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)				

QNO1: What is risk assessment in the context of software projects, and why is it essential?

Ans: Risk assessment in software projects is the process of identifying, analysing, and prioritising potential risks and uncertainties that could impact the project's objectives, including scope, schedule, and budget. It is essential for several reasons:

Proactive Planning: Risk assessment enables project managers to anticipate and plan for potential issues, reducing the likelihood of project delays and cost overruns. By identifying risks early, they can develop strategies to mitigate or manage them effectively.

Resource Allocation: By identifying and prioritising risks, project teams can allocate resources more efficiently. They can focus their efforts on addressing high-priority risks, ensuring that the most critical issues are addressed promptly.

Quality Assurance: Risks, if not managed, can lead to poor software quality. Identifying and addressing risks early in the project lifecycle can help maintain software quality by preventing issues that could otherwise compromise the product's reliability, performance, or security.

Client Satisfaction: Managing risks ensures that project stakeholders have realistic expectations and are satisfied with the project's progress. It can lead to better communication with clients and a higher likelihood of delivering a successful product.

QNO2: Explain the concept of software configuration management and its role in ensuring project quality.

Ans: Software Configuration Management (SCM) is a set of processes and tools for tracking and controlling changes to software artefacts throughout the software development lifecycle. SCM plays a vital role in ensuring project quality:

Version Control: SCM tools manage different versions of software artefacts, ensuring that the correct version is used at each stage of development. This prevents confusion, reduces errors, and maintains consistency in the codebase.

Change Control: SCM provides mechanisms for controlling and documenting changes. This helps prevent unauthorised or unplanned alterations to the software, which can introduce defects and negatively impact quality.

Baseline Management: SCM helps define and manage baselines, which serve as reference points for quality and consistency throughout the project. These baselines can include requirements, design, and code, ensuring that changes are well-documented and approved.

Traceability: SCM facilitates traceability, allowing project teams to trace changes back to their origins and understand their impact. This is essential for debugging, quality assessment, and ensuring that requirements are met.

QNO3: How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?

Ans: Formal Technical Reviews (FTR) are structured, systematic evaluations of software work products, such as code, design documents, or requirements. They contribute to software quality and reliability in the following ways:

Defect Detection: FTRs focus on identifying defects and issues in work products. By catching problems early in the development process, FTRs reduce the cost of fixing issues and prevent defects from propagating to later stages.

Knowledge Sharing: FTRs promote knowledge sharing among team members. Review participants gain a deeper understanding of the work product, which can lead to better code quality and adherence to standards.

Consistency and Standards: FTRs ensure that the software adheres to defined standards, guidelines, and best practices. This consistency enhances software reliability by reducing the likelihood of errors and deviations from quality expectations.

Risk Mitigation: FTRs contribute to risk mitigation by identifying issues that could affect software quality. This allows project teams to take corrective actions, reducing the likelihood of quality-related problems.

QNO4: Describe the process of conducting a formal walkthrough for a software project.

Ans: The process of conducting a formal walkthrough for a software project typically includes the following steps:

Preparation: The author of the work product (e.g., code, design document) prepares the documentation to be reviewed. This may include creating a document or presentation that describes the work product's purpose, context, and important details.

Scheduling: A meeting is scheduled with appropriate team members who will participate in the review. The participants may include developers, testers, and subject matter experts.

Overview: The author gives an overview of the work product, explaining its purpose, how it fits into the project, and any specific areas or issues that should be focused on during the review.

Review: Review participants examine the work product in detail, looking for issues, defects, or deviations from standards. They may use checklists or guidelines to guide their evaluation.

Discussion: Participants discuss the identified issues, providing feedback, suggestions, and corrections. This discussion is a critical part of the review process, as it allows for a thorough examination of the work product.

Documentation: All issues, decisions, and actions taken during the walkthrough are documented. This documentation ensures that agreed-upon changes or actions are properly recorded and tracked.

Follow-up: After the review, the author incorporates the suggested changes and improvements into the work product. If necessary, a follow-up review may be scheduled to ensure that the issues have been addressed satisfactorily.

QNO5: Why is it important to consider software reliability when analysing potential risks in a project?

Ans: Considering software reliability in risk analysis is of utmost importance because:

Operational Consequences: Unreliable software can lead to operational failures, causing financial losses, downtime, and damage to an organisation's reputation. This can be especially critical in industries where software is a mission-critical component.

Safety and Security: In certain domains like healthcare, automotive, or aerospace, software reliability is directly tied to safety and security. Unreliable software can result in life-threatening situations or security breaches.

Cost Implications: Addressing reliability issues late in the project lifecycle can be extremely costly. It's more efficient and cost-effective to identify and mitigate potential reliability risks during the early stages of the project.

Resource Allocation: By analysing reliability risks, project managers can allocate resources for testing, verification, and validation activities to ensure that the software meets its reliability requirements. This allocation helps in preventing reliability-related failures.

Client Satisfaction: Reliability issues can lead to client dissatisfaction and a loss of trust. By addressing reliability risks proactively, project teams can ensure that the software meets or exceeds client expectations.

Incorporating software reliability into the risk analysis process is crucial for delivering high-quality, dependable software that meets the needs of both clients and end-users.