

**Fr. Conceicao Rodrigues College of Engineering, Mumbai**  
**SOFTWARE ENGINEERING (CSC601)**

**Assignment -II**

**Date: 17-10-23**

**CO5:** Identify risks, manage the change to assure quality in software projects.

**Assignment 2**

1. What is risk assessment in the context of software projects, and why is it essential?
2. Explain the concept of software configuration management and its role in ensuring project quality.
3. How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?
4. Describe the process of conducting a formal walkthrough for a software project.
5. Why is it important to consider software reliability when analyzing potential risks in a project?

**Rubrics :**

Indicator	Average	Good	Excellent	Marks
<b>Organization (2)</b>	Readable with some mistakes and structured (1)	Readable with some mistakes and structured (1)	Very well written and structured (2)	
<b>Level of content(4)</b>	Minimal topics are covered with limited information (2)	Limited major topics with minor details are presented (3)	All major topics with minor details are covered (4)	
<b>Depth and breadth of discussion(4)</b>	Minimal points with missing information (1)	Relatively more points with information (2)	All points with in depth information (4)	
<b>Total Marks(10)</b>				

## Assignment-2 (Software Engineering)

mohtashim Ali

9644(Te-Comps-B)

### 1. Risk Assessment in Software Projects:

Risk assessment in software projects is a systematic process of identifying, analyzing, and prioritizing potential risks that could affect the project's success. This process is essential for several reasons:

a. Proactive Risk management: Risk assessment allows project managers and teams to proactively identify potential issues and challenges that may arise during the software development process. By identifying these risks early, the project team can develop mitigation strategies and contingency plans to address them effectively.

b. Resource Allocation: By assessing risks, project managers can allocate resources, both in terms of personnel and budget, more effectively. They can allocate resources to high-priority risks and ensure that the project remains on track.

c. **Schedule and Budget Control:** Understanding the risks associated with a project helps in creating realistic schedules and budgets. It prevents unrealistic expectations and helps in avoiding project delays and cost overruns.

d. **Stakeholder Communication:** It facilitates effective communication with stakeholders. By identifying and addressing risks, project managers can communicate potential challenges and how they plan to mitigate them, building trust with project sponsors and clients.

e. **Quality Assurance:** Identifying risks related to software quality and reliability early in the project can lead to improved software quality. This is because potential issues can be addressed before they escalate into significant defects.

2. **Software Configuration management (SCM):**  
Software Configuration management (SCM) is a discipline that focuses on controlling and tracking changes to software artifacts throughout the software development lifecycle. SCM plays a vital

role in ensuring project quality for the following reasons:

a. Version Control: SCM provides version control, ensuring that different versions of the software and its components are well-managed. This prevents confusion and helps in tracking the evolution of the software.

b. Change Tracking: It allows for the tracking of changes made to the software. This ensures that all changes are documented and can be traced back, which is crucial for understanding the evolution of the project and identifying issues.

c. Configuration Consistency: SCM ensures that the software is built from a consistent and well-defined configuration. This reduces the risk of issues arising from discrepancies in software components.

d. Security and Compliance: SCM helps in ensuring the security and compliance of the software by controlling who can make changes and what changes are permitted. This is essential for



maintaining the integrity of the software.

e. Quality Assurance: By managing changes and configurations effectively, SCM contributes to software quality by preventing unauthorized or untested changes, thus reducing the risk of introducing defects into the software.

### 3. Formal Technical Reviews (FTR):

Formal Technical Reviews (FTR) are a structured and systematic approach to examining and evaluating software work products such as requirements, designs, and code. FTR plays a critical role in ensuring software quality and reliability in the following ways:

a. Defect Detection: FTRs are designed to identify defects and issues in software artifacts. This helps in catching errors early in the development process when they are less costly to fix.

b. Adherence to Standards: FTRs ensure that the software work products adhere to coding

standards, design principles, and project requirements. This consistency is essential for maintaining software quality.

c. Knowledge Sharing: FTRs encourage collaboration and knowledge sharing among team members. They provide a forum for team members to discuss the software and share their insights and expertise, leading to improved software quality.

d. Quality Improvement: FTRs result in corrections and improvements to the software artifacts. This iterative process enhances the quality and reliability of the software.

e. Risk Reduction: By systematically reviewing and addressing potential issues, FTRs help in reducing the risk of defects and failures in the software. This is vital for ensuring the reliability of the final product.

4. Formal Walkthrough for Software Projects:  
A formal walkthrough is a structured process for reviewing and evaluating software documents, code,

or designs. The process involves several key steps:

a. Preparation: The person responsible for the walkthrough prepares the materials to be reviewed. This may include code, design documents, or requirements specifications.

b. Walkthrough meeting: The review team, typically consisting of developers, testers, and other stakeholders, convenes to discuss the materials. The presenter explains the content, and participants ask questions, offer suggestions, and identify potential issues.

c. Issue Identification: During the meeting, the participants identify defects, inconsistencies, and areas for improvement. These are documented for later correction.

d. Correction and Follow-up: After the walkthrough, the issues identified are addressed, and necessary corrections are made to the software work product. Follow-up activities ensure that the identified issues have been resolved.

e. Documentation: The results of the formal walkthrough, including identified issues and resolutions, are documented for reference and audit purposes.

Formal walkthroughs help improve software quality by providing a structured and collaborative approach to identifying and rectifying issues in software work products.

### 5. Importance of Software Reliability in Risk Analysis:

Software reliability is a critical factor in risk analysis for software projects. Here's why it's important:

a. Customer Satisfaction: Reliable software ensures a positive user experience. Unreliable software can lead to user frustration and dissatisfaction, potentially harming the project's reputation and success.

b. Cost management: Unreliable software often leads to post-release defects, which can be



expensive to fix. Analyzing reliability risks allows project managers to allocate resources for testing and quality assurance, reducing the cost of fixing defects after release.

c. Project Delays: Unreliable software may cause project delays as defects are discovered and rectified late in the development process. Analyzing reliability risks early can prevent such delays.

d. Risk mitigation: By identifying potential reliability risks, the project team can proactively address them through testing, code reviews, and other quality assurance measures, reducing the likelihood of reliability issues in the final product.

e. Compliance and Safety: In certain applications, like medical devices or autonomous vehicles, software reliability is a matter of compliance and safety. Analyzing reliability risks is essential to meet regulatory requirements and ensure user safety.