

20INMCA302	INTRODUCTION TO SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT
		GENERAL	3	1	0	4

Preamble: This course introduces students to the basic theory of software engineering, and to the applications of these basic theoretical principles to a software development project.

Course Outcomes: After the completion of the course the student will be able to

CO 1	Understand the different development practices and its advantages
CO 2	Understand the different process models and choose the best model for their project
CO 3	Understand and meet ethical standards and legal responsibilities.
CO 4	Work as an effective member or leader of software engineering teams.
CO 5	Understand the environment and work culture in a software organization

Mapping of course outcomes with program outcomes

	PO 1	PO 2	PO 3	PO 4	PO 5	PO 6	PO 7	PO 8	PO 9	PO 10	PO 11	PO 12
CO 1			2		3		1	3	2	1	3	1
CO 2			2		3		1	3	2	1	3	1
CO 3			2		3		1	3	2	1	3	1
CO 4			2		3		1	3	2	1	3	1
CO 5			2		3		1	3	2	1	3	1

Assessment Pattern

Bloom's Category	Continuous Assessment Tests		End Semester Examination
	1	2	
Remember(K1)	5	5	6
Understand(K2)	10	10	15
Apply(K3)	20	20	18
Analyse(K4)	10	10	15
Evaluate(K5)	5	5	6
Create(K6)			

Mark distribution

Total Marks	CIE	ESE	ESE Duration
100	40	60	3 hours

Continuous Internal Evaluation Pattern:

Attendance	: 8 marks
Continuous Assessment Test(2numbers)	: 20 marks
Assignment/Quiz/Course project	: 12marks

End Semester Examination Pattern: There will be two parts; Part A and Part B. Part A contains 10 compulsory short answer questions, 2 from each module. Each question carries 3 marks. Part B contains 2 questions from each module of which student should answer any one. Each question can have maximum 2 sub-divisions and carry 6 marks

Course Level Assessment Questions**Course Outcome 1 (CO1):**

1. Summarize the various approaches for “Recording” the Requirements.
(K2)
2. The high-level design should sketch out all the application’s security needs. Evaluate the methods by which we can ensure security for our application
(K5)
3. Imagine you are in a project team and the Team Leader asked you to sketch out an UML diagram for a particular scenario so that the diagram should focus on a sequence of messages. Which will be the best UML diagram you can draw in this particular scenario? Justify your answer with an example.
(K5)

Course Outcome 2 (CO2):

1. Compare the waterfall with feedback model with the sashimi model? (K5)
2. The waterfall model and the prototyping model can be accommodated in the Spiral process model. Justify (K5)
3. Demonstrate any one project size estimation technique and its effort calculation using COCOMO with example. (K3)

Course Outcome 3(CO3):

1. Illustrate agile process? List out its characteristics. (K4)
2. Justify the importance of pair programming?(2) (K5)
3. Illustrate Refactoring? Justify its importance. List out different Refactoring techniques.(K4)

Course Outcome 4 (CO4):

1. Describe the characteristics of user stories. (K1)
2. Explain Scrum framework. (K1)
3. Illustrate the various tools for Agile project management. (K4)

Course Outcome 5 (CO5):

1. Demonstrate the importance of CI/CD (K3)
2. DevOps is a set of practices that works to automate and integrate the processes between software development and IT teams. Justify. (K5)
3. State the benefits of release monitoring (K4)

Syllabus**Module 1 (Introduction to Software Engineering)**

What is Software Engineering - Why is software engineering important, Details around requirements gathering, Software design, Development, Testing, Deployment, Maintenance. Planning phase – project planning objective, software scope, empirical estimation, models, COCOMO, staffing and personal planning.

Module 2 (Software Engineering models)

Predictive software engineering models and its application - Model Approaches – Prerequisites - predictive and adaptive waterfall - waterfall with feedback - Sashimi - incremental waterfall - V model–System development lifecycle- IterativevsPredictive–prototypes-Spiral- unified process- Clean room - Rapid Application development principles – risk management.

Module 3 (Fundamentals of Agile Development)

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.

The Agile lifecycle and its impact on testing, Test-Driven Development (TDD), JUnit framework and tools for TDD, Testing user stories - acceptance tests and scenarios, Planning and managing testing cycle, Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools to support the Agile tester.

Module 4 (Scrum Framework)

Introduction to Scrum, Project phases, Agile Estimation, Planning game, Product backlog, Sprint backlog, Iteration planning, User story definition, Characteristics and content of user stories, Acceptance tests and Verifying stories, Project velocity, Burn down chart, Sprint planning and retrospective Daily scrum, Scrum roles - Product Owner, Scrum Master, Scrum Team, Scrum case study

Module 5 (Industry Trends)

Introduction to DevOps- A unified process between development and operations-Continuous Integration (CI), continuous testing, and continuous deployment - Configuration management, release management, and monitoring and learning.

Text Books

1. Rod Stephens, “Beginning Software Engineering”, Wrox Series, Wiley India Pvt Ltd, (2015).

Reference Books

1. Alistair Cockburn, “Agile Software Development: The Cooperative Game”, Addison Wesley, 2nd Edition(2006).
2. Andrew Hunt, David Thomas, “The Pragmatic Programmer: From Journeyman to Master”, Pearson India, 1st Edition(2008).
3. Ken Schwaber, Mike Beedle, “Agile Software Development with Scrum”, Pearson (2008).
4. Lisa Crispin, Janet Gregory, “Agile Testing: A Practical Guide for Testers and Agile Teams”, Addison Wesley Professional, 1st Edition(2008).
5. Mike Cohn, “User Stories Applied: For Agile Software Development”, Addison Wesley, 1st Edition,(2004).
6. Pressman, R.S., “Software Engineering: A Practitioner's Approach”, McGraw Hill SE, 7th Edition, (2010).
7. Robert C. Martin, “Agile Software Development, Principles, Patterns and Practices”, Prentice Hall Imprint, Pearson Education, 2nd Edition(2002).
8. Rod Stephens, “Beginning Software Engineering”, Wrox Series, Wiley India Pvt Ltd (2015).
9. RyPress “Ry's Git Tutorial” (Free-book)
10. Jennifer Davis & Katherine Daniels, Effective DevOps Building a Culture of Collaboration, Affinity, and Tooling at Scale, O'Reilly Media, Inc, First Edition, 2016
11. Gene Kim, Jez Humble, Patrick Debois, John Willis, “The DevOps Handbook How to create world-class agility, reliability and security in technology organizations”, IT Revolution press, First Edition, 2016

Web Resources

Introduction to DevOps

1. <https://www.edx.org/course/introduction-to-devops-transforming-and-improving>
2. <https://courses.edx.org/courses/course-v1:Microsoft+DEV212x+4T2017/course/>
3. <https://guides.github.com/introduction/git-handbook/>

Course Contents and Lecture Schedule

No	Topic	No. of Lectures
1	Module 1	9 hrs.
1.1	What is Software Engineering - Why is software engineering important	1
1.2	Details around requirements gathering	2
1.3	Software design	1
1.4	Development, Testing	1
1.5	Deployment, Maintenance	1
1.6	Planning phase – project planning objective, software scope	1
1.7	Empirical estimation, models, COCOMO	1
1.8	Staffing and personal planning	1
2	Module 2	10 hrs.
2.1	Predictive software engineering models and its application	1
2.2	Model Approaches – Prerequisites - predictive and adaptive waterfall	2
2.3	Waterfall with feedback	1
2.4	Sashimi - incremental waterfall - V model	1
2.5	System development life cycle - Iterative vs Predictive – prototypes	1
2.6	Spiral - unified process	2
2.7	Cleanroom - Rapid Application development principles – risk management.	2
3	Module 3	8 hrs.
3.1	Introduction to agility, Agile Principles	1
3.2	Extreme Programming, FeatureDriven development, Lean Software	2
3.3	Design and development practices in Agile projects, Test Driven Development,	1
3.4	Refactoring, Pair Programming, Simple Design, User Stories, Agile Testing,	
3.5	The Agile lifecycle and its impact on testing, Test-Driven Development (TDD),	
3.6	Exploratory testing, Risk based testing, Regression tests, Test Automation, Tools	1
4	Module 4	10 hrs.
4.1	Introduction to Scrum, Project phases	1
4.2	Agile Estimation, Planning game	1
4.3	Product backlog, Sprint backlog, Iteration planning	1

4.4	User story definition, Characteristics and content of user stories, Acceptance tests	2
4.5	Project velocity, Burn down chart	1
4.6	Sprint planning and retrospective Daily scrum, Scrum roles - Product Owner,	2
4.7	Scrum case study	1
4.8	Tools for Agile project management.	1
5	Module 5	11 hrs.
5.1	Introduction to DevOps, A unified process between development and operations	2
5.2	Continuous Integration (CI), continuous testing	2
5.3	Continuous deployment	1
5.4	Configuration management	2
5.5	release management	2
5.6	monitoring and learning	2

