Course Code & Name	26CAPC7004 Advanced Software Engineering		
Teaching Hours /week (L: T:P: R)	3:0:0:1	Credits	4
CIE Marks	40	ESE Marks	60
Course Type	Theory (PCC)	Exam Hours	3
Prerequisites (if any)	-		

Course Objectives:

- To have a basic theory of software engineering, and to the applications of these basic theoretical principles to a software development project.
- To Provide fundamental knowledge in the Software Development Process including Software Development, Project Management concepts and technology trends.

Course Outcomes (COs)

At the end of the course students should be able to:

CO#	Course Outcomes	Bloom's Knowledge Level (KL)
CO 1	Describe the software life cycle, software planning, analysis, design, and various software engineering models.	K2
CO 2	Apply the git version control system.	K3
CO 3	Explain testing and maintenance strategies on the developed software product to enhance quality	K2
CO 4	Explain Agile principles, methodologies, development practices, testing strategies, and refactoring techniques used in Agile software projects.	K2

Note: *K1- Remember, K2- Understand, K3- Apply, K4- Analyse, K5- Evaluate, K6- Create*

CO-PO Mapping Table (Mapping of Course Outcomes to Program Outcomes)

CO #	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8
CO 1	3	2	2	1	-	-	-	1
CO 2	2	1	1	-	3	-	2	1
CO 3	3	3	2	2	2	1	1	1
CO 4	2	2	2	1	2	1	2	1

Note: *1: Slight (Low), 2: Moderate (Medium), 3: Substantial (High), - : No Correlation*

Course Assessment Method

(CIE: 40 marks, ESE: 60 marks)

Continuous Internal Evaluation Marks (CIE)

Attendance	Assignment/ Micro-project	Internal Examination-1 (Written)	Internal Examination- 2 (Written)	Total
8	12	10	10	40 Marks

End Semester Examination Marks (ESE)

In Part A, all questions need to be answered and in Part B, each student can choose any one full question out of two questions.

PART A	PART B	TOTAL
■ 2 Questions from each module. ■ Total of 8 Questions, each carrying 3 marks **(8x3 =24 marks)**	■ Each question carries 9 marks. ■ Two questions will be given from each module, out of which 1 question should be answered. ■ Each question can have a maximum of 3 subdivisions. **(4x9 = 36 marks)**	**60 Marks**

SYLLABUS

Module #	Syllabus Description	Hours
1	Introduction to Software Engineering: What is Software Engineering, Characteristics of Software. Life cycle of a software system: software design, development, testing, deployment, Maintenance. Project planning phase: project objectives, scope of the software system, empirical estimation models, COCOMO, staffing and personnel planning. Software Engineering models: Predictive software engineering models, model approaches, prerequisites, predictive and adaptive waterfall, waterfall with feedback (Sashimi), incremental waterfall, V model; Prototyping and	12

	prototyping models. Overview of modern lifecycle models: System SDLC, DevOps, Data Science, MLOps, and IT service lifecycle. Software requirements specification, Eliciting Software requirements, Requirement specifications, Software requirements engineering concepts, Requirements modelling, Requirements documentation. Use cases and User stories.	
2	Version control systems basic concepts; Concept of Distributed version control system and Git; Setting up Git; Core operations in Git version control system using command line interface (CLI): Clone a repository; View history; Modifying files; Branching; Push changes, Clone operation, add, commit, log, diff commands, conflict resolution. Pushing changes to the master.	**12**
3	Coding, Testing and Maintenance: Coding guidelines - Code review, Code walkthrough and Code inspection, Code debugging and its methods. Testing - Unit testing , Integration testing, System testing and its types, Black box testing and White box testing, Regression testing. Overview of DevOps and Code Management - Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD), modern development practices including API-first development and secure software development.	**12**
4	Introduction to agility, Agile Principles, Overview of Scrum, Extreme Programming, Feature Driven development, Lean Software Development, Agile project management. Design and development practices in Agile projects, Test Driven Development, Pair Programming, Simple Design, User Stories, Agile Testing, Agile Tools - Agile design practices- Refactoring, Need and significance of Refactoring, Refactoring Techniques.	**10**

Text Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition & Year**
1	Software Engineering: A practitioner's approach	Roger S. Pressman	McGraw-Hill International edition	8/e, 2014
2	Agile Software Development with Scrum	Ken Schwaber	Pearson	1st Edition

Reference Books				
Sl. No	**Title of the Book**	**Name of the Author/s**	**Name of the Publisher**	**Edition & Year**
1	Pankaj Jalote's Software Engineering: With Open Source and GenAI	Pankaj Jalote	Wiley India	1/e, 2024
2	Software Engineering: A Primer	Waman S Jawadekar	Tata McGraw-Hill	1/e, 2008

Web References/ Video Links (NPTEL, SWAYAM…)	
Sl.No	**Link ID**
1	Git Handbook https://guides.github.com/introduction/git-handbook/ Retrieved 8 July 2020
2	https://en.wikibooks.org/wiki/Introduction_to_Software_Engineering/Quality Retrieved 8 July 2020
3	https://opensource.com/article/19/7/understanding- software-design-patterns Retrieved 8 July 2020
4	https://www.scrumguides.org/docs/scrumguide/v2017/2017-Scrum-Guide-US.pdf Retrieved 8 July 2020
5	https://www.redhat.com/en/topics/devops/what-is-ci-cd Retrieved 8 July 2020

Course Level Assessment Questions:

Course Outcome 1 (CO1):

1. Explain the main characteristics of software that make software engineering essential.
2. Describe the phases of the software life cycle with a neat diagram.
3. Explain the difference between predictive and adaptive software engineering models.
4. Describe the steps involved in the project planning phase.
5. Explain the different types of waterfall models such as Sashimi, incremental, and V-model.

Course Outcome 2(CO2):

1. Demonstrate how to initialize a Git repository and commit changes using command-line operations.
2. Apply the branching and merging features of Git to manage multiple developers working on the same project.
3. Show how to resolve a merge conflict that occurs when two users edit the same file.
4. Illustrate how to push local changes to a remote repository and pull updates using Git commands.
5. Create a small Git workflow using add, commit, diff, and log commands to track changes in a project.

Course Outcome 3(CO3):

1. Explain the need for software testing and its importance in software quality assurance.
2. Describe the different types of testing—unit, integration, and system testing.
3. Explain the differences between black box and white box testing techniques.
4. Describe the various methods used for debugging and maintaining software.
5. Explain the purpose and process of regression testing in maintaining software reliability.

Course Outcome 4(CO4):

1. Explain the concept of DevOps and its benefits to the software development lifecycle.
2. Describe the key stages of a CI/CD pipeline.

3. Explain how Continuous Integration helps detect and fix issues early in development.
4. Describe the importance of automation in DevOps practices.
5. Explain the relationship between Continuous Delivery and Continuous Deployment

Model Question Paper

Reg No.:_____		Name:_____

FEDERAL INSTITUTE OF SCIENCE AND TECHNOLOGY

1MCA S7 (R,S) Examination ---------- 2026 (2026 Scheme)

Course Code: 26CAPC7004

Course Name: Advanced Software Engineering

Max. Marks: 60		Duration: 3 Hours

PART A

	Answer all questions, each carries 3 marks.	***CO***	Marks
1	Define Software Engineering. List any three key characteristics of software	CO1	(3)
2	What is the relevance of Software Requirement Specification (SRS) in software engineering?	CO1	(3)
3	What is a Version Control System? How does it help in software development?	CO2	(3)
4	List any three commonly used Git commands and explain their purpose.	CO2	(3)
5	What is the difference between code walkthrough, and code inspection?	CO3	(3)
6	Distinguish between black box testing and white box testing.	CO3	(3)
7	List any three key principles of Agile methodology.	CO4	(3)
8	What is refactoring? State its need and significance in software development.	CO4	(3)

PART B

(Answer any one full question from each module, each question carries 9 marks)

Module I

9	Explain the COCOMO model in detail and describe its different types.	CO1	(9)

OR

10	a) Explain the steps in waterfall model b) Differentiate between sashimi model and incremental waterfall model with suitable diagram.	CO1 CO1	(3) (6)
	Module II		
11	a) Explain the working of Git as a Distributed Version Control System. b) Illustrate the steps involved in cloning, committing, and pushing changes to a repository.	CO2 CO2	(3) (6)
	OR		
12	a) Explain the use of branching and merging in GIt using suitable examples. b) Differentiate between Fast-forward merge and Three-way merge with suitable diagram.	CO2 CO2	(4) (5)
	Module III		
13	a) Explain the relevance of unit testing in Software development b) Explain the architecture of xUnit test Framework	CO3 CO3	(3) (6)
	OR		
14	a) What is Continuous Integration? b) Explain the prerequisites in Continuous integration	CO3 CO3	(3) (6)
	Module IV		
15	Explain Scrum methodology along with its roles, artifacts and ceremonies with suitable diagram	CO4	(9)
	OR		
16	a) What is agility in agile methodology? b) Explain the principles followed in Agile methodologies.	CO4 CO4	(3) (6)
	*****		

Focus on Excellence