Object Oriented Software Engineering

MCA Semester III

Department of Computer Science

Gujarat University

Unit – 6 Software Configuration Management, Risk management and maintenance

- Software Configuration Management (SCM) is a process that systematically manages, organizes, and controls changes in documents, codes, and other entities throughout the software development lifecycle (SDLC).
- Importance: SCM ensures that changes are made in a controlled manner, preventing unauthorized modifications that can lead to inconsistencies and defects. It enables teams to work in parallel while maintaining the integrity of the software product.
- Example: In a large software project like an e-commerce platform, SCM helps track changes to the source code as multiple developers work on different features simultaneously. This coordination is essential to ensure that all changes are integrated smoothly without introducing errors.

SCM Repository

- Software Configuration Management (SCM) is a process that systematically manages, organizes, and controls changes in documents, codes, and other entities throughout the software development lifecycle (SDLC).
- Importance: SCM ensures that changes are made in a controlled manner, preventing unauthorized modifications that can lead to inconsistencies and defects. It enables teams to work in parallel while maintaining the integrity of the software product.
- Example: In a large software project like an e-commerce platform, SCM helps track changes to the source code as multiple developers work on different features simultaneously. This coordination is essential to ensure that all changes are integrated smoothly without introducing errors.

SCM Process

- The SCM process consists of several phases that ensure effective management of software configuration items throughout their lifecycle.
- The first phase is Planning and Identification, where teams identify which configuration items need management based on project requirements.
- This phase sets the foundation for effective SCM practices by establishing clear guidelines for what constitutes a configuration item.

SCM Process

- Next is Version Control, which involves managing different versions of configuration items to ensure consistency across development efforts.
- This phase includes creating branches for new features or bug fixes while maintaining the integrity of the main codebase.
- For example, if a team is developing an online shopping platform, they might create separate branches for implementing payment processing features while keeping the main branch stable for ongoing production use.

SCM Process

- The third phase is Change Control, where teams implement procedures for requesting and approving changes to configuration items.
- Change requests must be documented and reviewed by relevant stakeholders before implementation.
- Finally, Configuration Auditing involves verifying that changes comply with established standards and that all configuration items are correctly documented.
- This audit process helps ensure accountability and traceability throughout the project lifecycle.

Types of Software Risks

- Technical Risks: Issues related to technology choices or integration challenges.
- Project Management Risks: Delays or resource constraints affecting project timelines.
- Business Risks: Changes in market conditions or customer requirements impacting project viability.
- Example: A technical risk in a cloud-based application could involve compatibility issues with new APIs introduced by third-party services, which may require additional development time to resolve.

Risk Identification

- Risk identification involves recognizing potential risks that could affect the project's success. This is typically done through brainstorming sessions, expert interviews, or analysis of historical data.
- Techniques:
- SWOT Analysis: Evaluates strengths, weaknesses, opportunities, and threats.
- Checklists: Use predefined lists of common risks in similar projects.
- Brainstorming Sessions: Gather input from team members across disciplines.

Risk Identification

- Example: During the planning phase of a healthcare application project, stakeholders might identify risks related to data privacy regulations that could impact development timelines.

Risk Projection and Refinement

- Risk projection involves estimating the likelihood and impact of identified risks on project objectives. Refinement includes continuously updating risk assessments as new information becomes available.
- Methods:
- Qualitative Analysis: Assess risks based on their severity and probability.
- Quantitative Analysis: Use statistical methods to calculate potential impacts on project costs or schedules.

Risk Projection and Refinement

- Example: In an agile software development environment, teams may hold regular sprint reviews to reassess risks based on current progress and adjust their strategies accordingly.

The RMMM Plan

- The Risk Management, Monitoring, and Mitigation (RMMM) plan outlines strategies for managing risks throughout the project lifecycle. It includes identification, assessment, response planning, monitoring, and communication strategies.
- Components:
- Risk Assessment Matrix: A tool for prioritizing risks based on their likelihood and impact.
- Mitigation Strategies: Specific actions to reduce or eliminate risks.
- Monitoring Procedures: Regular reviews of risk status throughout the project.

The RMMM Plan

- Example: In a software development project for a financial institution, the RMMM plan may include specific measures for ensuring compliance with regulatory requirements as part of risk mitigation efforts.

Project Scheduling

- Project scheduling involves creating a timeline for project activities to ensure timely completion. It includes defining tasks, estimating durations, assigning resources, and setting milestones.
- Tools Used:
- Gantt Charts
- Critical Path Method (CPM)
- Program Evaluation Review Technique (PERT)
- Example: For a software upgrade project at a tech company, a Gantt chart could visually represent tasks such as requirement gathering, development phases, testing periods, and deployment timelines.

Timeline Chart

- A timeline chart visually represents project milestones over time. It helps stakeholders understand progress and deadlines at a glance.
- Components:
- Milestones: Key events or deliverables.
- Duration: Time allocated for each task.
- Dependencies: Relationships between tasks that affect scheduling.
- Example: A timeline chart for launching a new mobile app might include milestones such as prototype completion, beta testing start date, final release date, and marketing campaign launch.

Categories of Software Maintenance

- Types of Software Maintenance:
- Corrective Maintenance: Fixing defects found after deployment.
- Adaptive Maintenance: Modifying software to work in new environments or with new technologies.
- Perfective Maintenance: Enhancing existing features based on user feedback or performance improvements.
- Preventive Maintenance: Making updates to prevent future issues from arising.

Categories of Software Maintenance

- Example: After releasing an online banking application update that introduces new features (perfective maintenance), developers might later need to address bugs reported by users (corrective maintenance).

Challenges of Software Maintenance

- Legacy Code Issues: Difficulty in understanding or modifying old codebases due to lack of documentation.
- Resource Constraints: Limited personnel available for maintenance tasks can lead to delays.
- Changing Requirements: Frequent updates from users can complicate maintenance efforts as priorities shift constantly.
- Example: A company maintaining an outdated customer relationship management (CRM) system may struggle with integrating new features due to poorly documented legacy code that was written years ago by different developers.

Happy Diwali
Happy New Year