of the most distinct agile methodology among the appropriate engineering practices for software development because it aims for speed and simplicity with small sprint cycles.

#### 3.6.1. Extreme Programming – Design

The Extreme Programming Model's phases are seen in the figure below.

## XP Workflow



The

### characteristics of extreme programming model

Extreme Programming (XP) boasts several key characteristics that differentiate it from other agile methodologies. The key characteristics are:

- **Communication:** Open and constant dialogue between developers, customers, and stakeholders.
- **Simplicity:** Favoring clear, concise code that avoids unnecessary complexity.
- **Feedback:** Obtaining continuous feedback on the software through testing, reviews, and user interaction.
- **Respect:** Valuing the contributions and perspectives of everyone involved in the project.
- **Courage:** Embracing change and being willing to make difficult decisions for the project's benefit.
- **Small Releases:** Delivering functional software in short cycles (weeks or even days), fostering quicker feedback and adaptation.
- **Pair Programming:** Two developers collaborating on a single machine, promoting knowledge sharing and code quality.
- **Continuous Integration:** Merging code changes frequently into the main



codebase, minimizing errors and ensuring seamless integration.

- **Test-Driven Development (TDD):** Writing unit tests beforehand to define desired behavior and ensure code correctness.
- **Refactoring:** Continuously improving code structure and design without affecting functionality, maintaining codebases long-term.
- **Collective Ownership:** Everyone on the team shares responsibility for code quality and actively contributes to its improvement.

### 3.6.2. Advantages of the Extreme Programming

- Reduces error through pair programming
- There is no overtime, teams work at their own pace
- With short notice changes can be made
- Code is clear and understandable
- Close contact with the client

### 3.6.3. Disadvantages of the Extreme Programming

- It is highly costly
- Customer needs to participate in the process
- It requires large time

### 3.7. Our Project Methodology

#### Benchmark Analysis

Criteria	Waterfall	Agile	Scrum	Spiral	Extreme Programming
Flexibility	Low	High	High	Moderate	High
Client Involvement	Low	High	High	High	High
Predictability	High	Low	Moderate	Low	Moderate
Resource Intensity	Low	High	Moderate	High	High
Risk Management	Moderate	Moderate	Moderate	High	High
Adaptability to Change	Low	High	High	Moderate	High
Iterative Development	No	Yes	Yes	Yes	Yes
Early Delivery of Features	No	Yes	Yes	No	Yes
Client Satisfaction	Low	High	High	High	High

### 3.7.1. Selected Methodology for CSE BUDDY:

In the sections above, we discussed the most commonly used methodologies in software development. After analyzing all of them, we finally came to choose the **Scrum Development Model** as our project methodology. The reasons behind choosing this methodology are:

**Flexibility:** Agile methodologies offer flexibility, allowing for changes and adaptations throughout the project lifecycle, which is crucial for a project like **CSE BUDDY** where requirements may evolve.

**Continuous Improvement:** Agile methodologies promote continuous improvement through iterative development and regular feedback loops, ensuring that the platform evolves to meet the needs of its users effectively.

**Client Involvement:** Scrum encourages active client involvement throughout the development process, ensuring that the platform aligns closely with the expectations and requirements of its users.

**Early and Predictable Delivery:** Scrum enables early and predictable delivery of features, allowing users to see tangible progress and provide feedback iteratively.

**Iterative Development:** The iterative nature of Agile methodologies allows for incremental development and refinement, ensuring that the platform remains adaptable to changing market conditions and user needs.

### 3.7.2. Phases of Agile (Scrum) Methodology throughout SDLC for CSE BUDDY:

- **Planning:** Define the project vision, create the product backlog, and prioritize features based on user needs.

- **Sprint Planning:** Select user stories from the product backlog for the upcoming sprint, estimate the effort required, and commit to delivering the selected items.
- **Sprint:** Develop and test the selected user stories within the sprint duration (typically 2-4 weeks).
- **Daily Stand-up:** Daily meetings to discuss progress, challenges, and plan for the day's work.
- **Sprint Review:** Review the completed user stories with stakeholders and gather feedback for future iterations.
- **Sprint Retrospective:** Reflect on the sprint, identify what went well and what could be improved, and make adjustments for the next sprint.
- **Repeat:** Repeat the sprint cycle, continuously delivering value to users and stakeholders while incorporating feedback and improvements.

By following the Agile (Scrum) methodology, the CSE BUDDY project can ensure continuous improvement, client satisfaction, and timely delivery of features, ultimately resulting in a successful and user-friendly learning platform for CSE students.

## Chapter 4: Work Breakdown Structure

### 4.1 What is Work Breakdown Structure (WBS)

A Work Breakdown Structure (WBS) is a hierarchical decomposition of the total scope of work to be carried out by the project team to accomplish the project objectives and create the required deliverables. It organizes and defines the

scope of the project in a hierarchical structure, breaking down complex tasks into smaller, more manageable components.

## **4.2 WBS Design Principles:**

### **4.2.1 The 100% Rule**

Every component in the WBS should represent 100% of the work necessary to complete the project objectives. The WBS should not overlook any aspect of the project scope.

### **4.2.2 Planned Outcomes, Not Planned Actions**

Each element in the WBS should represent a project deliverable or outcome rather than the actions required to produce it. This ensures that the focus remains on the end result rather than the process.

### **4.2.3 Level 2 is the Most Important**

Level 2 of the WBS hierarchy is critical as it defines the major deliverables of project phases. It serves as the backbone of the WBS and provides a clear overview of the project scope.

### **4.2.4 The Four Elements in Each WBS Element**

Each WBS element should include a unique identifier, a description of the work, the responsible party, and a clear acceptance criterion. These elements ensure clarity and accountability for each component of the project.

### **4.2.5 Mutually-exclusive Elements**

WBS elements should be mutually exclusive, meaning that no work is duplicated, and each element fits within one category without overlap.

### **4.2.6 The 40-Hour Rule of Decomposition**

Work packages in the WBS should be decomposed to a level where each task can be completed in approximately 40 hours. This ensures that tasks are manageable and well-defined.

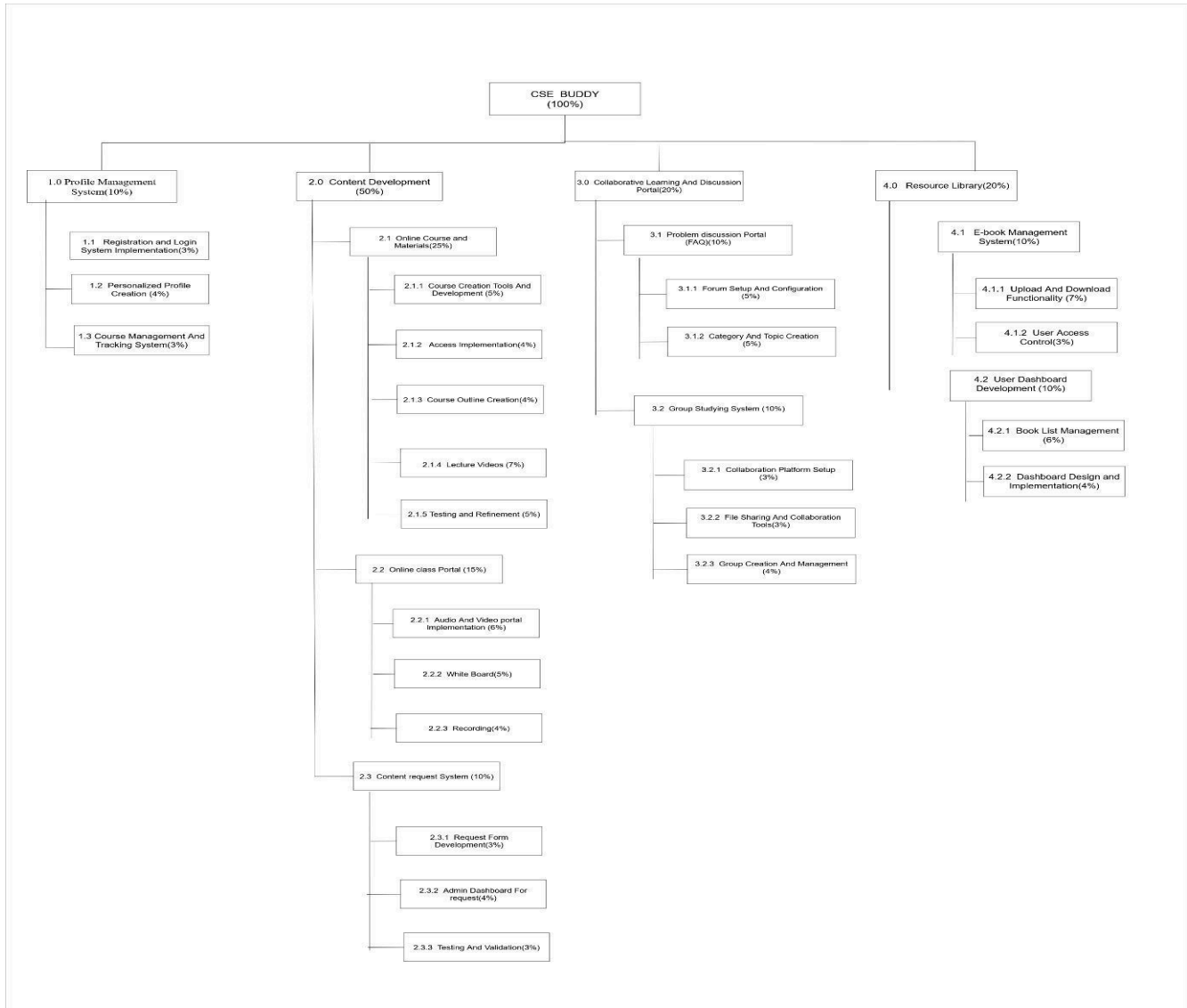
#### **4.2.7 The 4% Rule of Decomposition**

Each level of the WBS should be decomposed into approximately 4% to 8% of the level above it. This guideline helps maintain the balance between detail and complexity within the WBS.

### **4.3. WBS of Our Project**

While doing our WBS for our project, we mainly focused on these parts:

- Followed the 100% Rule for the breakdown.
- In our project perspective we basically focused on our project outcome, not the actions.
- Do not repeat any of our tasks or subtasks which can violate the 100% Rule.
- Followed the 4% Rule for decomposing our work breakdown structure



## Chapter 5: Wideband Delphi

### 1 What is Wideband Delphi

Wideband Delphi Technique is a consensus-based estimation technique for estimating effort. It is useful when estimating the time needed to complete a task. Participation of experienced people and individual estimating would lead to reliable results.

## 2 Wideband Delphi Process

- Choose the team
- Kick-off Meeting
- Individual Preparation
- Estimation Session
- Assemble tasks
- Review result

### 5.2.1 Choose the team

- Project manager chooses the team
  - Select Coordinator
    - There will be 3 to 7 members under the coordinator.
    - Coordinator must have an idea about Delphi.
  - Project Manager can not be a moderator
  - Team members must be willing to work with each task honestly and work with the team.
- In our project:
- We have created a project team
  - There 5 members in our project
    - Everyone of us have the idea of Delphi method
  - Team members must be willing to work with each task honestly and work with the team

### 5.2.2 Kickoff-Meeting:

The moderator explains wideband Delphi to a new evaluator.

- If any team member hasn't any idea or read the vision and scope and any supporting documentation, the facilitator will review it.
- Generate a task list of 10-20 main tasks. These tasks will represent the highest level of the work breakdown structure.
- Agreed units of estimate.
  - Moderator guides the entire discussion, monitors time and prepares a structure



document containing problem specification, high level task list and assumption.

→ In our project:

- Our team leader explains wide-band to all other team members.
- We have discussed the vision and scope in our kickoff-meeting among our team members.
- Generate a task list of 5 to 8 main tasks and these 5 to 8 tasks represent the highest level of work breakdown structure.
- Agreed units of estimate.
- The team leader guides the entire discussion, monitors time and prepares a structured document containing problem specification.

### 5.2.3 Individual Preparation

- Estimate for each task.
- Any assumption made by team members when creating the estimate.
- Any work related to project expenses should not be taken into account. It should be added to the Project Overhead.
- Individually generates a detailed WBS and documents the assumption made.
  - Every estimate should be based on effort, not calendar time.

→ **In our project:**

- After the kickoff meeting we have Estimated each of the tasks. • We should not take work related to project expenses
- We have individually generated a detailed WBS and documents the assumptions made.
- Our team has decided every estimate should be based on effort, no calendar time.

### 5.2.4 Estimation Session

- The moderator collects all estimates. Draw the estimated total on a line on the whiteboard and tabulate it.
- The estimator reads out clarifications and revisions to the list of tasks written on the estimator. Propose new or change tasks, discover hypotheses or questions.
- The team then discusses any doubt/problem they have about the tasks they have arrived at, assumptions made, and estimation issues.
- One effective way to resolve these disagreements is to talk about both sides of the issue, and then agree on an assumption that takes one of those sides.
- The moderator notes their feedback, to include it in the final estimation report
- The moderator notes their feedback, to include it in the final estimation report.

#### → In Our Project:

- Our team leader collects all estimates. Then, tabulate the estimated total on a line on the whiteboard.
- Our estimator(Team Members) reads out clarifications and revisions that the list of tasks are written. Propose new or change tasks, discover hypotheses or questions.
- The team members then discuss any doubt/problem they have about the tasks that we have made documents or assumption issues.
- We have used an effective way to resolve these disagreements by talking about both sides of the issue and then agreeing on an assumption that takes one of those sides.
- Then we take notes of all other member's feedback so that we can include it in the final estimation report

### 5.2.5 Assemble Tasks

- The project manager works with the moderator to gather all of the results from

the individual preparation and the estimation session.

- Removes redundancies and resolves remaining estimate differences to generate a final task list.
- The project manager should create a spreadsheet that lists the final estimates that each person came up with.
- An important part of the Delphi process is that it generates consensus among the team about the final schedule.
- Sometimes there may be a team member who disagrees with the team's assessment estimate for a task.
- During this meeting, the project manager must take the time to listen to everything that the person who differs from the rest of the team has to say.
- Spreadsheet should indicate the best-case and worst-case scenarios.

→ **In our project:**

- The team leader works with other members to gather all of the results from the individual preparation and the estimation session.
- Heremoves redundancies and resolves remaining estimate differences to generate a final task list.
- The project manager creates a spreadsheet that lists the final estimates that each person came up with.
- During this meeting, our team leader must take the time to listen to everything that the member who differs from the rest of the team has to say.
- We made a spreadsheet to indicate the best-case and worst-case scenarios at the end of tasks.

### 5.2.6 Review Result

- The project manager then assembles the result from the Estimation meeting
- He compiles the individual task list and the corresponding estimates into a master task list.
- He also compiles the individual list of assumptions.
- Once results are ready the Project Manager calls a final meeting to review the estimation result with the team.

→ **In our project:**

- Our team leader assembles the result from the Estimation meeting.
- He compiles the individual task list and the corresponding estimates into a master task list.
- He also compiles the individual list of assumptions.
- Once results are ready the team leader calls a final meeting to review the estimation result.

### **5.3 Individual Estimation Form**

Before generating the Estimation Form we have created a task list form to understand and confirm the final tasks that we will be working on. For that, each task has its estimated time given by the moderator. After that, based on that estimated time, team members have given their estimated time on the estimated form accordingly. So, the final form with the task list is:

## Task List

Task to achieve goal	Time
1. Research and Survey	3
2. Documentation	5
3. User Authentication	2
4. System Design	9
5. Performance Optimization	12
6. Risk Management	10
7. Maintenance and Updates	15
8. Deployment	2
<b>Calendar waiting times, delays</b>	
1. Approval of executing plans	7
2. Testing and debugging	8
<b>Project overhead tasks</b>	
1. Stakeholders meetings and feedback sessions	
2. Resource allocation and procurement	
3. Holidays on Occasions	

**Name:** Mehedi Hasan

**Date** 27/11/24

**Estimation form** 1/5

**Goal Statement:** To estimate the time to develop CSE Buddy **Units:** days



**Category:**      Goal Task      Quality Task      Waiting time      Project Overhead

WBS# or Priority	Task name	Est.	Delta	Delta 2	Delta 3	Delta	Total	Assumptions
1	Research and Survey			2				requires more time since field trips are included
	Documentation	5 -	+2	+2	+1	-	+1	requires more time since client discussions will be needed
	User Authentication	2	+1	-	-	-	+1	requires some time to identify the best user authentication
4	System Design		4	3	1		+15	requires a good deal of time since client interactions will be involved

	Performance Optimization	2	3	3	2		+20	requires plenty amount of time to optimize it perfectly
	Risk Management	0					+1	
	Maintenance & Updates	5	3	3	2	2	+1	
8	Deployment	+1+1--					+4	Additional required for safe

**Goal Statement:** To estimate the time to develop CSE Buddy     **Units:** days

**Category:**    ☒ Goal Task    ☒ Quality Task    ☐ Waiting time    ☐ Project Overhead

WBS	Task name	Est.	Delta 1	Delta 2	Delta 3	Delta 4	Total	Assumptions
1	Research and Survey		-1	+2	+3	+1	+8	Needs additional time to gather more information. (Delta +5)
2	Documentation		+2	-	+1	-1	+7	Needs additional time to focus more on document creation. (Delta +2)
3	User Authentication		-	+1	-	+2	+5	Needs additional time to conduct more surveys to understand user needs. (Delta +3)



4	System Design		-2	-	-	+3	+10	he increase in estimation (Delta +1) might indicate additional complexity discovered during the research and survey phase (Task 1).
5	Performance Optimization	2	-	-	-	-2	+10	his might not take that long. That's why (Delta-4)
6	Risk Management	0	-1	-2	-	+1	+8	here aren't many risks, so the estimate is reduced. (Delta -2)

7	Maintenance and Updates	5	+4	-3	+1	+4	+21	he system design (Task 4) might have become more complex, requiring more extensive maintenance efforts in the long run. (Delta +6)
8	Deployment		+1	-	-	-	+3	needs additional time to allow for a more thorough

								ployment process. (Delta +1)
9								
10								
	<b>Delta</b>		+3	-2	+5	+8		
	<b>Total</b>	8	61	59	64	72		

Name: Rukaiya Rahim Rumi

Date 27/11/24

Estimation form 1/5

WBS	Task name	Est.	Delta 1	Delta 2	Delta 3	Delta 4	Total	Assumptions
1	Research and Survey	3	+2	-	-	-	+5	Needs additional time to gather more information. (Delta +2)
2	Documentation	5	-2	+4	+6	-	+13	May need more time to organize information and data (Delta +8)
3	User authentication	2	-	-	-	+1	+3	The authentication process needs more time for the security

[illegible]

4	System Design		-3	+1	-	-	+7	might not need that much time to implement the design (Delta 2)
5	Performance Optimization	12	-2	-1	-	-	+9	this might not need so much time (Delta -3)
6	Risk Management	0	+4	-	-	-	+14	for the safety more focus should be put on the risk analysis (Delta 4)
7	Maintenance and Updates	15	-	-	+2	-	+17	because of the complexity, it might take much more time (Delta +2)
8	Deployment	2	-	+1			+3	can take a little bit more time (Delta +1)
9								
10								
	<b>Delta</b>		5	8	1			

			57	62	70	1		
		Total						

Name: Rabbi Islam Yeasin

Date 27/11/24

Estimation form 1/5

Goal Statement: To estimate the time to develop CSE Buddy      Units: days

Category:    ☒ Goal Task    ☒ Quality Task    ☐ Waiting time    ☐ Project Overhead

WBS	Task name	Est.	Delta 1	Delta 2	Delta 3	Delta 4	Total	Assumptions
1	Research and Survey	3	+1	-	-	-	+4	Need more additional time to gain information. (Delta +1)
2	Documentation	5	-1	+2	+1	-	+7	Need more time to make a combined and structural documentation. (Delta +2)
3	User authentication	2	-	+1	-	-	+3	The authentication process needs more time to develop. (Delta 1)

4	System Design	9	-1	-	+2	-	+10	he total system design needs more time to implement with the designing team. (+1)
5	Performance Optimization	12	-3	+2	-	-	+11	we are able to maintain the optimization process well in the beginning then it does not require that much time. (Delta -1)
6	Risk Management	10	+2	-3	-	-	+9	here won't be that much of an issue when it comes with the risks of our project. (Delta )
7	Maintenance and Updates	15	+3	-2	-	+1	+17	requires more real time basis work with the help of the maintenance team. (Delta +2)

8	deployment	2	+1	-	-	-	+3	for the project deployment work we have to take more time to extract it with the deployability. (Delta +1)
9								
10								
	Delta		+2	0	+3	+1		
	Total	58	60	60	63	64		

Name: Aftab Ali Snigdho

Date 27/11/24

Estimation form 1/5

Goal Statement: To estimate the time to develop CSE Buddy      Units: days

Category: ☒ Goal Task ☒ Quality Task ☐ Waiting time ☐ Project Overhead

WBS	Task name	st.	delta 1	delta 2	delta 3	delta 4	Total	assumptions
-----	-----------	-----	---------	---------	---------	---------	-------	-------------

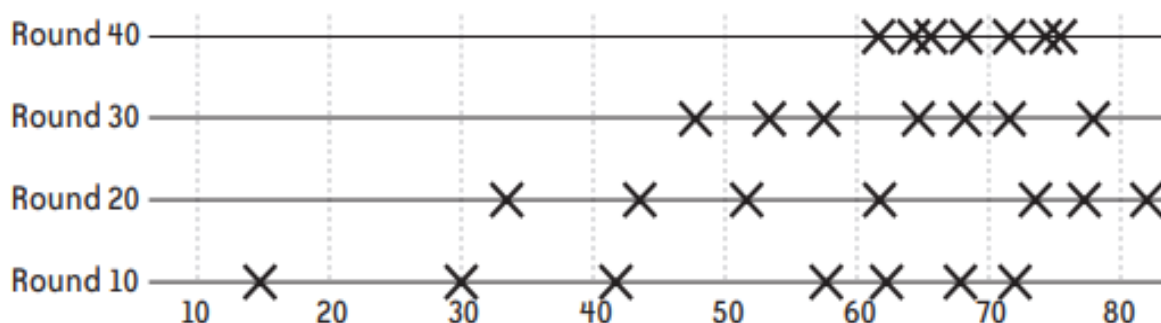
1	Research and Survey		+2	-1	-	-	+4	nce research and survey is a important part it needs more time(Delta +1)
2	Documentation		-1	+2	-	-	+6	we need proper documentation. That's why we need time(Delta +1)
3	User Authentication		+3	+1	+2	-	+8	needs authentication in various ways. (Delta +6)

4	System Design		+2	+3	-1		+13	needs proper design which needs more time(Delta +4)
5	Performance Optimization	2	+3	+1	-2	+1	+15	needs more time for performance optimization(delta +3)
6	Risk Management	0	-4	+2			+8	Risk management is a crucial part of a project. But here don't need more time (Delta -2)



7	Maintenance and Updates	5	-2	-1	+1		+13	on't need so much time in that task( Delta )
8	Deployment		+1				+3	ttle extra time may need in that task (Delta +1)
9								
10								
	Delta	4	7		1			
	Total	8	2	9	9	0		

## 5.4 Estimation session



5 Assemble Table

**Goal Statement:** To estimate the time to develop CSE Buddy

**Estimator:** Mehedi Hasan,Md. Abu Bakar Siddik, Rukaiya Rahim Rumi, Rabbi Islam Yeasin,Aftab Ali Snigdho  
**Unit:** Working Day

WBS	Task Name	A	A	TI	T	R	Best Case	Worst Case	Avg. h-low working day	Assumptions
1	Research & Survey	5	4	8	4	6	4	8	5.4	relevant resource for research and surveys are available.
2	Documentation	13	7	7	6	10	6	13	8.6	Access to necessary templates and guidelines for documentation.

3	ser uthentication	3	3	5	8	3	3	8	4.4	lear requiremen for user authentication are provided.
---	----------------------	---	---	---	---	---	---	---	-----	--

	ystem Design	7	10	10	13	15	7	15	11	availability of experienced personnel for system architecture and design.
	erformance Optimization	9	11	10	15	0	9	20	13	erformance benchmarks an criteria are clearly defined.
	isk anagement	14	9	8	8	10	8	14	9.8	omprehensive risk assessment has been conducted.
	aintenance & Update	17	17	21	13	5	13	25	18.6	ccess to necessary resources for updating and maintaining the system.

	ployment	3	3	3	3	4	3	4	3.2	ployment environment is properly configured and ready.
	total	71	4	72	0	3	53	107	74	

# **Chapter 6 Risk Management**

## **1 What is Risk Management?**

Risk management encompasses the identification, analysis, and response to risk factors that occur in a business's life. Effective risk management means attempting to control future outcomes as much as possible by acting proactively rather than reactively. Therefore, effective risk management offers the potential to reduce both the possibility of a risk occurring and its potential impact.

## **2 Risk Mitigation, Monitoring, and Management (RMMM)**

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.

### **2.1 Risk Mitigation**

Risk Mitigation is a technique for avoiding risks (Risk Avoidance). The following are steps to take to reduce the risks:

Identifying the risk and getting rid of the causes that lead to the production of risk

Controlling the relevant documents regularly.

Conducting timely reviews to accelerate the process.

### **2.2 Risk Monitoring**

Risk monitoring is an activity used to track a project's progress. The following are the critical goals of the task.

- See if the risks that were anticipated actually happen.

- Verify that the risk aversion steps defined for risk are adequately implemented.

- Gather information for future risk assessments.

- Determine which risks generate which problems throughout the project.

## **2.3 Risk Management and Planning**

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.

### 3 Risk Plan

<b>Risk plan for project</b>	<b>SE BUDDY : An EduTech platform</b>
<b>Assessment team members</b>	Abdullah, Rezoan, Rumi, Rabby, Mehedi, Snigdho

<b>Risk</b>	<b>Probability</b>	<b>Impact</b>	<b>Priority</b>	<b>Action</b>
Server crashes	4	5	20	Implement a disaster recovery plan and regularly back up data.
Financial constraints	3	5	15	Explore potential funding sources, such as grants, donations, or partnerships.
Security breaches	3	5	15	Implement robust security measures and conduct regular security audits.
Plagiarism of content	3	3	9	Implement plagiarism detection tools and educate students on academic integrity.

copyright infringement	2	5	10	develop clear copyright guidelines and obtain proper licenses for used materials.
------------------------	---	---	----	---

4 Individual Risk Information Sheet

Task 1 : Server crashes (SC-1)

Risk Information Sheet			
Risk ID: SC-1	Date: 5-3-24	Prob: 4	Impact: 5
<p><b>Description:</b></p> <p>Server crashes (SC-1) pose a high threat due to unforeseen technical issues. These crashes can cripple access to the platform, leading to downtime and data loss. This disrupts user experience, hinders learning progress, and damage the platform's reputation, potentially hindering adoption.</p>			



**Refinement/ Context:**

Since the probability is high,

Unforeseen technical issues can cause significant downtime and data loss,

Disrupting user experience, hindering student progress, and damaging platform reputation.

Regular backups, maintenance, and exploring cloud-based solutions are crucial for mitigation.

This risk is especially critical in the project's initial stages due

to potential unforeseen issues and the reliance on continuous platform availability for learning activities.

**Mitigation/ Monitoring:**

Track the frequency and duration of server downtime (IT Team).

Conduct regular reviews of the disaster recovery plan (Project Manager).

- Test the effectiveness of backup and recovery procedures (IT Team)

## Management/Contingency Plan/Trigger:

### Management:

**Implement a disaster recovery plan:** This plan should outline procedures for data backup, restoration, and platform recovery in case of a server crash.

**Conduct regular system maintenance:** This includes software updates, hardware checks, and performance monitoring to identify and address potential issues before they escalate.

**Invest in redundant hardware or cloud-based solutions:** Consider implementing a hybrid approach, utilizing on-premise servers with cloud backups for increased scalability and reliability.

### Contingency plan:

In the event of a server crash, immediately activate the disaster recovery plan.

Isolate the crashed server to prevent further damage and data loss.

Restore data from the latest backup and bring the platform back online as quickly as possible.

Communicate the outage and estimated recovery time to users through appropriate channels.

### Trigger:

Server unresponsiveness or any hardware/ software malfunction leading to platform downtime.

**Current Status:**

- Risk Assessment Completed: Yes

Action Plan Drafted: Yes

Monitoring and Review Plan Defined: Yes

The risk of server crashes has been identified and assessed as a medium probability with a high impact on the CSE Buddy project.

An action plan has been drafted to mitigate the risk.

This includes:

- Implementing a disaster recovery plan with regular data backups.
- Conducting regular system maintenance and upgrades.
- Investing in redundant hardware or cloud-based solutions.

A monitoring and review plan has been outlined to track the effectiveness of the implemented actions. This involves:

- Tracking the frequency and duration of server downtime.
- Conducting regular reviews of the disaster recovery plan.
- Testing the effectiveness of backup and recovery procedures.

Next Steps:

- Finalize and implement the action plan.
- Communicate the risk and mitigation plan to relevant stakeholders.
- Establish a schedule for monitoring and reviewing the effectiveness of the implemented actions.

<b>originator:</b> Rezoan Ahmed	<b>ssigned:</b> IT team
---------------------------------	-------------------------

## Risk 2 : Financial constraints (FC-2)

Risk Information Sheet			
Risk ID: FC-2	Date: 5-3-24	Prob: 3	Impact: 5
<p><b>Description:</b></p> <p>Financial constraints (FC-2) are limitations that restrict your ability to take certain financial actions. These limitations can arise from various factors and significantly impact your financial decisions and overall well-being.</p>			
<p><b>Refinement/ Context:</b></p> <p>While the probability is moderate,</p> <p style="padding-left: 40px;">Building the platform, including web development, design, and potential licensing fees for specific tools, can be still expensive. Developing high-quality educational content in Bangla requires subject matter experts, educators, and/or translators, and their fees can be significant.</p> <p style="padding-left: 40px;">Reaching the target audience and raising awareness about the platform might involve marketing costs, such as online advertising or social media campaigns.</p> <p>Relying solely on initial funding or donations might not be sustainable in the long continuous platform availability for learning activities.</p>			

**litigation/ Monitoring:**

Offering a freemium model with limited free features and premium tiers with additional functionalities could provide some revenue stream.

Seeking support from educational institutions, NGOs, or corporations interested in promoting education might help offset costs.

Engaging volunteers with relevant expertise in content creation, translation, or development can significantly reduce costs.

## Management/Contingency Plan/Trigger:

### Management:

**Identifying potential limitations:** Analyze the financial needs at each stage, including development, content creation, launch, and ongoing maintenance.

**Implementing strategies to mitigate them:** Focus on developing core functionalities first, ensuring they meet the immediate needs of the target audience. This helps manage initial costs and allows for phased development based on available resources.

### Contingency plan:

Inform stakeholders about potential financial challenges and the developed contingency plans to ensure transparency and maintain trust.

Regularly monitor your financial situation, track expenses, and identify early signs of potential constraints.

Define clear trigger points, such as reaching a specific funding threshold or encountering unexpected expenses, that signal the need to activate specific contingency plans.

Regularly reassess the situation and adapt your plan based on the evolving circumstances and effectiveness of implemented strategies.

### Trigger:

Poor allocation of resources, such as time, personnel, or budget, can lead to increased costs and delays, impacting financial stability.

**Current Status:**

Risk Assessment Completed: Yes

Action Plan Drafted: Yes

Monitoring and Review Plan Defined: Yes

The risk of financial constraints has been identified and assessed as a lower probability with a high impact on the CSE Buddy project.

An action plan has been drafted to mitigate the risk. This includes:

- Identifying potential financial challenges and creating response strategies for each scenario
- Continuously monitor the financial situation and be prepared to adapt strategies based on changing circumstances
- Inform stakeholders about financial needs and challenges transparently, fostering trust and potentially attracting additional support.

A monitoring and review plan has been outlined to track the effectiveness of the implemented actions. This involves:

- Keeping contingency plans prepared based on the financial challenges that can occur
- Regularly monitor the financial situation and keeping tracks of expenses
- Regularly reassessing the situation and adapting the plan based on the evolving circumstances



Next Steps:

Finalize and implement the action plan.

Communicate the risk and mitigation plan to relevant stakeholders.

Establish a schedule for monitoring and reviewing the effectiveness of the implemented actions.

**Originator:** Md. Abu Bakar Siddik

**Assigned:** Financial team

### Risk 3 : Security Breaches (SB - 3)

Risk Information Sheet			
<b>Risk ID:</b> SB-3	<b>Date:</b> 5-3-24	<b>Prob:</b> 3	<b>Impact:</b> 5
<p><b>Description:</b></p> <p>Security breaches (SB-3) pose a moderate threat due to potential vulnerabilities in the platform's security infrastructure. These breaches can compromise user data, leading to privacy concerns, identity theft, and legal ramifications. Additionally, unauthorized access to sensitive information undermines trust in the platform and may result in financial losses or damage to the organization's reputation.</p>			

**Refinement/ Context:**

While the probability of security breaches is moderate, the potential impact is significant.

Potential vulnerabilities in the security infrastructure may lead to unauthorized access and data breaches, resulting in privacy violations and legal consequences.

Addressing security gaps through robust encryption protocols, access controls, and regular security audits is essential for safeguarding user data and maintaining trust in the platform.

**Litigation/ Monitoring:**

Implement stringent encryption protocols to protect sensitive user data during transmission and storage (IT Team).

Enforce strong access controls to restrict unauthorized access to sensitive information (IT Team).

Conduct regular security audits to identify and address potential vulnerabilities in the platform's infrastructure (Security Team).

## **Management/Contingency Plan/Trigger:**

### **Management:**

Implement robust encryption protocols: Utilize industry-standard encryption algorithms to safeguard sensitive user data from unauthorized access during transmission and storage.

Enforce strong access controls: Implement multi-factor authentication and role-based access controls to restrict access to sensitive information based on user roles and permissions.

Conduct regular security audits: Perform comprehensive security assessments to identify and mitigate potential vulnerabilities in the platform's infrastructure.

### **Contingency plan:**

In the event of a security breach, immediately activate the incident response plan.

Isolate affected systems to prevent further unauthorized access and data exfiltration.

Notify relevant stakeholders, including users, regulatory authorities, and law enforcement, about the security breach and take appropriate measures to mitigate its impact.

### **Trigger:**

Detection of unauthorized access, suspicious activity, or data breaches within the platform's infrastructure.

**Current Status:**

Risk Assessment Completed: No

Action Plan Drafted: No

Monitoring and Review Plan Defined: No

The risk of security breaches has been identified and assessed as a moderate probability with a significant impact on the CSE Buddy project.

An action plan needs to be drafted to mitigate the risk, including

Implementing robust encryption protocols

Enforcing strong access controls 3. Conducting regular security audits

A monitoring and review plan should be defined to track the effectiveness of the implemented actions, including monitoring for unauthorized access and conducting periodic security assessments.

**Next Steps:**

Initiate a risk assessment to determine the probability and impact of security breaches.

Draft an action plan outlining mitigation strategies, including encryption protocols, access controls, and security audits.

Define a monitoring and review plan to track the effectiveness of the implemented security measures and ensure continuous improvement.

**Originator:** Mehedi Hasan

**Assigned:** Security Team

Task 4 : Plagiarism of content (PC-4)

**Risk Information Sheet**

<b>Risk ID: PC-4</b>	<b>Date: 5-3-24</b>	<b>Prob: 3</b>	<b>Impact: 3</b>
<b>Description:</b> <p>Plagiarism of content (PC-4) poses a moderate threat due to the potential for unauthorized copying or reproduction of educational materials within the platform. This could lead to academic dishonesty, loss of credibility, and legal ramifications. Additionally, plagiarism undermines the integrity of the learning environment and diminishes the value of original content.</p>			
<b>Refinement/ Context:</b> <p>While the probability of content plagiarism is moderate, the potential impact is significant.</p> <p>Unauthorized copying or reproduction of educational materials may result in academic penalties for students, tarnishing the reputation of the platform and its associated institutions.</p> <p>Implementing robust plagiarism detection mechanisms, fostering a culture of academic integrity, and providing guidance on proper citation practices are essential for preventing content plagiarism and upholding academic standards.</p>			
<b>Litigation/ Monitoring:</b> <p>Implement robust plagiarism detection tools to identify instances of content plagiarism (Content Moderation Team).</p> <p>Educate users on the importance of academic integrity and proper citation practices (Education Team).</p> <p>Monitor user-generated content for signs of plagiarism and take appropriate action (Content Moderation Team).</p>			

### Management/Contingency Plan/Trigger:

#### Management:

Implement robust plagiarism detection tools: Utilize advanced algorithms to scan user-generated content and identify potential instances of plagiarism. Educate users on academic integrity: Provide guidelines and resources on proper citation practices and discourage plagiarism through educational initiatives.

Monitor user-generated content: Regularly review and assess user-submitted materials for signs of plagiarism and take prompt action to address any violations.

#### Contingency plan:

In the event of confirmed plagiarism, take immediate action to remove the infringing content and notify the relevant stakeholders.

Provide support and guidance to users involved in plagiarism incidents, emphasizing the importance of academic integrity and proper citation practices.

Implement measures to prevent future occurrences of plagiarism, such as enhancing plagiarism detection algorithms and strengthening user education efforts.

#### Trigger:

Detection of suspicious similarities or unauthorized use of copyrighted material within user-generated content.

#### Current Status:

Risk Assessment Completed: No

Action Plan Drafted: No

Monitoring and Review Plan Defined: No

The risk of plagiarism of content has been identified and assessed as a

moderate probability with a significant impact on the CSE Buddy project.

An action plan needs to be drafted to mitigate the risk, including

Implementing robust plagiarism detection tools 2. Educating users on academic integrity

3. Monitoring user-generated content.

A monitoring and review plan should be defined to track the effectiveness of the implemented actions, including monitoring for instances of plagiarism and assessing user compliance with citation practices.

#### Next Steps:

Initiate a risk assessment to determine the probability and impact of content plagiarism.

Draft an action plan outlining mitigation strategies, including plagiarism detection tools, user education initiatives, and content moderation procedures

Define a monitoring and review plan to track the effectiveness of the implemented measures and ensure adherence to academic integrity standards.

**Originator:** Rabbi Islam Yeasin

**Assigned:** Content Moderation Team

**Risk 5: Copyright Infringement ( CI - 5 )**

Risk Information Sheet			
Risk ID: CI - 5	Date: 5-3-24	Prob: 2	Impact: 5



**Description:**

Copyright infringement (CI-5) poses a substantial risk of legal action and financial burdens due to the unauthorized usage of protected content. Such actions can lead to costly litigation, significant fines, and damage to the platform's reputation. The unauthorized use of copyrighted materials can occur inadvertently through oversight, necessitating stringent content review processes and copyright education for staff. Ensuring compliance is critical to maintaining the platform's integrity and avoiding the high costs associated with copyright disputes.

**Refinement/Context:**

Given the moderate probability, the use of copyrighted materials without proper licensing or acknowledgment can occur inadvertently due to oversight or misunderstanding of copyright law.

Moderate probability of inadvertent copyright infringement due to oversight or misunderstanding of copyright laws.

Essential to educate the team on copyright norms and ensure thorough review of content for compliance.

Risk is particularly relevant during content development phases, where external resources are integrated into the platform.

Necessitates the implementation of a comprehensive content review process to avoid unauthorized use of copyrighted materials.

## **Litigation/Monitoring :**

Legal Team to establish guidelines for using copyrighted materials, including obtaining permissions and licenses.

Training Manager to organize educational programs for the content development team to understand and adhere to copyright norms.

Compliance Officer to conduct periodic checks on all platform content to identify and rectify potential copyright breaches.

Implement a structured process for reviewing content for copyright compliance before publication or use on the platform.

Where feasible, form relationships with copyright owners to facilitate legal use of materials and explore licensing opportunities.

## **Management/Contingency Plan/Trigger:**

### **Management:**

Develop clear guidelines for the use of copyrighted materials, detailing procedures for obtaining permissions and licenses.

Set up a systematic approach where all content is meticulously reviewed for copyright compliance before its release or utilization.

Establish Partnerships with Copyright Holders: Formulate relationships with content creators to legally use their materials, potentially through agreements that benefit both parties.

### **Contingency plan:**

Engage Legal Team Immediately upon Infringement Claim: Activate the legal team at the first sign of a copyright infringement claim to assess and respond appropriately.

Cease Use of Disputed Material: Temporarily halt the use of material under dispute until the issue is fully resolved.

Negotiate Settlements or Licensing if Infringement Occurs: Work towards a settlement or secure licensing agreements for the disputed content to mitigate legal repercussions.

Review and Update Compliance Procedures: Following an infringement incident, thoroughly review and refine compliance

procedures to prevent future occurrences.

**Trigger:**

Receipt of a Copyright Infringement Claim: Any formal notification or claim of copyright infringement acts as a trigger for the contingency plan.

Identification of Potential Infringement during Audits: Discovery of content that may infringe on copyright during routine audits also serves as a trigger for immediate review and action.

**Current Status:**

Risk Assessment Completed: No

Action Plan Drafted: No

Monitoring and Review Plan Defined: No

The risk of Copyright Infringement has been identified and assessed as a moderate probability with a significant impact on the CSE Buddy project.

An action plan needs to be drafted to mitigate the risk, including

- Implement the copyright compliance policy

- Stakeholder Communication

- Launch Monitoring Efforts

**Next Steps:**

Complete the finalization of the action plan and proceed with its implementation across the relevant sectors of the project.

Ensure that all parties involved are informed about the identified risks and the strategies devised to address them.

Set up a systematic schedule to regularly monitor and review the impact of the actions taken, ensuring continuous improvement and adherence to copyright compliance.

**Originator:** Rukaiya Rahim Rumi

**Assigned:** Security Team

Let's move forward to the most crucial part of the project.

# Chapter 7 Knowledge Sharing in Software Project Management

## 7.1 Importance of Knowledge Sharing

Knowledge sharing in software engineering is essential for process improvement, innovation, and problem-solving. It enables organizations to improve productivity, ensure consistency, and enhance software quality by leveraging collective expertise. Effective knowledge sharing minimizes redundancy, improves decision-making, and allows teams to build upon prior knowledge, ensuring faster issue resolution and optimized workflows.

## 7.2 Methods of Knowledge Sharing

### Documentation & Wikis:

- Maintains a structured repository for knowledge storage and easy retrieval.
- Ensures that both new and existing team members can access critical project-related information quickly.
- Includes technical documentation, process guidelines, and coding standards.
- Code Reviews & Pair Programming:
  - Helps developers exchange knowledge by reviewing each other's code, improving software quality.
  - Pair programming allows a senior and junior developer to collaborate, enabling real-time learning and knowledge transfer.
  - Code reviews catch defects early, ensuring adherence to coding best practices.
- Meetings & Retrospectives:
  - Regular team meetings facilitate open discussions about project progress, challenges, and solutions.
  - Sprint retrospectives help in analyzing past mistakes and implementing process improvements.
- Encourages feedback loops to ensure continuous enhancement of software processes.

- Training & Mentorship Programs:
- Organizes formal training sessions to upskill employees in specific technologies and methodologies.
- Mentorship programs pair experienced professionals with junior team members for guidance.
- Encourages peer learning and organizational knowledge retention.
- Version Control & Issue Tracking Systems:
- Enables transparent collaboration by maintaining logs of changes and issues.
- Provides clear traceability of modifications, facilitating effective debugging and software evolution.
- Tools like Git, Jira, and Confluence help track project progress efficiently.

### **7.3 Benefits of Knowledge Sharing**

- Reduces redundancy by preventing duplication of effort and ensuring efficient problem-solving.
- Enhances software quality through collective expertise and early defect detection.
- Promotes innovation by fostering collaboration and cross-functional knowledge exchange.
- Ensures process standardization in software development by incorporating best practices.
- Improves team collaboration and productivity, leading to optimized project delivery.

# Chapter 8 Capability Maturity Model Integration (CMMI)

## 8.1 Overview of CMMI

CMMI is a **framework for software process improvement** developed by the **Software Engineering Institute (SEI)** at Carnegie Mellon University. It provides a structured methodology to improve software development by defining maturity levels, ensuring process efficiency, and reducing project risks.

## 8.2 Maturity Levels of CMMI

1. **Level 1 - Initial:**
  - i. Processes are unpredictable and chaotic, often leading to project failure.
  - ii. Success depends on individual efforts rather than defined processes.
2. **Level 2 - Managed:**
  - i. Basic project management practices are established.
  - ii. Processes are repeatable, but variations still exist across projects.
  - iii. Project tracking and oversight are introduced.
3. **Level 3 - Defined:**
  - i. Organization-wide standardization of processes ensures consistency and efficiency.
  - ii. Tailored processes are established, improving coordination between teams.
  - iii. Emphasizes process documentation and institutionalization.
4. **Level 4 - Quantitatively Managed:**
  - i. Decisions are data-driven, based on measured performance and statistical process control.
  - ii. Quantitative techniques are applied to improve project predictability and minimize defects.
  - iii. Metrics-based decision-making enhances process efficiency.
5. **Level 5 - Optimizing:**
  - i. Continuous process improvement through innovation and automation.
  - ii. Proactive identification of weaknesses and improvements.
  - iii. Focuses on agility, adaptability, and emerging technological integration.

## 8.3 Benefits of CMMI Implementation

- **Improves software quality** and reliability by following structured process guidelines.
- **Enhances process efficiency** by reducing defects and improving workflow management.
- **Ensures compliance** with industry standards and regulatory frameworks.
- **Facilitates risk management** in project execution by using data-driven insights.
- **Increases customer satisfaction** by delivering high-quality products on time and within budget.
- **Encourages organizational learning and process innovation** through iterative improvement cycles.

## Chapter 9 Role of a Project Manager at CMMI Levels 3, 4, and 5

### 9.1 Role at Level 3 (Defined)

At this level, organizations establish **standardized processes** across projects. The project manager ensures adherence to these standards and fosters process consistency.

#### Key Responsibilities:

- Implement **organization-wide standard operating procedures (SOPs)** to streamline workflows.
- Establish **knowledge-sharing frameworks** to ensure cross-team collaboration and learning.
- Ensure **process documentation and compliance** with organizational standards.
- Conduct **software reviews, retrospectives, and process improvement initiatives** to enhance efficiency.
- Facilitate **training and mentorship programs** to upskill project teams and promote best practices.

### 9.2 Role at Level 4 (Quantitatively Managed)

This level emphasizes **data-driven project management**. Project managers use **metrics and statistical data** to optimize project execution and ensure continuous performance improvement.

#### Key Responsibilities:

- Implement **performance metrics and statistical process controls** to track project efficiency.
- Use **historical data and predictive analysis** for accurate project forecasting.
- Conduct **trend analysis** to identify process inefficiencies and areas for improvement.
- Drive **data-driven decision-making** by relying on objective insights and performance indicators.
- Foster **risk management strategies** to proactively mitigate potential issues.
- Collaborate with **cross-functional teams** to align business goals with software development processes.

### 9.3 Role at Level 5 (Optimizing)

At this level, organizations focus on **continuous process improvement** and **innovation**. The project manager leads efforts to **adopt emerging technologies**, optimize workflows, and enhance organizational agility.

#### Key Responsibilities:

- Promote **continuous process improvement (CPI) initiatives** to refine workflows and boost efficiency.



- Encourage the adoption of **automation, AI-driven tools, and DevOps practices** to enhance software development cycles.
- Lead **Agile and DevOps transformations** to foster iterative development and continuous delivery.
- Conduct **post-mortem reviews and root cause analysis** to identify lessons learned and drive future improvements.
- Implement **adaptive project management methodologies** to align with evolving business needs.
- Drive **innovation and technology adoption** to keep the organization competitive in a rapidly changing market.