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

John Rajan George Roll No: 9611 TE Comps B

SE Assignment 2

1) Risk assessment in the context of software projects is the process of identifying, analyzing and prioritizing potential risks and uncertainties that would affect the successful completion of a software development project. These risks can range from technical issues and resource constraints to changes in project requirements, market conditions and external factors. The primary goal of risk assessment is to proactively manage and mitigate these risks to ensure the project's objectives are met. Following are key reasons as to why risk assessment is essential in software projects.

i) Early problem identification - spot problems before they escalate.

ii) Efficient Resource Allocation - allocate resources effectively.

iii) Cost Control - Identifying and managing risks can help control project costs.

iv) Schedule management - maintaining project timelines.

v) Quality assurance - address quality risks to

ensure the final product meets expectations.

vii) Reputation management - Protect organization's image and avoid legal issues by manging risks.

viii) Stakeholder communication - keep clients, management and team informed about potential challenges to set realistic expressions.

viii) Increasing project success rate - projects that manage risks effectively have a better chance of success.

2) Software Configuration Management (SCM) is a set of practices and processes used to systematically, control, organize and track changes in software projects. It's primarily role is to ensure the integrity stability and quality of a software system throughout its development lifecycle. Here's how SCM contributes to project quality.

i) Version Control: SCM tracks and manages different versions of software ensuring the right version is used, reducing errors.

2) change management: Orgainizes changes, ensuring thorough testing and documentation to prevent defects.

iii) Traceability: SCM links changes to specific requirements enhancing understanding and meeting project requirements.

iv) Configuration management - It controls all software components preventing configuration release errors in each release.

v) Parallel development - SCM allows multiple developers to work concurrently without conflicts, maintaining code quality.

vi) Automated Build & Deployment: Integration with SCM ensures consistent error free software building and development.

vii) Backup and recovery - SCM provides backup & recovery mechanisms to protect against data loss.

viii) Auditing and Compliance: Tracks changes for auditing & regulatory compliance crucial in regulated industries to ensure quality and compliance standards.

3) Formal Technical Reviews (FTR) are systematic well structured processes for reviewing and evaluating various aspects of software development such as requirements, design, code and documentation. FTRFTRs play a crucial role in ensuring software quality and reliability through

the following mechanisms.

i) Error detection & prevention: FTRs catch and prevent errors early in development.

2) Knowledge Sharing: Team collaboration enhances understanding.

3) Compliance: Ensures adherence to coding & design standards.

4) Requirement Validation: Verifies clear & complete requirements.

5) Risk Mitigation: Addresses potential issues before they escalate.

6) Consistency: Enforces clear documentation & communication.

7) Quality improvement: Feedback loop leads to ongoing improvement.

8) Enhanced process: Structured reviews cover all the aspects thoroughly, boosting reliability.

4] A formal walkthrough in the context of a software project is a structured and systematic way process for reviewing and evaluating software artifacts such as code, design documents or requirements. The primary goal is to identify issues, ensure quality and improve the overall project. The following is the step by step process for conducting a formal walkthrough.

i) Preparation: Preparing the artifact & assembling a review team.
ii) Scheduling: Scheduling a meeting and setting an agenda.
iii) Conducting the walkthrough: Conducting a structured review where team members discuss and document issues.
iv) Resolution: Resolving issues and assigning responsibilities for improvements.
v) Documentation: Documenting the review.
vi) Follow-up: After the review, follow up on the assigned actions.
vii) Closure: Closing the review process once ~~the~~ all issues are addressed.
viii) Feedback and Continuous Improvement: Gathering feedback to improve future reviews.

5) Considering software reliability is crucial when analyzing potential risks in a project for several reasons
a) User Expectations: Users expect software to be reliable, ensure software meets user expectations.

b) Business Impact: Software failures can have significant financial implications, prevent financial losses and extra costs.

c) Reputation: Safeguard the organization's image.

d) Maintenance Costs: Reducing long term support expenses.

e) Safety Critical Applications: Avoid catastrophic consequences

f) Regulatory Compliance: Ensure adherence to industry regulations.

g) Data Integrity: Protect data from corruption or loss.

h) Market Competition: Stay competitive with reliable software.

i) Customer Satisfaction: Enhance user experience and loyalty.

j) Project Success: Critical for successful project outcomes.