Uka Tarsadia University

B.Tech. Computer

Semester VI

SOFTWARE ENGINEERING

CE5007

EFFECTIVE FROM July-2021

Syllabus version:1.00

Subject Code	Subject Title	Teaching Scheme			
		Hours		Credits	
		Theory	Practical	Theory	Practical
CE5007	Software Engineering	4	2	4	1

Subject Code	Subject Title	Theory Examination Marks		Practical Examination Marks	Total Marks
		Internal	External	CIE	
CE5007	Software Engineering	40	60	50	150

Objectives of the course:

- To apply their foundations in software engineering to adapt to readily changing environments using the appropriate theory, principles and processes.
- To make strong contributions to teams that are responsible for the specifications, design, construction, testing, deployment, maintenance, or use of software systems.
- To demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment.

Course outcomes:

Upon completion of the course, the student shall be able to

CO1: Plan and deliver an effective software engineering process, based on knowledge of widely used development lifecycle models.

CO2: Prepare SRS (Software Requirement Specification) document and translate a requirements specification into an implementable design, following a structured and organized process.

CO3: Make effective use of UML, along with design strategies such as defining a software architecture.

CO4: Recognize how to ensure the Scheduling, Risk, quality of software product, different quality standards and software review techniques.

CO5: Formulate a testing strategy for a software system, employing techniques such as unit testing, test driven development and functional testing.

CO6: Expose Software Process Improvement and Reengineering.

Sr. No.	Topics	Hours
	Unit – I	
1	**Introduction to Software Engineering:** Introduction of software engineering, Changing nature of the software, Legacy software, Software development myths, Generic view of Process, Software engineering - A layered technology, Process framework, Capability Maturity Model Integration, (CMMI), Process technology, Product and process, Perspective process models - Waterfall model, Incremental; Evolutionary process models - Prototype, Spiral, Concurrent development model. **Agile Development:** Agility and Agile Process model, Extreme programming, Other process models of Agile Development and Tools.	8
	Unit – II	

2	**Requirements Engineering and Structured System Design:** Requirement engineering tasks, Initiating the requirements engineering processes, Eliciting requirements, Developing use cases, Negotiating and validating requirements, Design concepts, The design model, Software architecture, Data design, Architectural styles and Patterns, Architectural design, Assessing alternative architectural designs; Modeling component level design - Component, Designing class-based component.	12
	Unit – III	
3	**Data Oriented Analysis and Design:** Building the analysis model: Data modeling concepts, Scenario – based modeling, Flow-Oriented modeling, Class based modeling and creating a behavioral model. User interface design - The Golden rules, User interface analysis and design, Interface analysis, Interface design steps, Design evaluation.	10
	Unit – IV	
4	**Software Project Planning and Quality Assurance:** Estimation - Software scope and feasibility, Software project estimation, Empirical estimation models; Project Scheduling: Project scheduling, Defining a task set for the software project, Defining a task network, Scheduling, Earned value analysis; Risk management: Reactive vs. Proactive risk strategies, Software risk, Risk identification, Risk refinement, Risk mitigation, Monitoring and management, The RMMM plan; Quality management - Quality concepts, Software quality assurance, Software reviews, Statistical software quality assurance, Software reliability, Quality standards-ISO 9000.	8
	Unit – V	
5	**Coding and Testing:** Programming principles and guidelines - Programming practices, Coding standards, Coding process; Verification - Code inspection, Unit testing; Metrics, Software testing fundamentals, Black-box and white box testing, Basis path testing, Control structure testing, Black-box testing - Graph-based testing method, Boundary value analysis; Testing strategies - A strategic approach to software testing, Test strategies for conventional and object-oriented software, Testing strategies for Web and Mobile applications.	12
	Unit – VI	
6	**Advanced Topics in Software Engineering:** Component-based software engineering - Engineering of component-based system, The CBSE process, Domain engineering, Component-based development; Reengineering - Business process reengineering, Reverse engineering.	10
	DevOps: Overview, Problem case definition, Benefits of fixing application development challenges, DevOps adoption approach through assessment, Solution dimensions, What is DevOps?, DevOps importance and benefits, DevOps principles and practices, 7 C’s of DevOps lifecycle for business agility, DevOps and continuous testing.	

Sr. No.	Software Engineering (Practical)	Hours
1	To select the project title and assign requirement engineering to the project title.	2
2	To perform the system analysis: Requirement analysis, SRS.	2
3	To perform the function-oriented diagram: DFD and Structured chart.	2
4	To perform the user's view analysis: Use case diagram.	2
5	To draw the structural view diagram: Class diagram.	2
6	To draw the behavioral view diagram: Sequence diagram, Activity diagram.	2
7	To draw the environmental view diagram: Deployment diagram.	2
8	Implementation of the project.	10
9	To study various testing tools.	2
10	To design test cases and to apply them using various testing tools.	2
11	To study cost estimation and preparation of timeline chart.	2

Text book:

1. Roger S. Pressman - "Software Engineering – A Practitioner's Approach", McGraw-Hill.

Reference books:

1. Deepak Gaikwad, Viral Thakkar – "DevOps Tools from Practitioner's Viewpoint", Wiley India.
2. Sommerville - "Software Engineering", Pearson Education.
3. Pankaj Jalote – "An Integrated approach to SE", Narosa.
4. Rajib Mall - "Software Engineering", PHI.
5. Ghezzi, Jazayeri, Mandrioli - "Fundamentals of Software Engineering", Pearson Education.
6. Stephen R.Schach - "Software Engineering with JAVA", TMH.
7. John M. Nicolas - "Project Management for Business, Engineering and Technology", Elsevier.

Course objectives and Course outcomes mapping:

- Apply their foundations in software engineering to adapt to readily changing environments using the appropriate theory, principles and processes: CO1, CO2
- Make strong contributions to teams that are responsible for the specifications, design, construction, testing, deployment, maintenance, or use of software systems:CO3, CO4, CO5
- Demonstrate the ability to work effectively as a team member and/or leader in an ever-changing professional environment: CO6

Course units and Course outcomes mapping:

Unit No.	Unit Name	Course Outcomes					
		CO1	CO2	CO3	CO4	CO5	CO6
1	Introduction to Software Engineering and Agile Development	✓					
2	Requirements Engineering and Structured System Design		✓				
3	Data Oriented Analysis and Design			✓			
4	Software Project Planning and Quality Assurance				✓		
5	Coding and Testing					✓	
6	Advanced Topics in Software Engineering and DevOps						✓

Programme outcomes:

PO 1: Engineering knowledge: An ability to apply knowledge of mathematics, science, and engineering.

PO 2: Problem analysis: An ability to identify, formulates, and solves engineering problems.

PO 3: Design/development of solutions: An ability to design a system, component, or process to meet desired needs within realistic constraints.

PO 4: Conduct investigations of complex problems: An ability to use the techniques, skills, and modern engineering tools necessary for solving engineering problems.

PO 5: Modern tool usage: The broad education and understanding of new engineering techniques necessary to solve engineering problems.

PO 6: The engineer and society: Achieve professional success with an understanding and appreciation of ethical behaviour, social responsibility, and diversity, both as individuals and in team environments.

PO 7: Environment and sustainability: Articulate a comprehensive world view that integrates diverse approaches to sustainability.

PO 8: Ethics: Identify and demonstrate knowledge of ethical values in non-classroom activities, such as service learning, internships, and field work.

PO 9: Individual and team work: An ability to function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.

PO 10: Communication: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give/receive clear instructions.

PO 11: Project management and finance: An ability to demonstrate knowledge and understanding of the engineering and management principles and apply these to one’s own work, as a member and leader in a team, to manage projects and in multidisciplinary environments.

PO 12: Life-long learning: A recognition of the need for, and an ability to engage in life-long learning.

Programme outcomes and Course outcomes mapping:

Programme Outcomes	Course Outcomes					
	CO1	CO2	CO3	CO4	CO5	CO6
PO1					✓	
PO2						
PO3						
PO4						
PO5	✓			✓	✓	✓
PO6		✓	✓	✓		✓
PO7						
PO8	✓		✓		✓	
PO9		✓	✓	✓	✓	✓
PO10	✓	✓				
PO11	✓	✓		✓		✓
PO12						✓