

Name ÷ Glison Pereira

Roll No ÷ 9568

Div ÷ TE Comp B

Batch ÷ A

Assignment ÷ 2

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

solⁿ: i) Risk assessment in the context of software projects is the process of identifying, analyzing, and evaluating potential risks or uncertainties that could impact the successful completion of a software development project.

ii) This assessment helps project teams and stakeholders understand the potential challenges and make informed decisions to mitigate or manage these risks effectively.

iii) Here's why risk assessment is essential in software projects:

a) Early Problem Identification: Risk assessment allows project teams to identify potential issues at an early stage, often during project planning or even before development begins.

b) Resource Allocation: Understanding the risks associated with a project helps in allocating resources more effectively.

c) Cost Control: Unmanaged risks can lead to project delays, cost overruns and scope changes.

d) Stakeholder Communication: Transparent risk assessment and management facilitate

communication with stakeholders.

c) Contingency Planning: Through risk assessment, teams can develop contingency plans that specify how to respond to various risk scenarios.

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

Solⁿ: i) Software Configuration Management (SCM) is a set of processes and tools used to systematically manage, control, and track changes to a software project's components and their relationships.

ii) Its primary role is to ensure the integrity of a software system throughout its development lifecycle and to support the delivery of a high-quality product.

iii) Here's how SCM contributes to project quality:

a) Version Control: SCM system, like version control systems (eg: GIT, SVN), enable the tracking of changes made to source code, documentation, and other project artifacts.

b) Change Management: SCM helps in controlling and documenting changes to the software.

c) Baselining: SCM establishes a baseline, which is a well-defined, stable reference point for a software project.

d) Configuration Identification: SCM helps in identifying and labelling the specific configuration items (CIs) that make up the software.

e) Build and Release Management: SCM plays a crucial role in automating the build and release processes.

Q3) How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?

Solⁿ i) Formal Technical Reviews (FTR) also known as Formal Inspections or Peer Reviews, are systematic and well-structured processes for evaluating and improving the quality and reliability of software during its development.

ii) FTRs contribute to ensuring software quality and reliability in several ways:

a) Defect Detection and Removal: FTRs are focused on identifying defects, errors, and issues in the software.

b) Knowledge Sharing: FTRs provide an

opportunity for knowledge sharing and cross-training among team members.

c) Consistency and Standard Adherence: FTRs help enforce coding and design standards.

d) Risk Identification: FTRs provide a platform for identifying and addressing potential risks early in the development process.

e) Improved documentation: FTRs can lead to improved documentation. Reviewers may identify documentation gaps or ambiguities and request necessary updates.

Q4) Describe the process of conducting a formal walkthrough for a software project?

Solⁿ: i) A formal walkthrough is a systematic and structured process for reviewing and discussing a software project, such as a design document, code, or requirements specifications, to identify issues, improve quality, and ensure that it meets its intended objectives.

ii) Here is step-by-step guide to conducting a formal walkthrough for a software project:

a) Preparation:

- **Select the Document:** Determine which document or artifact needs to be reviewed. This could be a software design, code or any other project-related document.
- **Choose Reviewers:** Assemble a team of reviewers, including subject matter experts and relevant stakeholders.
- **Schedule the Meeting:** Set a date and time for the formal walkthrough meeting. Ensure that all key participants can attend.
- **Distribute the Document:** Share the document with all reviewers in advance of the meeting to give them time to review and prepare their comments.

b) Introduction:

- **Meeting Kickoff:** At the beginning of the meeting, the moderator (often the author of the document) welcomes the participants and outlines the purpose and agenda of the review.

c) Presentation:

- **Author's Overview:** The author presents an overview of the document, highlighting key points, goals, and context. The author should avoid just reading the document.

celebration but instead focus on providing insights and clarifications.

d) Documenting Results:

- Meeting Minutes: The outcomes of the formal walkthrough, including the issues raised and resolutions reached, should be documented in meeting minutes.

e) Closure:

- Meeting Conclusion: The moderator concludes the meeting, expressing appreciation to the participants for their contributions.
- Feedback: Encourage participants to provide feedback on the review process itself, looking for ways to make it more effective and efficient for future reviews.

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

Solⁿ:- i) Considering software reliability when analyzing potential risks in a project is essential because software reliability is a critical factor that can significantly impact the success and overall quality of a software project.

ii) Here are several reasons why software reliability is important in risk analysis:

a) User Satisfaction: Reliable software is more likely to meet user expectations and perform as intended.

b) Financial impact: Unreliable software can lead to financial losses due to potential customer churn, costly support and maintenance, and lost business opportunities.

c) Operational Efficiency: Unreliable software can disrupt business operations, leading to downtime and increased support and maintenance efforts.

d) Time Delays: Unforeseen reliability issues can cause project delays.

e) Complex Debugging and Maintenance: Fixing reliability issues in software can be challenging

and time-consuming. It often involves
extensive debugging, retesting, and code
refactoring.