

SEMESTER V

UECSA20 - ELECTIVE - I A: SOFTWARE ENGINEERING

Year: III	Course Code: UECSA20	Title of the Course: Elective - I A:Software Engineering	Course Type: Theory	Course Category: Elective	H/W 5	Credits 5	Marks 40+60
----------------------------	--------------------------------	--	-------------------------------	-------------------------------------	-----------------	---------------------	-----------------------

Course Learning Objectives (CLO)

1. On learning this paper students will gain the knowledge of developing software with its techniques.
2. Be successful professionals in the field with solid fundamental knowledge of software engineering.
3. Capable of team and organizational leadership in computing project settings and have a broad understanding of ethical application of computing-based solutions to societal and organizational problems.
4. Be agile software developers with a comprehensive set of skills appropriate to the needs of the dynamic global computing-based society.
5. Acquire skills and knowledge to advance their career, including continually upgrading professional, communication, analytic and technical skills.

Course Outcomes (COs)

The Learners will be able to

1. Apply the software engineering life cycle by demonstrating competence in communication, planning, analysis, design, construction and deployment.
2. Discuss the function effectively on a team whose members together provide leadership, create a collaborative and inclusive environment, establish goals, plan tasks and meet objectives.
3. Manage the time, processes and resources effectively by prioritizing competing demands to achieve personal and team goals Identify and analyzes the common threats in each domain
4. Understand architectural design in order to minimize the risks and errors.
5. Test the techniques for ensuring high quality software and understand the capabilities of cost estimation.

CO	PSO					
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	M	M	M	M	M	H

CO2	M	H	M	M	H	L
CO3	M	L	L	H	M	M
CO4	L	L	M	M	M	M
CO5	H	M	L	H	M	L

(Low -L, Medium -M, High-H)

CO	PO					
	1	2	3	4	5	6
CO1	H	M	L	H	H	L
CO2	H	M	M	L	L	M
CO3	M	H	M	M	M	L
CO4	L	M	L	M	M	M
CO5	M	L	H	M	H	L

(Low -L, Medium -M, High-H)

Course Syllabus

Unit I

(Hour 15)

1.1 Introduction - Computer Based System Engineering - Emergent System Properties.
(K3, K4, K5)

1.2 System and Their Environment - System Modeling. (K3, K4, K5)

1.3 System Engineering Process - System Procurement. (K3, K4, K5)

1.4 Software Process - Software Process Models. (K3, K4, K5)

1.5 Process Iteration Software Design and Implementation-Software Validation.
(K3, K4, K5)

1.6 Software Evolution- Automated Process Support. (K3, K4, K5)

Unit II

(Hour 15)

2.1 Project Management: Management Activities. (K2)

2.2 Project Planning. (K2)

2.3 Project Scheduling - Risk Management. (K2)

2.4 Software Requirement: Functional and Non_Functional. (K2)

2.5 Functional Requirements. (K2)

2.6 User Requirements - System Requirements Software Requirements Documents. (K2)

Unit III

(Hour 15)

- 3.1 Requirement Engineering Processes - Feasibility Study. (K4)
- 3.2 Requirement Elicitation and Analysis. (K4)
- 3.3 Requirement Validation - Requirements Management. (K4)
- 3.4 System Model: Context Models. (K4)
- 3.5 Behavioural Models. (K4)
- 3.6 Data Models – Object Models. (K4)

Unit IV

(Hour15)

- 4.1 Architectural Design: Architectural Design Decisions. (K2)
- 4.2 System Organization - Modular Decomposition Styles.
- 4.3 Control Styles - User Interface Design. (K2)
- 4.4 Design Issues-User Interface Design Process. (K2)
- 4.5 User Analysis. (K2)
- 4.6 User Interface Prototyping. (K2)

Unit V

(Hour15)

- 5.1 Software Testing: System Testing - Component Testing. (K2, K6)
- 5.2 Test Case Design - Test Automation. (K2, K6)
- 5.3 Software Cost Estimation: Productivity. (K2, K6)
- 5.4 Estimation Techniques. (K2, K6)
- 5.5 Algorithmic Cost Modelling. (K2, K6)
- 5.6 Project Duration and Staffing. (K2, K6)

Text Book

1. Ian Sommerville, “Software Engineering”, 10th Edition, Pearson Education, 2011.

Reference Books

1. Roger S. Pressman, “Software Engineering: A Practitioner’s Approach”, 7th Edition, McGraw Hill, New York, 2016.
2. Pankaj Jalote, “An Integrated Approach to Software Engineering”, 3rd Edition, Narosa Publication, 2018.

Open Educational Resources (OER)

1. [http://ff.tusofia.bg/~bogi/knigi/SE/Software%20Engineering%209th%20ed%20\(intro%20txt\)%20-%20I.%20Sommerville%20\(Pearson,%202011\)%20BBS.pdf](http://ff.tusofia.bg/~bogi/knigi/SE/Software%20Engineering%209th%20ed%20(intro%20txt)%20-%20I.%20Sommerville%20(Pearson,%202011)%20BBS.pdf).
2. <http://inspirit.net.in/books/academic/Ian%20Sommerville%20Software%20Engineering,%209th%20Edition%20%20%202011.pdf>.
3. <http://www.amazon.in/Integrated-Approach-Software-Engineering/dp/8173197024>.

SEMESTER VI

UECSC20 –ELECTIVE II A: SOFTWARE TESTING

Year: III	Course Code: UECSC20	Title of the Course: Elective – II A: Software Testing	Course Type: Theory	Course Category: Core	H/W 5	Credits 5	Marks 40+60
---------------------	--------------------------------	--	-------------------------------	---------------------------------	-----------------	---------------------	-----------------------

Course Learning Objectives (CLO)

1. Define the characteristics of testing and software development life cycle models.
2. Recognize the facts of software development models to adopt with product characteristics.
3. Understand different types of testing, their objectives and challenges.
4. Interpret the software products to execute and report test cases.
5. Apply software testing methods and to perform various types of testing in a software project.

Course Outcomes (COs)

The Learners will be able to

1. Test various processes and continue quality improvement.
2. Verify types of errors and fault models.
3. Analyze methods of test generation from requirements.
4. Input space modeling using combinatorial designs.
5. Test adequacy assessment using control flow, data flow and program mutations.

CO	PSO					
	PSO1	PSO2	PSO3	PSO4	PSO5	PSO6
CO1	L	H	M	L	M	L
CO2	M	L	L	L	M	L
CO3	M	L	M	L	L	H
CO4	L	M	M	M	L	M
CO5	L	M	L	H	L	L

(Low -L, Medium -M, High-H)

CO	PO					
	1	2	3	4	5	6
CO1	H	H	H	H	H	L
CO2	H	H	H	H	H	L
CO3	H	H	H	H	M	M

CO4	H	H	H	H	H	L
CO5	H	H	M	H	M	H

(Low -L, Medium -M, High-H)

Course Syllabus

Unit I

(Hours 15)

- 1.1 A Perspective on Testing - Basic Definitions. (K1)
- 1.2 Test Cases- Insights from a Venn Diagram. (K1, K2)
- 1.3 Identifying Test Cases – Specification - Based Testing - Code-Based Testing. (K2)
- 1.4 Fault Taxonomies- Levels of Testing – Examples - Generalized Pseudocode. (K1, K2, K3)
- 1.5 The Triangle Problem - Problem Statement- Discussion. (K2, K4)
- 1.6 Traditional Implementation - Structured Implementations. (K2)

Unit II

(Hours 15)

- 2.1 Role of Testing - Verification and Validation. (K1, K2)
- 2.2 Failure, Error, Fault, and Defect -Notion of Software Reliability. (K2, K3)
- 2.3 Objectives of Testing - What Is a Test Case? (K2)
- 2.4 Expected Outcome - Concept of Complete Testing. (K1,K2)
- 2.5 Central Issue in Testing - Testing Activities. (K2, K4)
- 2.6 Test Levels - Sources of Information for Test Case Selection. (K1, K2, K3)

Unit III

(Hours 15)

- 3.1 White-Box and Black-Box Testing - Test Planning and Design. (K1, K2)
- 3.2 Monitoring and Measuring Test Execution - Test Tools and Automation- Test Team Organization and Management. (K2, K3)
- 3.3 Unit Testing- Concept of Unit Testing - Static Unit Testing- Defect Prevention - Dynamic Unit Testing. (K1, K2)
- 3.4 Debugging- Tools for Unit Testing - Control Flow Testing. (K4, K5)
- 3.5 Basic Idea- Outline of Control Flow Testing - Control Flow Graph- Paths in a Control Flow Graph. (K2, K3)
- 3.6 Life Cycle Based Testing - Traditional Waterfall Testing- Waterfall Testing - Pros and Cons of the Waterfall Model. (K2, K6)

Unit IV

(Hours 15)

- 4.1 Integration Testing – Decomposition-Based Integration - Top–Down Integration - Bottom–Up Integration - Sandwich Integration - Pros and Cons. (K2,K4)
- 4.2 Data flow testing -System integration testing - Functional testing - Acceptance testing - Domain testing - Exploratory testing. (K4)
- 4.3 System Test Planning and Automation - Structure of a System Test Plan - Introduction and Feature Description. (K4)
- 4.4 Assumptions - Test Approach - Test Suite Structure - Test Environment - System Test Execution- Basic Ideas - Modeling Defects. (K4,K6)
- 4.5 Preparedness to Start System Testing - Metrics for Tracking System Test - Metrics for Monitoring Test Execution. (K6)
- 4.6 Test Execution Metric Examples - Metrics for Monitoring Defect Reports - Defect Report Metric Examples. (K4,K5)

Unit V

(Hours 15)

- 5.1 Software Reliability - What Is Reliability?(K1,K2)
- 5.2 Definitions of Software Reliability - Factors Influencing Software Reliability. (K2)
- 5.3 Applications of Software Reliability - Test Team Organization - Test Groups. (K1,K2)
- 5.4 Software Quality Assurance Group - System Test Team Hierarchy- Effective Staffing of Test Engineers. (K1,K2)
- 5.5 Recruiting Test Engineers - Retaining Test Engineers - Team Building. (K2,K4)
- 5.6 Software Testing Excellence – Craftsmanship - Best Practices of Software Testing. (K2,K5,K6)

Text Book

1. Kshirasagar Naik, Priyadarshi Tripathy, "Quality Assurance Theory and Practice" John Wiley & Sons, Inc., Publication, 2008.
2. Paul C.Jorgensen, "Software Testing A Craftsman's Approach" CRC Press, 2014.

Reference Books

1. Srinivasan Desikan, Gopaldaswamy Ramesh, "Software Testing Principles and Practice", Dorling Kindersley (India), 2008, ISBN 9788177581218, 817758121X
2. Nageshwar Rao Pusuluri, "Software Testing Concepts And Tools", Dreamtech Publishers, 2008.

Open Educational Resources (OER)

1. <https://www.javatpoint.com/software-testing-tutorial>
2. https://en.wikipedia.org/wiki/Software_testing
3. <https://www.youtube.com/watch?v=sO8eGL6SFsA>