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)				

9567
Comps B

Assignment 2

1) What is risk assessment in the context of software projects and why is it essential?

→ Risk assessment in the context of software projects is the process of identifying, analyzing and prioritizing potential risks and uncertainties that could 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 assesment is to proactively manage and mitigate these risks to ensure the projects objectives are met.

Following are key reasons as to why risk assessment is essential in software projects

1) Early problem identification - spot problems before they escalate
2) Efficient Reasurce allocation - allocate resources effectively
3) cost control - identifying and managing risks can help control projects costs.
4) Schedule management - maintaining project timelines.
5) Quality assurance - address quality risks to ensure the final product meets expectations
6) Reputation management - protect organization's image and avoid legal issues by managing risks.
7) Stakeholder communication - keep clients, management and team informed about potential challenges to set realistic expressions
8) 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. Its primary role is to ensure the integrity, stability and quality of a software system throughout its development lifecycle. Heres how SCM contributes to project quality.

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

b) Change management :- Organizes changes, ensuring through testing and documentation to prevent defects.

c) Traceability:- SCM links changes to specific requirements, enhancing understanding and meeting project requirements.

d) Configuration management: It controls all software components, preventing configuration-release errors in each release

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

f) Automated Build and deployment: Integration with SCM ensures consistant, error-free software building and development.

g) Backup and recovery- SCM provides backup and recovery mechanism to protect against data loss

h) Auditing and compliance:- Tracks changes for auditing and regulatory compliance, coucial in regulated industries to ensure quality and compliance standard.

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. FTR's play a crucial role in ensuring software quality and reliability through the following mechanisms

1) Error detection and prevention: FTRs catch and prevent errors early in development

2) Knowledge sharing:- Team collaboration enhances understanding

3) Compliance: Ensures adherence to coding and design standard

4) Requirement validation:- Verifies clear and complete requirements.

5) Risk mitigation:- Addresses potential issues before they escalate.

6) Consistency: Enforces clear documentation and communication

7) Quality improvement: Feedback loop leads to ongoing improvement
8) Enhanced process:- Structured reviews cover all aspects thoughly boosting reliability

u)

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

1) Preparation: preparing the artifact and assembling a review team.
2) scheduling:- scheduling a meeting and setting an agenda.
3) Conducting the walkthrough: anducting a structured review where team membes disscuss and document issues.
4) Resolution:- Resolving issues and assigning responsibilities for improvements.
5) Documentation:- Documenting the review
6) Follow-up: After the review, follow up on the assigned actions.
7) Closure:- closing the review process once all issues are addressed
8) 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 Expectation: 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.