22CSC51 - AGILE METHODOLOGIES

Programme & Branch	**B.E. - Computer Science and Engineering**	**Sem.**	**Category**	**L**	**T**	**P**	**Credit**
Prerequisites	**Nil**	**5**	**PC**	**3**	**0**	**2**	**4**

Preamble	This course introduces software engineering concepts and agile principles which is to be acquired by software engineers and developers. It also focuses on providing hands-on experience in designing and developing software systems.

Unit – I	**Process Models, Analysis and Design**	**9**

The Software Process –Software process structure – Process models: Waterfall model – Incremental process models – Evolutionary process models – Understanding Requirements–Requirement Engineering–Eliciting Requirements –Requirement Modeling: Scenario Based Modeling–Class-Based Modeling– Design Concepts.

Unit – II	**Agile Principles and Scrum**	**9**

Understanding Agile Values–Agile Principles: 12 Principles of Agile Software–-Scrum and Self-Organizing Teams: Rules of Scrum–Scrum Values–Daily Scrum–Sprints, Planning and Retrospectives-Scrum Planning and Collective Commitment: User stories–Conditions of Satisfaction–Story Points and Velocity–Burn down Charts– Planning and Running a Sprint–Generally Accepted Scrum Practices

Unit – III	**XP and Embracing Change, Lean, and Kanban**	**9**

Primary Practices of XP–The XP values – Understanding the XP principles–Feedback Loops-Lean Thinking–Eliminate Waste–Value Stream Map–Deliver As Fast As Possible–WIP Area Chart–Pull Systems – The Principles of Kanban- – Improving Process with Kanban – Measure and Manage Flow – Emergent Behavior with Kanban

Unit – IV	**Software Testing Fundamentals**	**9**

Software testing strategies: Strategic approach – Issues – Test strategies for conventional and Object-Oriented software – Validation and System testing–Debugging–Testing conventional applications: White box testing–Basis path testing–Control structure testing–Black box testing.

Unit – V	**Software Project Management**	**9**

Software Project Management Concepts–Process and Project Metrics: Software Measurement- Metrics for Software Quality– Estimation for Software Projects: Decomposition Techniques – COCOMO Model–Project Scheduling: Basic Principles – Scheduling-Earned Value Analysis– Software Process Improvements (SPI) – The SPI Process –Capability Maturity Model Integration (CMMI).

LIST OF EXPERIMENTS / EXERCISES:

1.	Create a product back log with stories.
2.	Determine Release plan to decide which stories can be accomplished in the release.
3.	Write Sprint plan to determine the features that can be accomplished in the first iteration, or sprint.
4.	Manage the workload by executing the sprint plan.
5.	Use predefined and user created queries to track project progress.
6.	Prepare Schedule for reviewing sprint.
7.	Create a plan to shut down the first sprint and start the next one
8.	Identify use cases and develop business use case model.
9.	Identify the conceptual classes (boundary, controller and entity classes) and develop a domain model with UML Class diagram.
10.	Develop user interface using Python, create DB using MySQL and Perform unit and integration testing.

Lecture:45, Practical:30, Total:75

TEXT BOOK:

1.	Roger S.Pressman & Bruce R.Maxim, "Software Engineering: A Practitioner‘s Approach", 8th Edition, McGraw-Hill Education, 2019.(Units –1,4,5)
2.	Andrew Stellman and Jennifer Greene, “Learning Agile: Understanding Scrum, XP, Lean, and Kanban”, First Edition, O’Reilly Media Inc,2015. (Units 2,3)

REFERENCES/ MANUAL / SOFTWARE:	
1.	Ian Sommerville, “Software Engineering”,10th Edition, Pearson Education,2014.
2.	Kenneth S. Rubin, “Essential Scrum: A Practical Guide to the Most Popular Agile Process”, Addison-Wesley,2012.
3.	Infosys spring board contents provided by Infosys at https://infyspringboard.onwingspan.com/web/en/page/home

COURSE OUTCOMES: On completion of the course, the students will be able to		**BT Mapped (Highest Level)**
CO1	apply the requirement engineering tasks and design concepts to the various software development models for a given scenario.	Applying (K3)
CO2	apply Scrum Principles for designing and implementing projects	Applying (K3)
CO3	Use XP, Lean and Kanban principles for developing software projects.	Applying (K3)
CO4	make use of various software testing techniques to test the software system and real world scenarios	Applying (K3)
CO5	Carry out different software project management activities for a given software application.	Applying (K3)

Mapping of COs with POs and PSOs

COs/ POs	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO1	3	3	2										3	2
CO2	3	2	1	1	1				2	1	2	1	3	1
CO3	3	2	1						2	1			3	1
CO4	3	2	1	1	1				1	1	2	1	3	1
CO5	3	2	1										3	1

1 – Slight, 2 – Moderate, 3 – Substantial, BT- Bloom’s Taxonomy

ASSESSMENT PATTERN – THEORY

Test / Bloom’s Category*	Remembering (K1) %	Understanding (K2) %	Applying (K3) %	Analyzing (K4) %	Evaluating (K5) %	Creating (K6) %	Total %
CAT1	-	70	30				100
CAT2	-	75	25				100
CAT3	-	75	25				100
ESE	-	75	25				100

* ±3% may be varied (CAT 1, 2, 3 – 50 marks & ESE – 100 marks)