Bryce Miranda
Roll : 9768
Comps B

Fr. Conceicao Rodrigues College of Engineering,Mumbai

SOFTWARE ENGINEERING (CSC601)

Assignment -II

Date: 17-10-23

CO5: Identify risks, manage the change to assure quality in software projects.

Assignment 2

1. What is risk assessment in the context of software projects, and why is it essential?
2. Explain the concept of software configuration management and its role in ensuring project quality.
3. How do formal technical reviews (FTR) contribute to ensuring software quality and reliability?
4. Describe the process of conducting a formal walkthrough for a software project.
5. Why is it important to consider software reliability when analyzing potential risks in a project?

Rubrics :

Indicator	**Average**	**Good**	**Excellent**	**Marks**
Organization (2)	Readable with some mistakes and structured (1)	Readable with some mistakes and structured (1)	Very well written and structured (2)	
Level of content(4)	Minimal topics are covered with limited information (2)	Limited major topics with minor detailsare presented(3)	All major topics with minor details are covered (4)	
Depth and breadth of discussion(4)	Minimal points with missing information (1)	Relatively more points with information (2)	All points with in depth information(4)	
Total Marks(10)				

① What is risk assessment in the content of software projects, and why it is essential?

→ Risk assessment in the content of software project is the process of identifying, analyzing, and mitigating potential risks or uncertainties that could affect the successful completion of a software development project. It is an essential component of project management and involves evaluating the probability. Here are some reasons why risk assessment is essential in software projects.

1) Early problem identification: Risk assessment allows project teams to identify potential issues before they escalate into critical problems. By recognizing and addressing risks early, teams can minimize the impact and reduce the cost of mitigation.

2) Resource Allocation:- It helps in effective allocation of resources, including time, budget, and personnel.

3) Budget and Schedule control: By identifying risks related to cost overruns and schedule delays, project managers can implement strategies to keep the project on-track and within budget.

② Software Configuration Managment (SCM) is a set of practices and processes that helps control, track, and manage changes to software throughout its development lifecycle. SCM encompasses and managment of various software artifacts, including sourcecode, documentation, binaries, and other project-related assets. Its primary role is to ensure project quality by maintaining consitency, traceability, and version control in software development process.

1) version control :- SCM system, often referred to a version control system (VCS), enable developers to track changes to source code and project files.
2) Configuration Identification :- SCM help in identifying and defining the software configuration
3) Change Managment :- SCM provides a structed process for requesting, reviewing, approving and implementing changes to the software. This helps in preventing unauthorized or uncontrolled changes, which can introduce bugs, or break existing functionality. Change managment ensures that changes are made with a clear purpose and impact assessment
4) Build managment :- SCM plays a role in automating and managing the build process,
5) Release Managment :- SCM is crucial in managing software releases. It helps in packaging, documenting, and erewing.

3) How do Formal technical Review (FTR) contribute to ensuring software quality and reliability?

→ ① Defect Detection and correction: FTRs involve a group of reviews who systematically examine the software artifacts in detail. This process is highly effective in detecting defects, such as coding errors, design flaws, and logic issues. By identifying these problems at an early stage, FTRs help prevent these defects from propagating through the development process, reducing the cost and effort required to fix them later.

② Knowledge sharing: FTRs provide a platform for knowledge sharing and learning within the development team. Team members can share their expertise, best practices, and domain knowledge during the review process. This not only helps in improving the quality of the artifact.

(4)

(A) Verification of Requirements :- FTRs help ensure that the software requirements. By systematically examining the design and code against requirements.

① Preparation :- Identify the artifact to be reviewed. Ensure its complete and ready for review.
- Assemble a review team consisting of relevant stakeholders, which may include developer, tester, architects, and subject matter experts.

② Distribution of Materials.
- Distribute the artifacts to be reviewed to all team members well in advance of the walkthrough.

③ Setting the Agenda:
- Define the goals and objectives of the review, including what aspects to focus on (eg. correctness, compliance with standards, performance, security)

④ Review Meeting:
- Gather all participants for review meeting, either in person or through a virtual collaboration problem.
- The moderator includes the agenda, explains the purpose of the review, and sets the ground rules for the meeting, including time limits for each phase of the review.

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

1) End-User satisfaction: software reliability directly impacts end-user satisfaction. Unreliable software can lead to frequent crashes, errors, and downtime, frustrating users and potentially causing them to abandon.

2) Reputation and Brand Image: Reliability issues in software can damage an organization's reputation and brand image. News of software failures or data breaches can spread quickly through social media and impact public perception.

3) Customer Retention:- satisfied and loyal customers are more likely to continue using a product and recommend it to others. Software reliability.

4) Financial Implications:- Software reliability issues can have significant financial consequences. The cost of addressing and resolving reliability problems, including bugs fixes, downtime, and customer support.

7) Operational Efficiency:- Unreliable software can disrupt business operations. When software fails or experience frequent issues. it can result in downtime, delays, and increased support and maintenance efforts. These disruptions can reduce operational efficiency and increase operational costs.