Kradyumnaa Kadam
9547
COMPS B

SE Assignment-2

Q1.

→ Risk assesment in the context of software project is the process of identifying, analyzing & prioritizing potential risks and uncertainities that could affect the successful completion of a software development project. These risk can range from technical issue and resource constraints to change in project requirements, market condition and external factors. The primary goal of risk assesement is to protectively manage and migrate this risks to ensure the project objectives are met. Following are key reasons as to why risk assesment is essential in software project.

1) Early problem identification - spot problems before they escalate
2) Efficient resource allocation
3) Cost control
4) Schedule management
5) Quality assurance
6) Reputation management - protect organization's image and avoid legal issue by managing risk.
7) stake holder communication - Keeps client management & team informed about potential challenges to set realistic expression
8) Increase project success rate - projects that manage risks effectively have a better chance of success.

02]
→ Software configuration management (SCM) is a set of practise & process 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 life cycle. Here's how SCM contributes to project quality.

1) Version control: SCM manages and track different version of software.
2) Change management: Organizes changes, ensuring through testing and documentation.
3) Traceability: SCM links changes to specific requirements, enhancing understanding and meeting requirements.
4) Configuration management. It controls all software components, preventing configuration - release error in each release.
5) Parallel development - SCM allows multiple developer to work concurrently
6) Automated Build and Deployment - Error free software.
7) Backup and recovery - SCM provides backup and recovery mechanism
8) Auditing and compliance - Tracks changes for auditing and regulatory compliance.

03]
→ Formal technical reviews (FTR) are systematic, well structured process for reviewing and evaluating various aspects of software development, such as requirement, design, code and documention FTR plays a crucial role in ensuring software quality and reliability through following mechanism.

1) Error detection and prevention: FTRs catch and prevent errors easily in development
2) Knowledge sharing: Team collaboration enhance understanding
3) Compilance: Ensure adherence to coding and design standard
4) Requirement validation: Verifies clear and complete requirements
5) Risk mitigation: Address potential issue before they escalate.
6) Improvement.

6) Consistency - Ensures clear documentation and communication
7) Quality improvement - Feedback loop lead to ongoing improvement
8) Enhanced process - Structured reviews cover all aspects.

Q4)

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

1) Preparation - Preparing the artifacts & assembling a review on team.
2) Scheduling - Scheduling a meeting and setting an agenda
3) Conducting the walkthrough - Conducting a structured review where team members discuss and document issues
4) Resolution - Resolving issues and assigning responsibilities for improvement
5) Documentation - Documenting the review
6) Follow-up: After the review, follow up on the assigned action
7) Closure: Closing the review process once all issues are addressed
8) Feedback and continous improvement: Gathering feedback for improvement.

05]

→ Considering software reliability is crucial when analyzing potential risk in all project for several reasons.

a) User Expectations - User expect software to be reliable Ensure software meets user expectations.

b) Bussiness Impact - Software failures can have significant financial implications. Prevent financial losses and extra cost.

c) Reputation - Safeguard the organization image.

d) Maintenance cost - Reducing long-term support expenses

e) Safety critical application : Avoid catastrophic consequences

f) Regulatory compilance : 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 succesful project outcomes.