Shreedhar Bhat
9590
TE comps B

Assignment - 2

1) → Risk assessment in the context of software projects is the process of identifying, analyzing & prioritizing potential risks & uncertainities that could affect the ~~successful~~ successful completion of a software development project

→ These risks can range from technical issues & resources constraints to changes in project requirements market condition and external factors.

→ The primary goal of risk assesment is to proactively manage & mitigate these risks to ensure the projects objectives are met.

→ Following are key reasons as to why risk assesment is essential in software projects.

1) Early problem identification - spot problems before they escalate.

2) Efficient Resource allocation - allocate resources effectively

3) cost control - Identifying & managing risks to ensure the final product meets expectations.

4) Schedule management - maintaining project timelines

5) Quality assurance - address quality risks to ensure the final product meets expectations.

6) Reputation management - Protect organizations image & avoid legal issues by managing risks

7) Stakholder communication - keep clients, management & avoid legal issues by managing risks

8) increasing projects success rate - project that manage risks effectively have a better chance of success

2) Software configuration Management (SCM) is a set of practices & process used to systematically control, organize & track changes in software projects. Its primary role is to ensure the integrity stability & quality of a software system throughout its development lifecycle. Here's how

1) version control: SCM tracks & manages different versions of software
2) Change management: organizes changes, ensuring through testing and documentation to prevent defects.
3) Traceability: Scm links changes to specific requirements, enhancing understanding & meeting project requirements.
4) Configuration management: It controls all software components to ensure reliable

3) Formal Technical Review (FTR) are systematic, well-structured process for reviewing & evaluating various aspects of software development, such as requirements, design, code & documentation. FTRs play a crucial role in ensuring software quality & reliability through the following mechanisms.

1) Error detection & prevention: FTRs catch & prevent errors early in development.
2) Knowledge sharing: Team collaboration enhances understanding
3) Compliance: Ensures adherence to coding & design standards.
4) Requirements validation: Verifies clear & complete requirements.
5) Risk mitigation: Addresses potential issues before they escalate.
6) Consistency: Enforces clear documentation & communication
7) Quality improvement: feedback loop leads to ongoing improvement.
8) Enhanced focus: Structured review covers all aspects thoroughly boosting reliability.

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

1) Preparation: Preparing the artifact & assembling a review team.
2) Scheduling: Scheduling a meeting & setting an agenda
3) Conducting the walkthrough conducting a structured review. where team members discuss & document issues.
4) Reduction: Resolving issues & assigning responsibilities for improvements.
5) Documentation: Documenting the review
6) Follow up: After the review, follow up on the assigned actions
7) Closure: closing the review process once all issues are addressed,
8) Feedback & continuous improvement: Gathering feedback to improve future reviews.

5) Considering Software reliability is crucial when analyzing potential risks in a project for several reasons.

a) user expectations: Users expect software to be reliable ensure software meets user expectations.

b) Business Impact: Software failures can have significant financial implications prevent financial losses & extra costs.

c) Reputation: Safeguard the organizations image.

d) Maintenance costs: Reducing long-term support expenses.

e) Safety critical Applications: Avoid catastrophic consequences

f) Regulatory Compliance: 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 & loyalty

j) Project success: Critical for successful project outcomes.