FMTH0301/Rev.5.3

Course Plan

Program: Bachelor of Engineering		**Semester-V**
Course Title: Software Engineering		**Course Code: 25ECSC301**
L-T-P: 3-0-0	**Credits: 3**	**Contact Hrs: 3 hrs/week**
ISA Marks: 50	**ESA Marks: 50**	**Total Marks: 100**
Teaching Hrs: 45	**Practical Hrs: --**	**Exam Duration: 3 hrs**

Semester: **V**	Year:2025-2026
Course Title: **Software Engineering**	Course Code: **25ECSC301**
Total Contact Hrs: 45	Duration of ESA: 3 Hours
ISA Marks: **50**	ESA Marks: **50**
Lesson Plan Author: Dr. Padmashree Desai, Dr. P.G. Sunitha Hiremath, Prof. Muskan Indikar and Prof. Shilpa Hosagoudra	Date: 01-09-2025
Checked By: Dr. Suvarna K	Date: 10-09-2025

Prerequisites: Knowledge of problem-solving with C programming, Data structures and algorithms and Object-oriented programming is essential.

Course Outcomes (COs):

At the end of the course, the student should be able to:

i. **Identify** the need to engineer a software system, software engineering principles and techniques to develop a solution.
ii. **Apply** requirements elicitation techniques to analyze system needs to prepare software requirements specifications.
iii. **Design** a software system using suitable design techniques for a given software requirements specification.
iv. **Perform** test planning and execution for a given system using relevant techniques and tools
v. Use tools to perform Software Development Life Cycle activities.

Course Articulation Matrix: Mapping of Course Outcomes (COs) with Program Outcomes (POs)

Course Title: Software Engineering	Semester: 5
Course Code: **25ECSC301**	Year: 2025-26

Course Outcomes (COs)/ Program Outcomes (POs)	**Program Outcome – PO***											**Specific**	
	1	**2**	**3**	**4**	**5**	**6**	**7**	**8**	**9**	**10**	**11**	**12**	**13**
i. Identify the need to engineer a software system, software engineering principles and techniques to develop a solution.	M												
ii. Apply requirements elicitation techniques to analyze system needs to prepare software requirements specifications.	M		H										
iii. Design a software system using suitable design techniques for a given software requirements specification.	M		M										
iv. Perform test planning and execution for a given system using relevant techniques and tools	M											H	
v. Use tools to perform Software Development Life Cycle activities					M						M		

Degree of compliance **L**: Low **M**: Medium **H**: High

Competency addressed in the Course and corresponding Performance Indicators

Competency	Performance Indicators	Course Specific Performance Indicator
1.4: Demonstrate the competency in Computer Science and Engineering (WK4)	**PI 1.4.1:** Apply systems engineering principles to design hardware and software solutions, adopt lifecycle models, and evaluate system performance, scalability, and reliability.	**CSPI 1.4.1.1:** Apply the fundamentals of software development lifecycle activities to streamline the software development.
3.1: Demonstrate an ability to design for a complex open-ended problem in engineering terms.	**PI 3.1.1:** Apply need analysis techniques to elicit requirements from stakeholders.	**CSPI 3.1.1.1:** Apply suitable process model to solve a real-world problem. **CSPI 3.1.1.2:** Apply requirements elicitation techniques to write functional and non-functional requirements
3.2: Demonstrate an ability to generate a diverse set of alternative design solutions.	**PI 3.2.2:** Build models/prototypes to develop a diverse set of design solutions (Theory).	**CSPI 3.2.2.1:** Apply UML techniques to design component/subsystem **CSPI 3.2.2.2:** Justify suitable architectural views/patterns to arrive at solution.
3.3: Demonstrate an ability to select the optimal design, considering cost effectiveness and environmental impact for further development	**PI 3.3.4:** Develop a system for the chosen design solutions (Lab).	**CSPI 3.3.4.1:** Build a software system for an identified functional and Non-functional requirements/user stories.
5.2: Demonstrate an ability to select and apply domain-specific tools, modelling techniques, and resources.	**PI 5.2.2:** Apply domain-specific tools and utilize computing platforms, libraries, and resources to solve engineering problems.	**CSPI 5.2.2.1:** Use project management tool to assign roles, responsibilities and plan for solve engineering problem. **CSPI 5.2.2.2:** Use relevant testing tool to automate the testing process. **CSPI 5.2.2.3:** Use CI/CD process to automate the continuous integration and continuous delivery/deployment in software development.
11.2: Demonstrate an ability to adapt new and emerging technologies in engineering practice.	**PI 11.2.1**: Adapt engineering methods and tools to incorporate	**CSPI 11.2.1.1:** Use appropriate tools for techniques for developing solution by self-learning.

	emerging technologies into professional practice.	
12.2: Ability to deploy reliable software systems through systematic testing and validation.	**PI 12.2.1:** Apply automated testing and debugging methods to detect software defects and improve overall system reliability	12.2.1.1: Develop test cases to validate the system for valid and invalid inputs.

Eg: 1.2.3: Represents Program Outcome ‘1’, Competency ‘2’ and Performance Indicators ‘3’.

Course Content

Course Code: 25ECSC301	Course Title: Software Engineering	
L-T-P: 3-0-0	Credits: 3	Contact Hrs: 3hrs/week
ISA Marks: 50	ESA Marks: 50	Total Marks: 100
Teaching Hrs: 45	Tutorial/Practical: --	Exam Duration: 3 Hrs

Contents		Hrs
Unit-I		
1	**Software Engineering Process and Requirements Engineering**: Software process models, Process activities, Agile versus traditional method comparisons. Functional and non-functional requirements, Use case diagrams and use case description, software requirements, requirement specification, requirements engineering processes and requirements elicitation.	07 hrs
2	**Agile Frameworks:** Agile Manifesto and its 12 Principles, Benefits, and Challenges of Agile, Scrum, Extreme Programming (XP), SAFe (Scaled Agile Framework)	05 hrs
3	**Agile Planning and Estimation:** User Stories and Epics, INVEST Model, Story Points and Planning Poker, Release Planning and Roadmaps, Velocity and Capacity Planning.	06 hrs
Unit - II		
4	**System Modeling & Design:** Context models, Interaction models, Structural models, and Behavioral models. Architectural Design Decisions, Architectural views, Architectural patterns, Application Architectures.	09 hrs
5	**Testing:** Software testing techniques, The Agile lifecycle and its impact on testing, Functionality Testing, UI Testing, Performance Testing, Security Testing, Test Driven Development (TDD).	08 hrs
Unit-III		
6	**Introduction to DevOps and CI/CD:** DevOps Principles, Lifecycle, Delivery pipeline, Technical challenges, and DevOps Tools.	05 hrs
7	**Continuous integration and continuous delivery (CI/CD):** Essentials of continuous integration, Jenkins architecture, Jenkins security management, Jenkins master-slave architecture, Jenkins delivery pipeline.	05 hrs

Text Books

1. Software Engineering by Ian Sommerville , 10th edition, Pearson publication-24 May 2017
2. Continuous Delivery: Reliable Software Releases through Build, Test, and Deployment Automation by Jennifer Davis & Ryn Daniels, 1st Edition, Addison-Wesley Signature Series (Fowler), 27 July 2010

Reference Books:

1. Robert C. Martin, "Agile Software Development: Principles, Patterns, and Practices" by, Person New International Edition 17 July 2013.
2. Andrew Stellman, Jennifer Greene - Learning Agile: Understanding Scrum, XP, Lean, and Kanban, O Reilly, 2015
3. Software Engineering: A Practitioner’s Approach, 8/eby Bruce R. Maxim, Roger S. Pressman , McGraw Hill Education; 19 March 2019
4. Software Engineering at Google: Lessons Learned from Programming Over Time by Titus Winters , Tom Manshreck & Hyrum Wright, 1st edition , O'Reilly Media - 28 February 2020

Evaluation Scheme

In-Semester Assessment Scheme

Assessment	Conducted for marks	Weightage in Marks
ISA-1 (Theory)	40	15
ISA-2 (Theory)	40	15
Self-(Learning_Activity)	20	20
Total		50

End-Semester Assessment Scheme

Assessment	Conducted for marks	Weightage in Marks
Theory	100	50

Course Unitization for ISA and ESA

Topics / Chapters	Teaching Credits	No. of Questions in ISA 1	No. of Questions in ISA 2	No. of Questions in Activity	No. of Questions in ESA
Unit I					
1. Software Engineering Process and Requirements Engineering	7	1.00	--	-	1.00
2. Agile Frameworks	6	1.00	--	-	1.00
3. Agile Planning and Estimation	5	1.00	--	-	1.00
4. Unit II					
5. System Modeling & Design	9	--	1.50	-	1.50
6. Testing	8	--	1.50	-	1.50
7. Unit III					
7. Introduction to DevOps and CI/CD	5	--	--	-	1.00
8. Continuous integration and continuous delivery (CI/CD):	5	--	--	-	1.00

Note

1. Each Question carries 20 marks and may consist of sub-questions.
2. Mixing of sub-questions from different chapters within a unit (only for Unit I and Unit II) is allowed in ISA I, II and ESA.
3. Answer 5 full questions of 20 marks each (two full questions from Unit I, II and one full question from Unit III) out of 8 questions in ESA.

Date:09-09-25

Head, CSE

Course Assessment Plan

Course Title: Software Engineering			Code: 25ECSC301		
Course outcomes (COs)	Weightage in assessment	Assessment Methods			
		ISA-1	ISA-II	Activity	ESA
1. Identify the need to engineer a software system, software engineering principles and techniques to develop a solution.	20%	✓		✓	✓
2. Apply requirements elicitation techniques to analyze system needs to prepare software requirements specifications.	30%		✓	✓	✓
3. Design a software system using suitable design techniques for a given software requirements specification.	30%	✓		✓	✓
4. Perform test planning and execution for a given system using relevant techniques and tools	30%		✓	✓	✓
5. Use tools to perform Software Development Life Cycle activities	20%			✓	✓

Self-Learning(45hrs)

Sl. No	Activity	Objective	Self-Learning Hours	Assessment	Weightage (Marks) In %	Deliverable/ Evidences
1	Concept understanding Scrum project management and roles for software development Chapter 2	Prepare sprints and user stories for incremental delivery	05	MCQ	10%	Quiz and Ans along with score CO2 PI: 1.4.1
2	Tool learning: Project management and planning tool learning-Trello Chapter 3	Use tool for managing tasks among team members	10	Demonstration of tool for each user story based on role (Review-1)	20%	Proof report in portfolio CO3 PI: 3.1.1
3	Tool learning: UML tool Chapter 4	Design using UML notations(class diagram, sequence diagram, usecase diagram)	05	Design the system (Review-2)	10%	Proof report in portfolio CO2 PI: 3.3.4
4	Tool Learning: Software Testing Tool (selenium) and implementation Chapter 5	Test each user story using tool	20	Implement the scrum and test the automation (Review-3)	50%	Prepare testing report And portfolio CO4 PI: 12.2.1
5	Tool learning: Continuous integration and Continuous development tool learning –JENKIN Chapter 6 and 7	Demonstrate the pipe line for software development using JENKIN	05	Demonstration of pipe line	05(10%)	Prepare report and portfolio CO5 PI: 5.2.2
Total Self Learning hrs			45 hrs		100% (scale down to 20)	Submission of code (github link) and portfolio

Integrated course activity: Develop web application applying Software Engineering principles, Integrated with Web technology course			
Review-1: Sprint Review	**Marks: 15**		**5th Week**
Parameters	**5** **(Excellent)**	**(4-3)** **(Good)**	**(2-1)** **(Satisfactory)**
Product backlog with clarity	• Scrum Roles are allotted. (Product Owner, Scrum Master, Development Team) • Defined Problem statement is precise. • Program Increments and their objectives are defined clearly. • Epics and features for each increment are stated clearly. • Identified the users of the system	• Scrum Roles are allotted. (Product Owner, Scrum Master, Development Team) • Defined Problem statement is precise. • Program Increments and their objectives are defined clearly. • Epics stated are clear but unable to list features for each increment. • Identified the users of the system	• Scrum Roles are allotted. (Product Owner, Scrum Master, Development Team) • Defined Problem statement is precise. • Program Increments are defined but objectives are not clear. • Epics stated are clear but unable to list features for each increment. • Identified the users of the system
Sprint Planning	• Goals set for the each of the sprint are clear. • User stories are written with story points for each of the identified features. • Features are set for each of the sprint based on the priority. • Acceptance criteria are set for 80-100% of the user story.	• Goals set for the each of the sprint are clear. • User stories are written with story points for 80-100% of the identified features. • 80-100% of the features are set for each of the sprint based on the priority. • Acceptance criteria are set for 60-80% of the user story.	• Goals set for the each of the sprint are clear. • User stories are written with story points for 60-80% of the identified features. • 60-80% of the features are set for each of the sprint based on the priority. • Acceptance criteria are set for <60 of the user story.
Tool usage & updates	• Board is set with columns: To Do → In Progress → Done • Able to show all the Stories with task breakdown • Able to show all Visual updates as the sprint progresses	• Board is set with columns: To Do → In Progress → Done • Able to show 60-80% of the Stories with task breakdown • Able to show 60-80% of Visual updates as the sprint progresses.	• Board is set with columns: To Do → In Progress → Done • Able to show <60% of the Stories with task breakdown. • Able to show <60% of Visual updates as the sprint progresses.

Daily Scrum reflections	• Each team member maintained 3 daily logs for sprint 1 as per the format • Listed the blockers	• Each team member maintained 3 daily logs for 80% of sprint 1 as per the format • Listed the blockers	• Each team member maintained 3 daily logs for the 60-80% of the sprint 1 as per the format • Listed the blockers

Review-2: Design (Front and Back end)	**Marks: 10 marks**		**9th week**
Parameters	**5 (Excellent)**	**(4-3) (Good)**	**(2-0) (Satisfactory)**
UML Techniques	Able to apply relevant UML techniques to model the complete system	Able to apply relevant UML techniques to model 60-80% of the system	Able to apply relevant UML techniques to model <60% of the system
Front End design	• Designed web page with all the relevant controls such as text boxes, drop down, menus etc according to the requirements. • Used appropriate color scheme for all the web pages • Validated all fields.	• Designed web page with 70-80% relevant controls such text boxes, drop down, menus etc according to the given problem statement. • Use of appropriate color scheme for the web page • Validated 70-80% of the fields.	• Designed web page with 50-60% relevant controls such text boxes, drop down, menus etc according to the given problem statement. • Use of appropriate color scheme for the web page • Validated 50-60% of the fields. .
Back End Design	• Connection with Database is established. • Performing CRUD operations on DB tables through front end application. • Configuring Database connections correctly with no errors,	• Connection with Database is established. • Only 80% queries working with DB. • Configuring Database connections partially	• Unable to establish connection with database • Configuring Database connections but with errors • No appropriate tool used for front end design

Review-3: System Testing Marks: 20 13th Week			
Parameters	**5** **(Excellent)**	**4** **(Good)**	**3** **(Satisfactory)**
Performance Tracking	• Able to Assess user stories with INVSET criteria • Able track completion (80-100%) of work using Burnup chart. • Able track pending work (10-20%) using Burndown chart • Able to maintain average velocity of team.	• Able to Assess user stories with INVSET criteria • Able track completion (60-80%) of work using Burnup chart. • Able track pending work (20-40%) using Burndown chart • Average velocity of team is decreased	• Able to Assess user stories with INVSET criteria • Able track completion (<60%) of work using Burnup chart. • Able track pending work (>40%) using Burndown chart • Average velocity of team is decreased
Writing Test cases.	• Able to write test cases for sprints that includes test ID, title, objective, preconditions, test data, steps, expected result, priority, type, and traceability	• Able to write test cases for 60-80% of the sprints that includes test ID, title, objective, preconditions, test data, steps, expected result, priority, type, and traceability	• Able to write test cases for <60% of the sprints that includes test ID, title, objective, preconditions, test data, steps, expected result, priority, type, and traceability
Implementation of test cases using a testing framework(Junit/ Selenium)	• Able to set up the Selenium Environment • Able to write program logic for all test cases using the assertion methods provided by the tool.	• Able to set up the Selenium Environment • Able to write program logic for 60-80% of test cases using the assertion methods provided by the tool.	• Able to set up the Selenium Environment • Able to write program logic for <60% of test cases using the assertion methods provided by the tool.
• Running Selenium scripts on different browsers	• Able to execute all Selenium tests within Eclipse IDE and view test results in the console. • Able to generate >80% of test reports • Able to perform cross-browser testing by running Selenium tests on different browsers (e.g., Chrome, Firefox) using WebDriver.	• Able to execute 60-80% of Selenium tests within Eclipse IDE and view test results in the console. • Able to generate 60-80% of the test reports • Unable to perform cross-browser testing.	• Able to execute <60% of Selenium tests within Eclipse IDE and view test results in the console. • Able to generate <60% of the test reports • Unable to perform cross-browser testing.

Assessment of course activity

Description	Week	Marks	CO	BL	PI
Review-1(sprint development)	5th	15	CO1, CO2	L3	PI 1.4.1 PI 3.1.1
Review-2 (Design front end & Back end)	9th	10	CO3	L3	PI 3.2.2
Review-3 (System testing)	13th	20	CO4	L3	PI 12.2.1 PI 5.2.2
Demonstrate an application using DevOPs principles and tool	11th	05	CO1 CO5	L3	PI 1.4.1 PI 5.2.2
Total		50 scale down to 20)			

Chapter wise Plan

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **1. Software Engineering Process and Requirements Engineering**	Planned Hours: **07 hrs**

Learning Outcomes: -

At the end of the topic, the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Select an appropriate process model for a software system being built.	1	L3	1.4
2. Write functional and nonfunctional requirements for the given case study	2	L4	3.1
3. Apply requirements engineering activities: Elicitation, analysis and validation for collecting requirements from a customer	2	L3	3.1
4. Write the Use Case diagram with its description for a given scenario	2	L3	3.1

Lesson Schedule Class No. - Portion covered per hour / Class
1. Software process models, Process activities
2. Agile versus traditional method comparisons
3. Functional Requirements
4. Non-functional requirements, software requirements specification
5. Use case diagrams and use case description
6. Requirements engineering process
7. requirements elicitation

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. Suggest the most appropriate generic software process model that might be used for the development of the following systems. Justify your answer. • A system to control anti-lock braking in a car. • A virtual reality system to support software maintenance. • A university accounting system that replaces an existing system. • An interactive travel planning system that helps users plan journeys with the lowest environmental impact	TLO1	L3	1.4.1
2. Compare the strengths and weakness of the traditional development methods with Agile.	TLO1	L2	1.4.1
3. Which of the development process models discussed would you follow for the following projects and justify your answers? An online store. • A Toll Collection System for Highway Authorities to collect Toll at multiple entry points between PUNE and HUBLI. • A new operating system for Mobile Devices to reach market in a defined amount of time, which is short	TLO1	L3	1.4.1
4. Write five functional requirements and two non-functional requirements for a toll collection system established by the State Highway Authority to collect tolls at multiple entry points on all highways in the state. The system should be able to generate intelligent reports doing data analysis.	TLO2	L4	3.1.1
5. Write Functional requirements for the Watching Videos of software Engineering course in LMS.	TLO2	L4	3.1.1
6. Correct the following incorrect statements (if any) and mention the appropriate metric (relating to Nonfunctional requirement) for the same: a) User must be able to see the homepage of the application immediately whenever he logs in to the LMS Application. b) All the options in the LMS application must take less time for navigation. c) It should be easy for the students to use the LMS application. d) Students can use the LMS application any time.	TLO2	L3	3.1.1
7. Draw the diagram for requirement engineering process and explain the significance of requirement elicitation process.	TLO 3	L2	3.1.1

8. Discuss the significance of Ethnography in requirement elicitation process.	TLO3	L3	3.1.1
9. Draw a use case diagram for WhatsApp application, consisting of at least two actors and 5 use cases/scenarios. Write a use case description using a template for one of the use cases from your list.	TLO4	L3	3.1.1
10. Draw a use case diagram for any mobile banking system of your choice and give a use case description about the same.	TLO4	L3	3.1.1

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **2. Agile Frameworks**	Planned Hours: **5 hrs**

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Discuss the rationale for agile software development methods	CO1	L2	1.4
2. Apply extreme programming and scrum model of agile software development for a given scenario	CO1	L3	3.1
3. Discuss Lean Software Development and SAFe (Scaled Agile Framework)	CO1	L3	1.4

Lesson Schedule Class No. - Portion covered per hour / per Class
1. Agile Manifesto, its 12 Principles, Benefits, and Challenges of Agile
2. Scrum and Kanban
3. Extreme Programming (XP)
4. Lean Software Development
5. SAFe (Scaled Agile Framework)

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. Differentiate between plan-driven and agile methods of software development.	TLO1	L2	1.4.1
2. Describe how SAFe impacts the roles and responsibilities of stakeholders involved, including Release Train Engineers, Product Owners, and Agile teams.	TLO3	L3	1.4.1
3. Explain the key components and core principles of the Scaled Agile Framework (SAFe). Illustrate how SAFe effectively coordinates multiple agile teams across different organizational levels to deliver continuous value.	TLO3	L3	1.4.1
4. Design two sprints for hospital software development clearly mentioning role of the Scrum Master, Product Owner and Developer using Scrum framework	TLO2	L3	3.1.1
5. How principles of extreme programming help in improving software development process	TLO2	L3	1.4.1

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **3. Agile Planning and Estimation**	Planned Hours: **6 hrs**

Learning Outcomes:-

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Explain importance of user stories and story points in agile software development.	CO2	L2	**3.1.1**
2. Write user stories and story points for an identified problem	CO2	L4	3.1.1
3. Perform release planning and write road maps for developing solution	CO2	L3	3.1.1
4. Discuss velocity and capacity planning for the agile software development	CO2	L2	3.1.1

Lesson Schedule Class No. - Portion covered per hour / per Class
1. User Stories and Epics
2. Use Stories and Epics contd..
3. INVEST Model
4. Story Points and Planning Poker
5. Release Planning and Roadmaps
6. Velocity and Capacity Planning

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. Why INVEST model is necessity for planning sprint	TLO1	L2	3.1.1
2. Write user stories and story points for student information system	TLO2	L4	3.1.1
3. Write release plan, roadmaps for delivery of sprints for student information system	TLO3	L3	3.1.1
4. Calculate velocity for each sprint delivered for student information system	TLO4	L3	3.1.1

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **4. System Modeling & Design**	Planned Hours: **9hrs**

Learning Outcomes: -

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Discuss different types of models viz. context, interaction, structure and behavior in modelling of a system	CO3	L2	1.4
2. Analyze the system with different UML diagrams in modelling systems (activity diagrams, use case diagrams, sequence diagrams, class diagrams, and state diagrams).	CO3	L4	3.2
3. Design the architecture of a software system using suitable architectural patterns and justify the selection of a specific architectural pattern based on the system's requirements.	CO3	L4	3.3

4. Use different types of application architectures to provide a solution.	CO2	L3	3.2

Lesson Schedule Class No. - Portion covered per hour / per Class
1. Introduction to system modelling:
2. Context models and Interaction Models
3. Structural models
4. Behavioral models
5. Architectural Design Decisions
6. Architectural views
7. Architectural patterns
8. Application Architectures
9. Application Architectures contd..

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. In the tiger monitoring project at Anshi National Park drones are used. Forest Reserve Officers operate these drones from the command center. They have unique IDs to validate their access to the drones. Drones have a communication module (to transmit and receive information) and a rotary module (to control drone movement). Draw escribe sequence diagram for moving the drone in the forward direction	TLO3	L3	3.2.2
2. Consider a web-based online bookstore system which sells books, music CDs, and software. Typically, a customer first logs on to the system, entering a customer ID and password. The customer can then browse for titles or search by keyword. The customer puts some of the titles into a "shopping cart" which keeps track of the desired titles. When the customer is done shopping, he/she confirms the order, shipping address, and billing address. The bookstore system then issues a shipping order, bills the customer, and issues an electronic receipt. At the end of the transaction, the customer logs off. "For this scenario design the class diagram.	TLO3	L4	3.3.4

3. Explain why it is important to model the context of a system that is being developed. Give two examples of possible errors that could arise if software engineers do not understand the system context.	TLO2	L3	3.2.2
4. Draw state diagrams of the control software for the DVD player.	TLO3	L3	3.2.2
5. As part of the embedded design team at Microchips, you are asked to present a design model for a new privacy sensor that needs to be added as a new feature in the next release of a phone. The following rules are given to you. i The sensor is active only when the user needs it. ii It has two modes: fully secure and partial secure iii When activated in full secure mode it will halt all background OS operations, deactivates the GPS sensor iv When activated in partially secure mode it will only halt background OS operations Which system design model will you consider? Justify and design the model	TLO2	L4	3.3.4
6. Elucidate the different UML notations available for Structural Models.	TLO3	L2	1.4.1
7. Differentiate between state diagram and sequence diagram. Give example for the same.	TLO3	L2	3.2.2
8. Select and justify suitable architectures for the following software systems and explain the rationale behind your suggestion: i) Lip-reading system ii) smart notice board iii) analytical module for customers buying in online website	TLO4	L4	3.2.2
9. Consider an interactive Web site which provides many different features to perform various tasks. Show that the architecture for this can be represented as a shared-data style as well as client-server style. Which one will you prefer and why?	TLO4	L3	3.3.4
10. Consider a compiler used to compile a source code to generate an object code. Apply suitable application architecture style draw the architecture diagram.	TLO4	L4	3.2.2
11. "HealthTrack" is a healthcare management system designed to integrate patient records, appointment scheduling, telemedicine services, and prescription management. Analyze how various architectural patterns can be applied to build it? Choose any two architectural patterns and draw architecture diagrams.	TLO3	L3	3.3.4

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **5. Software Testing**	Planned Hours: 8 **hrs**

Learning Outcomes: -

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Explain different types of testing performed during development of software system.	CO4	L2	12.2
2. Analyze a given scenario and write test cases using test-driven development	CO4	L4	12.2
3. Differentiate between Functionality, Performance, Security, UI testing processes.	CO4	L3	12.2
4. Describe the Agile Lifecycle and Its Impact on Testing.	CO4	L3	12.2

Lesson Schedule Class No. - Portion covered per hour / per Class
1. Software Testing Techniques: Unit, integration, system, acceptance testing
2. Software testing Techniques: Boundary value Analysis, Equivalence Class Partitioning
3. Software testing Techniques: black box and white box testing
4. The Agile lifecycle and its impact on testing
5. Functionality Testing, UI Testing
6. Performance Testing, Security Testing
7. Test Driven Development (TDD)
8. Test Driven Development (TDD)

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. What are the different levels of testing and the goals of different levels? For each level, specify which of the testing approaches is most suitable.	TLO1	L2	12.2.1
2. An Online Journal Portal system need to be developed to manages subscriptions for readership, online submission of papers from authors, online review of papers from experts. Write possible test cases using test driven approach.	TLO2	L4	12.2.1
3. You need to find largest of N numbers. Design test cases using path testing technique.	TLO2	L4	12.2.1
4. Given the following fragment of code, how many tests are required for 100% decision coverage? if width > length thenbiggest_dimension = width if height > width thenbiggest_dimension = height end_if elsebiggest_dimension = length if height > length thenbiggest_dimension = height end_if end_if	TLO2	L4	12.2.1
5. You have designed test cases to provide 100% statement and 100% decision coverage for the following fragment of code. if width > length then biggest_dimension = width else biggest_dimension = length end_if The following has been added to the bottom of the code fragment above. print "Biggest dimension is " &biggest_dimensionprint "Width: " & width print "Length: " & length How many more test cases are required?	TLO2	L4	12.2.1
6. What is the difference between negative and positive testing? Give example for both	TLO1	L3	12.2.1
7. You have written a module to replace old string with new string if there is a match. Write test cases to perform the same.	TLO2	L4	12.2.1
8. Why different testing techniques are most important for system	TLO3	L3	12.2.1
9. A program reads three integer values. The three values are interpreted as representing the lengths of the sides of a triangle. The program prints a message that states whether the triangle is scalene, isosceles, or equilateral. Develop a set of test cases (at least 6) that you feel will adequately test this program.	TLO2	L4	12.2.1
10. Describe the phases of the Agile lifecycle, highlighting the key testing activities carried out during each phase.	TLO4	L3	12.2.1
11. Explain in detail how adopting Agile practices transforms traditional testing processes, roles, and responsibilities.	TLO4	L3	12.2.1
12. Discuss the advantages and challenges that testing teams face when working within Agile projects.	TLO 4	L3	12.2.1

Course Code and Title: **25ECSC301 / Software Engineering**	
Chapter Number and Title: **6. Introduction to DevOps**	Planned Hours: **5 hrs**

Learning Outcomes: -

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Explain DevOps principles and environment	CO3	L2	1.4
2. Explain DevOps Lifecycle and stages	CO3	L2	1.4
3. Identify delivery pipeline and technical challenges for an application	CO3	L3	3.2
4. Explain DevOps tool	CO5	L2	5.2

Lesson Schedule Class No. - Portion covered per hour / per Class
1. DevOps Principles, Benefits of working in a DevOps environment
2. Lifecycle, stages
3. Delivery pipeline
4. Technical challenges and
5. DevOps Tools

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
Explain the DevOps Principles	TLO4	L2	1.4.1
Why DevOps lifecycle is suitable for business applications. Justify the answer.	TLO3	L2	3.2.2
Identify a possible DevOps pipeline for a simple calculator application.	TLO4	L3	3.2.2
Discuss any one example of real-world scenarios where the DevOps lifecycle has significantly improved software delivery and operations.	TLO1	L3	3.2.2

Course Code and Title: **22ECSC301 / Software Engineering**	
Chapter Number and Title: **7. Continuous integration and continuous delivery (CI/CD)**	Planned Hours: **5 hrs**

At the end of the topic the student should be able to:

Topic Learning Outcomes	COs	BL	CA Code
1. Essentials of continuous integration	CO3	L2	1.4
2. Jenkins architecture, Jenkins security management	CO3	L2	3.2
3. Differentiate between traditional architecture and Jenkins master-slave architecture,	CO3	L3	3.2
4. Validating the Jenkins delivery pipeline and authentication	CO3	L3	3.2

Lesson Schedule Class No. - Portion covered per hour / per Class
1. Essentials of continuous integration
2. Jenkins architecture,
3. Jenkins security management
4. Jenkins master-slave architecture
5. Jenkins delivery pipeline with example

Review Questions

Sl.No. - Questions	TLOs	BL	PI Code
1. Discuss the benefits of CI/CD	TLO1	L2	1.4.1
2. Why Jenkins master-slave architecture is suitable for business applications. Give a reason by taking an appropriate example.	TLO3	L3	3.2.2
3. How do you validate the Jenkins delivery pipeline with different authentication processes?	TLO4	L3	3.2.2
4. What measures can you take to secure the Jenkins build process against threats such as code injection, dependency attacks, or unauthorised changes to the built environment?	TLO2	L3	3.2.2

Model Question Paper for Minor Examination (ISA-I)		
Course Code: 25ECSC301		Course Title: Software Engineering
Duration (H:M):1:15		
Max. Marks: 40		
Note: Answer any two full questions		

Q.No.	Questions	Marks	CO	BL	PO	PI Code
Q1a	Propose and justify process model for each of the following software projects: - A mobile banking application with frequent updates - Software for a new computed tomography (CT scan) scanning machine. Above systems need to be developed to meet customer requirements. You are expected to identify at least two characteristics for each of the software that ideally suits the choice of a particular process model. (Ch 1)	08	CO1	L3	1	1.4.1
Q1b	Differentiate between plan driven and agile software development. List out advantages and disadvantages for each.(Ch 2)	07	CO1	L3	1	1.4.1
Q1c	Discuss how INVEST criteria (any three) are used to evaluate user stories by giving an example for each. (Ch3)	05	CO3	L2	10	3.1.1
Q2a	Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system: An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a	08	CO2	L3	1	3.1.1

	personal identifier. When the credit transaction has been validated, the ticket is issued.					
Q2b	Illustrate how SAFe effectively coordinates multiple agile teams across different organizational levels to deliver continuous value. Write key components and core principles of SAFe (Ch 3.)	07	CO2	L3	1	3.1.1
Q2c	Discuss four core values of the Agile manifesto (ch2)	05	CO1	L2	1	1.4.1
Q3a	Write at least six functional requirements for one of the software mentioned in Q 1(a). Write two verifiable non-functional requirements. (Ch 1)	08	CO2	L4	3	3.1.1
Q3b	Design a sprint for developing the software described in Q2.a clearly mentioning role of the Scrum Master, Product Owner and Developer using Scrum framework (ch 3)	08	CO2	L4	3	3.1.1
Q3c	Write in brief about refactoring and pair programming with example (ch2)	04	CO1	L2	1	1.4.1

Model Question Paper for Minor Examination (ISA-II)		
Course Code: 25ECSC301		Course Title: Software Engineering
Duration (H:M):1:15		
Max. Marks: 40		
Note : Answer any two full questions		

Q.No	Questions	Marks	CO	BL	PO	PI Code
Q1a	SMART DOOR lock system is a security system that utilizes a unique password that can only be used once to unlock the door. The system typically involves a door lock mechanism connected to a microcontroller that generates an OTP. The OTP is usually sent to the user's mobile device via SMS or a mobile app. The user	08	CO3	L3	3	3.2.2

	then enters the OTP into the keypad or mobile app, which sends a signal to the microcontroller to unlock the door. This smart lock generates a new password every time you unlock it, which further enhances your security level. Draw a **state diagram** for automating stages of SMART DOOR lock system. (Ch 4)					
Q1b	Write seven test cases for ticket reservation and cancellation in an online railway reservation system using Test Driven Development approach, mentioning testid, test input and expected output (Ch5)	7	CO4	L4	12	12.2.1
Q1c	What is Agile Performance Testing? Explain, how you would conduct it for any UPI (Unified Payment Interface) based mobile application. (ch5)	05	CO4	L3	12	12.2.1
Q2a	Differentiate between the following: • Black box testing and White box testing • Boundary value analysis and Equivalence Class partitioning (ch5)	8	CO4	L3	12	12.2.1
Q2b	A software system will be built to allow drones to autonomously herd cattle in farms. These drones can be remotely controlled by human operators. Draw appropriate two architectures and justify how **multiple architectural patterns** can fit together to help build this kind of system. (ch4)	07	CO3	L4	3	3.3.4
Q2c	Discuss application architecture with example(ch4)	05	CO3	L2	3	3.2.2
Q3a	A software system will be built to allow drones to autonomously herd cattle in farms. These drones can be remotely controlled by human operators. Write a sequence diagram for any identified requirements clearly showing objects, time line and msgs(ch4)	08	CO3	L4	3	3.2.2
Q3b	An online Journal portal system need to be developed to manage subscriptions for readership, online submission of papers from authors, online review of papers from experts. Write possible test cases using test driven approach. (ch5)	07	CO4	L4	3	3.2.2
Q3c	Why system modelling is important while designing a system. Illustrate with examples.(ch4)	05	CO3	L3	10	1.4.1

Model Question Paper for End Semester Assessment (ESA)		
Course Code: 25ECSC301		Course Title: Software Engineering
Duration:3 Hours		
Max. Marks: 100		
Note: *Answer any TWO full questions from UNIT-I, any TWO full questions from UNIT-II*		

Q.No.	Questions	Marks	CO	BL	PO	PI Code
Q1a	Propose and justify process model for each of the following software projects: • A mobile banking application with frequent updates • Software for a new computed tomography (CT scan) scanning machine. Above systems need to be developed to meet customer requirements. You are expected to identify at least two characteristics for each of the software that ideally suits the choice of a particular process model. (Ch 1)	08	CO1	L3	1	1.4.1
Q1b	Differentiate between plan driven and agile software development. List out advantages and disadvantages for each.(Ch 2)	07	CO1	L3	1	1.4.1
Q1c	Discuss how INVEST criteria (any three) are used to evaluate user stories by giving an example for each. (Ch3)	05	CO3	L2	10	3.1.1
Q2a	Discover ambiguities or omissions in the following statement of requirements for part of a ticket-issuing system: An automated ticket-issuing system sells rail tickets. Users select their destination and input a credit card and a personal identification number. The rail ticket is issued and their credit card account charged. When the user presses the start button, a menu display of potential destinations is activated, along with a message to the user to select a destination. Once a destination has been selected, users are requested to input their credit card. Its validity is checked and the user is then requested to input a	08	CO2	L3	1	3.1.1

	personal identifier. When the credit transaction has been validated, the ticket is issued.					
Q2b	Illustrate how SAFe effectively coordinates multiple agile teams across different organizational levels to deliver continuous value. Write key components and core principles of SAFe (Ch 3.)	07	CO2	L3	1	3.1.1
Q2c	Discuss four core values of the Agile manifesto (ch2)	05	CO1	L2	1	1.4.1
Q3a	Write at least six functional requirements for one of the software mentioned in Q 1(a). Write two verifiable non-functional requirements. (Ch 1)	08	CO2	L4	3	3.1.1
Q3b	Design a sprint for developing the software described in Q2.a clearly mentioning role of the Scrum Master, Product Owner and Developer using Scrum framework (ch 3)	08	CO2	L4	3	3.1.1
Q3c	Write in brief about refactoring and pair programming with example (ch2)	04	CO1	L2	1	1.4.1
Q4a	SMART DOOR lock system is a security system that utilizes a unique password that can only be used once to unlock the door. The system typically involves a door lock mechanism connected to a microcontroller that generates an OTP. The OTP is usually sent to the user's mobile device via SMS or a mobile app. The user then enters the OTP into the keypad or mobile app, which sends a signal to the microcontroller to unlock the door. This smart lock generates a new password every time you unlock it, which further enhances your security level. Draw a **state diagram** for automating stages of SMART DOOR lock system. (Ch 4)	08	CO3	L3	3	3.2.2
Q4b	Write seven test cases for ticket reservation and cancellation in an online railway reservation system using Test Driven Development approach, mentioning testid, test input and expected output (Ch5)	7	CO4	L4	12	12.2.1
Q4c	What is Agile Performance Testing? Explain, how you would conduct it for any UPI (Unified Payment Interface) based mobile application. (ch5)	05	CO4	L3	12	12.2.1
Q5a	Differentiate between the following: • Black box testing and White box testing • Boundary value analysis and Equivalence Class partitioning (ch5)	8	CO4	L3	12	12.2.1

Q5b	A software system will be built to allow drones to autonomously herd cattle in farms. These drones can be remotely controlled by human operators. Draw appropriate two architectures and justify how **multiple architectural patterns** can fit together to help build this kind of system. (ch4)	07	CO3	L4	3	3.3.4
Q5c	Discuss application architecture with example(ch4)	05	CO3	L2	3	3.2.2
Q6a	A software system will be built to allow drones to autonomously herd cattle in farms. These drones can be remotely controlled by human operators. Write a sequence diagram for any identified requirements clearly showing objects, time line and msgs(ch4)	08	CO3	L4	3	3.2.2
Q6b	An online Journal portal system need to be developed to manage subscriptions for readership, online submission of papers from authors, online review of papers from experts. Write possible test cases using test driven approach. (ch5)	07	CO4	L4	3	3.2.2
Q6c	Why system modelling is important while designing a system. Illustrate with examples.(ch4)	05	CO3	L3	10	1.4.1
UNIT-III						
Q7a	How will the DevOps environment be applied to the development of the banking system? Discuss the flow of system implementation.	10	CO3	L3	1	1.4.1
Q7b	Consider that you are developing a UPI based mobile application. Do you feel that incorporating DevOps principle will ensure faster delivery of the product? Discuss. (Ch6)	10	CO3	L3	1	3.2.2
Q8a	Apply Jenkins master-slave architecture for the banking system and a diagram for the same.	8	CO3	L3	5	3.2.2
Q8b	How Jenkins delivery pipeline is different from other traditional development models. Justify with an example and explain the authentication process.	8	CO3	L3	5	3.2.2