Internal Assessment Test 3 July 2024									
Sub:	**Software Engineering and Project Management**				Sub Code:	**21CS61**	Branch:	AIML	
Date:	**8/07/24**	Duration:90 m	Max Marks:	50	Sem :	**V1**		OBE	

	Answer any FIVE FULL Questions	Marks	CO	RBT
1	Discuss software quality and its attributes. Explain process based quality **Software Quality** refers to the degree to which a software product meets specified requirements and satisfies the needs of its users. It encompasses both external and internal characteristics. External characteristics are those that are observable by users, such as usability and reliability, while internal characteristics are concerned with the code and architecture of the software, impacting its maintainability and efficiency. Key attributes of software quality include: 1. **Functionality**: The software's ability to provide functions that meet stated and implied needs when used under specified conditions. 2. **Reliability**: The capability of the software to maintain its level of performance under stated conditions for a specified period. 3. **Usability**: The ease with which users can learn and use the software. 4. **Efficiency**: The software's capability to provide appropriate performance relative to the amount of resources used. 5. **Maintainability**: The ease with which a software product can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. 6. **Portability**: The ability of the software to be transferred from one environment to another. **Process-Based Quality:** Process-based quality emphasizes the methods and processes used to develop the software, rather than just the final product. It ensures that the development process itself is efficient and capable of producing high-quality software. This approach involves: ● **Process Management**: Establishing and maintaining an efficient and effective development process. ● **Standards Compliance**: Adhering to established standards and best practices to ensure consistent quality. ● **Continuous Improvement**: Regularly assessing and improving processes based on feedback and results. Process-based quality is crucial in software engineering as it allows for systematic control and improvement of the quality of software products through well-defined and repeatable processes	[10]	5	L2

2.	Explain software reviews and inspections of quality assurance **S**oftware Reviews: Software reviews are systematic evaluations of software products or processes conducted to identify defects, ensure compliance with standards, and assess quality. Reviews are conducted at various stages of development, including requirements, design, code, and test cases. These reviews are often categorized into informal (like peer reviews) and formal reviews (like walkthroughs and inspections). Inspections: Inspections are a formal type of review where a team examines software artifacts with the goal of detecting defects early in the development process. Inspections are more structured than other types of reviews and typically follow a defined process that includes preparation, overview, inspection meeting, rework, and follow-up. A specific type of inspection, known as Fagan Inspection, involves a detailed process: ● Preparation: Reviewers study the software artifact individually. ● Overview Meeting**:** The team discusses the artifact to ensure a common understanding. ● Inspection Meeting: A moderator leads the meeting where defects are identified. ● Rework: The author of the artifact addresses the defects found. ● Follow-Up: The moderator ensures that all issues have been resolved. Inspections are effective for improving quality by detecting and correcting defects early, which in turn reduces the cost of fixing bugs at later stages. Key Benefits: ● Early defect detection reduces rework. ● Facilitates knowledge sharing and team collaboration. ● Promotes adherence to standards and best practices.	[10]	4	L2
3	Explain SEI CMM levels and key process area SEI CMM Levels: 1. Level 1: Initial o Characteristics: Chaotic and ad hoc development processes. Lack of defined processes or management practices. Relies heavily on individual heroics to complete projects. o Outcome: Project success depends largely on the capabilities of individual team members. High risk of project failure or delays. 2. Level 2: Repeatable o Characteristics: Basic project management practices like planning and tracking costs/schedules are in place.	[10]	5	L22

Processes are somewhat documented and understood by the team.
o Outcome:
Organizations can repeat successful practices on similar projects.
Improved project consistency and some level of predictability.
3. Level 3: Defined
o Characteristics:
Processes for both management and development activities are defined and documented.
Roles and responsibilities are clear across the organization.
Training programs are implemented to build employee capabilities.
Systematic reviews are conducted to identify and fix errors early.
o Outcome:
Consistent and standardized processes across the organization.
Better management of project risks and quality.
4. Level 4: Managed
o Characteristics:
Processes are quantitatively managed using metrics.
Quality goals are set and measured against project outcomes.
Process metrics are used to improve project performance.
o Outcome:
Focus on managing and optimizing processes to meet quality and performance goals.
Continuous monitoring and improvement of project execution.
5. Level 5: Optimizing
o Characteristics:
Continuous process improvement is ingrained in the organization's culture.
Process metrics are analyzed to identify areas for improvement.
Lessons learned from projects are used to refine and enhance processes.
Innovation and adoption of new technologies are actively pursued.
o Outcome:
Continuous innovation and improvement in processes.
High adaptability to change and efficiency in handling new challenges.
Leading edge in technology adoption and process optimization

Level	Key process areas
1. Initial	Not applicable
2. Managed	Requirements management, project planning and monitoring and control, supplier agreement management, measurement and analysis, process and product quality assurance, configuration management
3. Defined	Requirements development, technical solution, product integration, verification, validation, organizational process focus and definition, training, integrated project management, risk management, integrated teaming, integrated supplier management, decision analysis and resolution, organizational environment for integration
4. Quantitatively managed	Organizational process performance, quantitative project management
5. Optimizing	Organizational innovation and deployment, causal analysis and resolution

4	Discuss project plan. Explain various sections of project plan **P**roject Plan Overview A project plan is a formal document that outlines how a project will be executed, monitored, and controlled. It serves as a roadmap for the project team, defining the scope, objectives, and deliverables of the project. The project plan is essential for ensuring that everyone involved in the project is aligned and aware of their roles and responsibilities. Sections of a Project Plan 1. Introduction: ○ Provides a high-level overview of the project, including its purpose, objectives, and scope. ○ Introduces the project team and key stakeholders. 2. Project Scope: ○ Defines what is included in the project and what is not. ○ Details the deliverables and major milestones. ○ Outlines the boundaries of the project to avoid scope creep. 3. Project Schedule: ○ Includes a timeline for the project with key milestones and deadlines. ○ Provides a detailed Gantt chart or similar scheduling tool to visualize the project timeline. 4. Resource Plan: ○ Lists the resources (human, financial, and material) needed for the project. ○ Assigns resources to specific tasks or activities in the project schedule. 5. Risk Management Plan: ○ Identifies potential risks to the project and their impact. ○ Includes mitigation strategies and contingency plans for each identified risk. 6. Communication Plan: ○ Details how information will be communicated among project stakeholders. ○ Specifies communication channels, frequency, and the responsible parties. 7. Quality Management Plan: ○ Defines the quality standards and criteria that the project deliverables must meet. ○ Outlines the processes for quality assurance and quality control. 8. Budget Plan: ○ Provides an estimate of the costs associated with the project. ○ Breaks down the budget into specific categories such as labor, materials, and overhead. 9. Change Management Plan: ○ Details the process for handling changes to the project scope, schedule, or resources. ○ Defines the approval process for changes and how they will be documented. 10. Conclusion:	[10]	4	L3

	○ Summarizes the project plan and reaffirms the commitment of the project team to achieve the project objectives. ○ May include a section on lessons learned or best practices to be followed.			
5	Explain verification versus validation .1. Objectives: o Both techniques aim to remove errors from software. 2. Definitions: o Verification: Ensures outputs of one development phase conform to the previous phase's outputs. o Validation: Ensures fully developed software meets its requirements specification. 3. Objectives Clarified: o Verification Objective: Check if artifacts produced after a phase conform to those from the previous phase (e.g., design documents conform to requirements specifications). o Validation Objective: Check if the fully developed and integrated software satisfies customer requirements. 4. Techniques: o Verification Techniques: Review, simulation, and formal verification. o Validation Techniques: Primarily based on product testing. 5. Process Stages: o Verification: Conducted during the development process to ensure development activities are correct. o Validation: Conducted at the end of the development process to ensure the final product meets customer requirements. 6. Phase Containment of Errors: o Verification aims for phase containment of errors, which is a cost-effective way to eliminate bugs and an important software engineering principle. 7. V-Process Model Activities: o All activities on the right side of the V-process model are verification activities except for the system testing block, which is a validation activity.	[10]	5	L3
6	Explain structured programming & clean software development. Structured programming and clean-room software development The late 1960s marked a pivotal period in software engineering where the complexity of software systems began to outstrip the capacity of human understanding and testing capabilities. Here are the key developments and concepts that emerged during this time: 1. Complexity and Human Limitations: o Software systems were becoming increasingly complex, making it impractical to test every possible input combination comprehensively. o Edsger Dijkstra and others argued that testing could only demonstrate the presence of errors, not their absence, leading to uncertainty about software correctness. 2. Structured Programming: o To manage complexity, structured programming advocated breaking down software into manageable components. o Each component was designed to be self-contained with clear entry and exit points, facilitating easier understanding and validation by human programmers.	[10]	5	L4

	3. Clean-Room Software Development: o Developed by Harlan Mills and others at IBM, clean-room software development introduced a rigorous methodology to ensure software reliability. o It involved three separate teams: Specification Team: Gathers user requirements and usage profiles. Development Team: Implements the code without conducting machine testing; focuses on formal verification using mathematical techniques. Certification Team: Conducts testing to validate the software, using statistical models to determine acceptable failure rates. 4. Incremental Development: o Systems were developed incrementally, ensuring that each increment was capable of operational use by end-users. o This approach avoided the pitfalls of iterative debugging and ad-hoc modifications, which could compromise software reliability. 5. Verification and Validation: o Clean-room development emphasized rigorous verification at the development stage rather than relying on extensive testing to identify and fix errors. o The certification team's testing was thorough and continued until statistical models showed that the software failure rates were acceptably low.			