

THE KAVERY ENGINEERING COLLEGE*(An Autonomous Institution)***UG/PG DEGREE END SEMESTER THEORY EXAMINATIONS, APRIL-MAY 2026****Third Semester****Master of Computer Applications****PMCE2426 - SOFTWARE QUALITY AND TESTING****Regulations - 2024****Duration : 3 Hrs****Maximum : 100 Marks****PART – A(ANSWER ALL QUESTIONS) (Marks 10*2=20)**

Q.No	Questions	BTL	CO	Marks
1.	List any two differences between verification and validation.	L1	1	2
2.	Mention two types of integration testing approaches.	L2	1	2
3.	What is test coverage? State any two reasons why measuring coverage is important in testing.	L1	2	2
4.	What is acceptance testing? Explain its primary purpose in ensuring software quality.	L2	2	2
5.	What is the Onsite/Offshore model in software testing? Give a simple example of its usage.	L1	3	2
6.	How do modern software testing tools enhance the speed and accuracy of test execution?	L2	3	2
7.	How can continuous improvement practices enhance overall software quality in an organization?	L1	4	2
8.	Judge the significance of the Malcolm Baldrige National Quality Award (MBNQA) in motivating organizations toward excellence.	L4	4	2
9.	State any two of Deming's quality principles and explain their relevance to modern software projects.	L5	5	2
10.	Why is continuous improvement through PDCA essential for maintaining long-term software quality? Give a concise justification.	L4	5	2

PART – B (ANSWER ALL QUESTIONS) (Marks 5*16 = 80)

Q.No	Questions	BTL	CO	Marks
11. a	i Describe unit testing in detail. Explain the steps involved in designing and executing unit test cases, the role of stubs and drivers, and give relevant examples of testing individual modules.	L1	1	8
	i Explain the structure and elements of a Control Flow Graph (CFG) in software testing.	L2	1	8
	i Discuss how CFGs are used to analyze program structure and assist in identifying critical test paths with suitable illustrations.			
	Or			
b	i Define control flow testing and explain its major concepts. Describe the process of constructing control flow graphs, identifying paths, and designing test cases using control flow methods.	L1	1	8
	i Discuss different types of data flow anomalies and explain how data flow testing helps identify them. Provide examples of common anomalies like undefined variables, improper usage, and unused data, and explain their impact on program quality.	L2	1	8
12. a	i Define a Software Test Plan and explain its purpose in software testing. Describe why preparing a proper test plan is crucial for ensuring systematic and efficient testing of software products.	L1	2	8
	i Differentiate between static testing and dynamic testing. Explain their methods, advantages, and use cases with examples to show how both contribute to effective software testing.	L2	2	8
	Or			
b	i Define test coverage and explain its different types. Discuss how coverage metrics are used to measure the extent of testing and ensure that all parts of the software are tested adequately.	L3	2	8

	i	Discuss the different types of technical reviews conducted during software development. Explain their objectives, participants, procedures, and how they help in improving software quality before testing begins.	L4	2	8
	i				
13.	a	i Apply the principles of Agile Testing to a continuous integration environment. How does agile testing support quick feedback and adaptability in fast-paced development cycles?	L3	3	8
	i	Critically analyze any two emerging trends in software testing, such as AI-based testing or scriptless automation. How do these trends transform traditional testing approaches?	L4	3	8
		Or			
	b	i Describe the major components of a Test Automation Framework. Apply your understanding by outlining how these components would work together in a web application testing scenario.	L3	3	8
	i	Compare and contrast Test Process Assessment and Test Automation Assessment in terms of objectives, techniques, and outcomes. Analyze how both complement each other in improving testing effectiveness.	L4	3	8
	i				
14.	a	Analyze the structure and key elements of a Software Quality Assurance Plan. How does an SQA plan ensure continuous quality improvement during the software development lifecycle?	L4	4	16
		Or			
	b	Compare and evaluate CMM, PCMM, and CMMI in terms of their scope, focus areas, and contributions to quality improvement. Which model offers the most comprehensive approach to organizational maturity, and why?	L6	4	16
15.	a	Kaveri Software Solutions is developing a new financial application. The project is suffering from high defect leakage (defects found post-release). Testing metrics show high variance in defect discovery rates between sprints. Furthermore, the testers complain that requirements are often vague, forcing them to make assumptions, which leads to unstable test cases and rework. The project manager suspects a fundamental flaw in the requirements-to-test-case transformation process.	L5	5	16
	i.	Analyze how the project manager can apply Statistical Process Control (SPC), specifically using Control Charts (e.g., p-charts or u-charts), to monitor the defect discovery rate during testing. Detail the steps required to set the control limits.			
	ii.	Based on the expected results from the SPC analysis, propose and justify a corrective action plan that integrates a formal requirements-to-test-case transformation technique (e.g., Decision Table Testing or Use Case Testing) to reduce the variance and stabilize the testing process. Explain how this process change aligns with the findings from the statistical data.			
		Or			
	b	Kaveri Games Studio wants to improve its deployment process, which currently takes three days and frequently introduces new bugs due to manual steps. The team decides to adopt an automated continuous integration/continuous delivery (CI/CD) pipeline. They decide to use the Plan-Do-Check-Act (PDCA) cycle to manage this significant change.	L6	5	16
	i.	Detail the specific activities, goals, and metrics the team must define and execute within the 'Plan' and 'Do' phases of the PDCA cycle for the CI/CD pipeline implementation.			
	ii.	Analyze how the 'Check' phase, using quantitative data (metrics) and qualitative feedback, determines if the automation was successful. Based on a scenario where the deployment time only dropped to two days (not the goal of one day) but defect introduction slightly increased, what actions would be required in the subsequent 'Act' phase to stabilize the process before standardizing the change?			

