Name - Mrunal Kotambkar

Roll no - 9552

Assignment 2.

1)

Ans Risk management in context of software project refers to the systematic process of identifying, analyzing priority and managing potential risk or uncertainties that can affect the successful completion of software development project.

Reasons for risk managements;

1) Early problem identification: Risk management allow project managers and team to identify potential issues an challenge early in the project's lifecycle.
2) Resource allocation: By accessing risks, project manager can allocate resource, both human & financial, more effectively.
3) Time management: Identifying risk helps in creating more realistic project schedules and milestones.
4) Cost control: Effective risk assessment help in estimating and controlling project cost. project can costly overrun & budget issues.
5) Quality Assurance: Risks related to software quality, such as code defects. or integration challenges, can be addressed during risk assessment.
6) Stakeholder communication: Clear risk management practice facilitate transparent communication with project stakeholders.

Q2)

Ans Software configuration management (SCM) is a set of process practice, and tools that are used to manage and control changes to software throughout its development lifecycle; The primary. goal of SCM is to ensure that the software product is of high quality, reliable and meets it intended requirements.

Concept of software configuration management

1) Version Control : SCM involves version control, which track changes to the software's source code, documents & artifacts
2) Change management: SCM defines proceduses for requesting, reviewing, approving, and implementing changes to the software
3) Baseline Management, SCM establishes baselines, which are well defined snapshot of the software at different stages of development.
4) Configuration Management : It involves identifying & documenting the software components, their versions, & their interrelationships

Roles in ensuring project Quality

1) Consistency & Reproducibility : SCM ensures that all team members work with consistent version of software artifacts, reducing the risk of error due to version mismatches-
2) Change control: By managing changes through a structured process, SCM prevents unauthorized or undocumented changes that could introduces defects.
3) Quality Assurance: SCM enables the enforcement of coding standard, testing procedure & documentation requirements

4) Baseline Management: The use of baseline allows project team to perform quality check at specific milestones. ~~they~~

5) Traceability: SCM provides traceability between requirement, design, source code, and testing artifacts.

6) Risk mitigation: By managing configurations & changes systematically, SCM helps identify & mitigate risks early in the development process

Q3)

Ans. Formal technical Review (FTRs) are a systematic and structured approach to evaluating and improving the quality & reliability of software. FTR play crucial role in software development by contributing of quality assurance in the following ways:

1) Defect detection! FTRs are designed to identify defects early in software development process. This help in detecting defects like ambiguities. inconsistencies, and errors before they .

2) Verification & validation: FTR's serve as a mean of verification & validation. They ensure that the softw are under review align's with the project require -ment & specifications & standards.

3) Consistency and Adherence to standards: FTR's ensur that the software follows coding standards, desig principles & best practices. By verifying adherence to

these standards FTR's contribute to the developments of consistent & maintainable software

4) Improved communication: FTR's encourage effective communication among team members. participants from various role & domain collaborate during the review process.

5) knowledge sharing: FTR's promote knowledge sharing among team members. Reviewers learn from one another, gaining insights into different aspects of the software. The knowledge transfer contributes to professional development & improved software quality.

6) Continuous improvement: FTR's provide an opportunity for teams to learn from their mistake & success. post-review discussions often lead to process improvements & the adoption of better practice in future software development cycles.

Q4)

Ans Conducting a formal walkthrough for a software project is systematic & structured process that involves group of stakeholders reviewing a software artifact to identify defect, verify compliance with requirement and ensure quality.

1) Preparation
- Identify software artifacts to be reviewed
- select the participants for the walkthrough
- schedule walkthrough & inform all the participants well in advance.

2) Introduction
- introduce goals & purpose of review
- clarify objective such detecting error, ensuring

3) Review

- Going through software artifacts section by section.
- author of document highlight key points & explain code.
- Reviewer analze error, ambiguities, inconsistencies and deviations from standards & requirements

4) Issue identification

- During review participants identify any issues, defects or concern.
- These issue may includes spelling, grammar, ambiguity, violation of coding standard.
- each issue should be documentated

5) Resolution & Action scheme.

- Issue are identified & team discusses solution at this stage
- participants takes their respective role

6) Documentation:

- Throughout the walkthrough, the recorder documents all issues and action items, along with any discussion & decision made during review.

• The recorded information will serve as the basis for subsequent follow up & tracking.

7) Follow up closure:

∴ Project manager track progress & ensure that all issues ared addressed.

Q5)

Ans Considering software reliability is essential when analyzing potential risk in project because.

1) It affect project success: Unreliable software can lead to project failures, delay or budget overruns, making it a significant risk.

2) Customer satisfaction: Reliability issue can result in dissatisfied customers, damaging reputation.

3) Cost implications: Fixing reliability post - development can be costly, impacting the project's budget.

4) Safety critical systems: In sectors like healthcare or aviation, unreliable software can pose serious safety risks

5) Legal consequences: Software failure can lead to legal issue & & liabilities.