Sanket Mane
9621 TE COMPS(B)

Page :
Date :

Q1] What is risk assessment in the context of software projects, and why is it essential?

→ Risk assessment in software project is a critical process for identifying and managing potential threats that could impact project success. It involves:

(1) identification: Recognizing risks like technical, operational, or finicial challenges that could hinder the project process.

(2) Analysis: Assessing the likelihood and impact of each risk; determining its severity.

(3) Prioritization: Ranking risks based on their significance to allocate resources and attention effectively.

(4) Mitigation Planning: Developing strategies to address high-priority risks and contingency plans if risks materialize.

Risk assessment is essential for several reasons:

(i) Issue Prevention
(ii) Resource Allocation
(iii) cost control
(iv) Time Management
(v) Quality Assurance
(vi) Project Success
(vii) Stakeholder Communication.

Q2] Explain the concept of software configuration management and its role in ensuring project quality.

→ Software Configuration Management (SCM) is a discipline that involves processes, practices, and tools for managing and controlling software changes throughout the development lifecycle.

(i) Version Control: SCM tracks and manages different version of software components.

(ii) Change Management: It establishes a structured process for proposing, reviewing and implementing changes.

Page :
Date :

(3) Traceability: SCM associates code changes with specific requirement or issues.
(4) Parallel Development: In larger project with multiple developers, SCM supports concurrent work on different code branches.
(5) Backup and Recovery: SCM tools provide data backups and recovery capabilities, protecting project

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

→ Formal technical reviews (FTR) are structured evaluation processes in software development that significantly contribute.

(i) Error detection: FTRs uncover defects and issues early, reducing the cost of fixing problems.
(ii) Quality Assurance: FTRs ensure that software adheres to predefined quality standards and best practices.
(iii) Knowledge sharing: FTRs facilitate understanding among team members, contributing to enchanced software reliability through shared insights.
(iv) Improved documentation: FTRs leads to better documentation aiding developers and maintainers and supporting software reliability.
(v) Enhanced communication: Improved team and stakeholders communication ensures alignment with project goals and reliability.
(vi) Validation of Requirement: FTRs validate that requirements and design are in sync.

Page :
Date :

Q4] Describe the process of conducting a formal walkthrough for a software project.

→ A formal software projet walkthrough is astructed process for reviewing and improving a software project.

(i) preparation: Define objectives, select participants, and gather relevent documentation.

(ii) Scheduling: Set a convenient time for the review, ensuring key stakeholders.

(iii) Conducting the walkthrough:
- Introduction: The moderator explain the objectives and rolcs
- Presentation: The author presents the software, explaing its purpose and design.
- Review and discussion: Participants review, ask questions and express concerns.
- Defect identification: Actively identify and categorize defects.

(iv) Recording and documention: Record identified issues and decisions.

(v) Reporting: Prepare a repoit summarizing results, issues, resolutions and recommendations.

Q5] Why is it important to consider software reliability when analyzing potential risks in a projects?

→ (i) user satisfaction: Reliable software ensures users have a positive experience, while unreliable software can lead to user frustration.

(ii) financial Implication: Unreliable software can be costly due to need for frequent bug fixes, updates, and customer support.

Page :
Date :

(iii) Project delay: Reliability issues often lead to project delays as developers divert their attention from new features to fixing problems.

(iv) Quality Assurance: Neglecting reliability can result in poor overall software quality, increasing the risk of project failure and negatively impacting the quality assurance process.

(v) competitive Advantage: Reliable software is a competitive advantages attracting users who prefer consistent and dependable software over unreliable alternatives