Software Risk Management Plan

1. Introduction

Software projects are inherently exposed to various uncertainties that may negatively impact their successful completion. These uncertainties, commonly referred to as risks, can arise from technical challenges, resource limitations, scheduling constraints, and external dependencies. The purpose of this Software Risk Management Plan is to systematically identify, analyze, mitigate, and monitor potential risks associated with the development of the Real-Time Collaborative Online Integrated Development Environment (RCO-IDE). Effective risk management ensures project stability, quality, and timely delivery.

2. Purpose of Risk Management

The primary purpose of risk management is to minimize the likelihood and impact of adverse events that may affect the project. By proactively addressing risks, the development team can reduce rework, cost overruns, and schedule delays. This document serves as a guideline for identifying risks early in the project lifecycle and implementing appropriate mitigation strategies.

3. Risk Identification

Risk identification involves recognizing potential threats that could hinder project progress or system performance. The major categories of risks identified for this project include:

• Technical Risks: Challenges related to real-time synchronization, WebSocket latency, sandboxed code execution, and system scalability.

• Schedule Risks: Delays caused by underestimated development effort, integration complexities, or academic deadlines.

• Resource Risks: Limited availability of skilled developers, hardware constraints, or cloud infrastructure limitations.

• Security Risks: Vulnerabilities related to remote code execution, data breaches, and unauthorized access.

• Dependency Risks: Reliance on third-party services such as Judge0 API and cloud hosting platforms.

4. Risk Analysis

Risk analysis evaluates each identified risk based on its probability of occurrence and potential impact on the project. Risks are categorized as low, medium, or high.

High-impact risks include security vulnerabilities and failures in real-time collaboration, as these can severely affect system reliability and user trust. Medium-impact risks involve performance degradation under load and delays in feature implementation. Low-impact risks include minor UI issues or temporary service interruptions.

5. Risk Mitigation Strategies

Risk mitigation focuses on reducing the probability or impact of identified risks through planned actions:

• Technical Risks Mitigation: Use proven frameworks, conduct regular code reviews, and perform extensive testing for real-time features.

• Schedule Risk Mitigation: Adopt agile development practices, define milestones, and regularly track progress.

• Resource Risk Mitigation: Ensure proper task allocation and maintain documentation to reduce dependency on individuals.

• Security Risk Mitigation: Implement strict sandboxing, authentication, encryption, and regular security audits.

• Dependency Risk Mitigation: Monitor third-party services and maintain fallback or alternative solutions where possible.

6. Risk Monitoring and Control

Risk monitoring is a continuous process throughout the project lifecycle. Identified risks are reviewed periodically to assess their current status. New risks are documented as they arise, and mitigation plans are updated accordingly. Regular

project meetings and progress reviews are used to track risk indicators and ensure timely corrective actions.

7. Roles and Responsibilities

The project manager is responsible for overseeing risk management activities and ensuring mitigation strategies are implemented. Development team members are responsible for reporting potential risks and adhering to mitigation measures. Stakeholders are informed about critical risks and their potential impact on project outcomes.

8. Conclusion

An effective risk management plan is essential for the successful development of the Real-Time Collaborative Online IDE. By identifying, analyzing, and mitigating risks proactively, the project team can enhance system reliability, security, and performance. Continuous monitoring ensures that risks are controlled throughout the project lifecycle, contributing to the successful completion of the academic project.