

Final Assessment Test – April 2025



VIT
Vellore Institute of Technology

Course: BCSE301L - Software Engineering
Class NBR(s): 2229/2231/2233/2236/2239/2242/2249/
2253/2255/2259/2262/2265/2268/2273/2279/2281/
2284/2289/2292/2299/2300/2301/2303/5540

Slot: G1+TG1

Max. Marks: 100

Time: Three Hours

- KEEPING MOBILE PHONE/ANY ELECTRONIC GADGETS, EVEN IN 'OFF' POSITION IS TREATED AS EXAM MALPRACTICE
- DON'T WRITE ANYTHING ON THE QUESTION PAPER

Answer ALL Questions

(10 X 10 = 100 Marks)

1. Suggest the most appropriate generic software process model that might be used as a basis for managing the development of the following systems. Explain your answer according to the type of system being developed:
 - a. A system to control antilock braking in a car.
 - b. A virtual reality system to support software maintenance.
 - c. A university accounting system that replaces an existing system.
 - d. An interactive travel planning system that helps users plan journeys with the lowest environmental impact.
2. A product development team is building a large-scale e-commerce application with frequently changing requirements from the client. Initially, the team followed a traditional Waterfall model but struggled to accommodate late changes, resulting in delays and cost overruns. The company now wants to transition to the Agile model. As a project lead, critically analyze how the Agile methodology would help in this scenario. How would Agile principles and practices address the limitations faced with the Waterfall model? What roles, ceremonies, and artifacts would you implement to ensure smooth and adaptive development? What challenges might arise while shifting to Agile, and how would you overcome them to maintain product quality and delivery timelines?
3. Assume that a university has contracted you to develop an online course registration system (OLCRS).
 - a. Define parallel work activities during the OLCRS project.
 - b. Establish milestones and for the project.
 - c. Represent Tasks, durations and dependencies
 - d. Develop a time-line chart for the OLCRS project.
4. Imagine you are developing an online food delivery app. Explain how you would identify and differentiate between functional and non-functional requirements during the early stages of the project. Provide atleast five examples for each.
5. Compare and contrast the layered architecture and client-server architecture styles. In what types of systems would each be most effective, and what are the trade-offs involved in choosing one over the other? Support your answer with real-world examples.

[2]

[2]

[3]

[3]

6. A startup is developing software for an autonomous drone navigation system. It requires real-time response, high complexity, and high reliability. The estimated code size is 40 KLOC. The team has limited C++ experience, but the platform and tools are highly stable. The development will use modern tools and advanced methodologies, but the schedule is compressed by 20% due to a product launch.
- Using Embedded mode, estimate the EAF using following cost drivers: Required Reliability (1.40), Product Complexity (1.30), Programmer Capability (1.15), Virtual Machine Exp (1.10), Use of Tools (0.90), Required Dev Schedule (1.29) and other drivers are Nominal (1.0).
 - Compute the effort, schedule, and required personnel.

Software Projects	a	b	c	d
Organic	3.2	1.05	2.5	0.38
Semi-Detached	3.0	1.12	2.5	0.35
Embedded	2.8	1.20	2.5	0.32

- Your team is working on a multi-developer project where frequent changes in requirements are causing integration issues and version mismatches. Describe how Software Configuration Management (SCM) practices can help address these challenges. What strategies would you adopt to maintain consistency across versions and teams? Illustrate with required examples.
 - A startup is planning to bid for large government contracts, but they lack formal software development processes. The client mandates process maturity based on CMMI Level 3 or above. As the project manager, how would you guide the team to adopt a suitable process improvement path? Compare CMM and CMMI, and explain how achieving maturity levels impacts software delivery, team performance, and client confidence briefly.
- 9.a) Prepare the CFG for determining cyclomatic complexity of the below code snippet and design a test suite for checking each of the logical decisions.

```
public double calculate(int amount)
{
    double rushCharge = 0;
    if (nextday.equals("yes"))
    { rushCharge = 14.50; }
    double tax = amount * .0725;
    if (amount >= 1000)
    { shipcharge = amount * .06 + rushCharge; }
    else if (amount >= 200)
    { shipcharge = amount * .08 + rushCharge; }
    else if (amount >= 100)
    { shipcharge = 13.25 + rushCharge; }
    else if (amount >= 50)
    { shipcharge = 9.95 + rushCharge; }
    else if (amount >= 25)
    { shipcharge = 7.25 + rushCharge; }
    else
    { shipcharge = 5.25 + rushCharge; }
    total = amount + tax + shipcharge;
    return total;
} //end calculate
```

OR

- 9.b) i. During a software audit, several undocumented test results and informal review notes were discovered. Evaluate how formal inspection and auditing practices could enhance test reliability and accountability in such a scenario. Discuss in detail the long-term impact on software quality. [5]
- ii. After releasing a new feature that allows users to filter products by rating and price, several unrelated modules—like the shopping cart and payment gateway—start behaving inconsistently. As a test lead, how would you ensure that recent code changes have not unintentionally impacted other parts of the application? What approach would you use to identify which areas to re-test, and how would automation help in this situation? Illustrate and discuss in detail with relevant scenarios. [5]
- 10.a) Design an object-oriented solution with suitable notations for a ride-sharing application (e.g., Uber). Identify key classes, their responsibilities, relationships, and interactions. Explain in detail how principles like inheritance, encapsulation, and polymorphism apply.

OR

- 10.b) Create a Context modelling for an online food delivery system. The system includes customers placing orders, restaurants confirming them, delivery agents picking them up, and payments being processed. Propose a Level 0 and a detailed Level 1 DFD structure. Justify the process decomposition in your Level 1 DFD. How would you handle external entities and data stores to avoid clutter and confusion? Explain briefly.