

[Apr-24]

GITAM (Deemed to be University)
[CSEN1131]
GST/GSS/GSB/GSHS Degree Examination

VI SEMESTER

SOFTWARE ENGINEERING

(Effective for the admitted batch 2021-22)

Time: 2 Hours

Max. Marks: 30

Instructions: All parts of the unit must be answered in one place only.

Section-A

1. Answer all Questions:

(5×1=5)

- a) What is the purpose of DevOps?
- b) Suppose a project has strict deadlines and requires early delivery of core functionalities. Which methodology would you choose to meet these requirements?
- c) Describe the purpose of UML activity diagrams in system design.
- d) Explain the significance of functional testing in software development.
- e) Why is maintenance essential in the software development lifecycle?

Section-B

Answer the following:

(5×5=25)

UNIT-I

- 2. Analyze the impact of a systematic approach in addressing issues faced during the software development lifecycle, providing examples of how it improves project outcomes and efficiency

OR

- 3. Compare and contrast the Waterfall, Iterative, and Spiral software development processes, highlighting their unique characteristics, advantages, and potential limitations in different project scenarios

UNIT-II

4. Analyze the benefits of using prototyping in software development, providing examples of situations where prototyping is particularly advantageous and discussing potential challenges.

OR

5. Explain the concept of functional and non-functional requirements in software engineering, by considering library management system as an example and discussing their significance in the development process.

UNIT-III

6. Evaluate the advantages and challenges of using a top-down approach versus a bottom-up approach in software structural design. Provide examples of scenarios where each approach would be suitable, considering factors such as project size, complexity, and development team expertise.

OR

7. Discuss the key components and interactions depicted in a UML collaboration diagram, explaining how it illustrates communication and collaboration between system objects or components. Provide a scenario to demonstrate the creation and interpretation of collaboration diagrams.

UNIT-IV

8. Explain the concept of software metrics, distinguishing between process metrics and product metrics. Discuss the importance of using metrics to measure and improve software quality.

OR

9. Evaluate the impact of coding practices on the overall quality and maintainability of software products, discussing common coding standards, code reviews, and automated code analysis tools used in industry practices.

UNIT-V

10. Compare and contrast COCOMO and Function Point Analysis as software estimation techniques, discussing their principles, advantages, and limitations. Provide examples of scenarios where each technique would be suitable.

OR

11. Evaluate the role of risk analysis in software project management, discussing common risk identification techniques, risk assessment methods, and risk mitigation strategies. Provide examples of potential risks in software projects and how they can be managed effectively.

[VIS/124]