

**Pre Requisite(s):** Nil

### Course Objectives

- To introduce basic software engineering concepts
- To introduce the Agile Software development process
- Hands on training (experiential learning) to digest the concepts learned in the class. This is a reading and discussion subject on issues in the engineering of software systems and software development project design. It includes the present state of software engineering, what has been tried in the past, what worked, what did not, and why

### Course Outcomes

**CO1:** To understand the fundamentals of cyber-physical systems and analyze their design in different applications

**CO2:** To understand the foundations of modeling in CPSs, software based feedback control and apply them in the context of sample CPS systems

**CO3:** To understand the design of distributed CPS systems with respect to synchronization, real-time scheduling and management and security issues

**CO4:** To understand the techniques for formal verification and model integration in CPS and apply them in different domain applications

### CO – PO Mapping

PO/PSO	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2
CO														
CO1	1	-	-	-	-	-	-	-	-	-	-	2	3	2
CO2	2	3	2	-	-	-	-	-	-	-	-	2	3	2
CO3	2	1	3	-	-	2	-	-	-	1	-	2	3	2
CO4	2	2	2	-	-	3	-	-	-	3	-	2	3	2
CO5	-	-	-	-	-	-	-	-	-	-	-	2	3	2

### Syllabus

#### Unit 1

Process Models – overview - Introduction to Agile - Agile Manifesto - principles of agile manifesto - Agile Requirements - User personas - story mapping - user stories - estimating and prioritizing stories – INVEST - acceptance criteria - Definition of Done - Release planning - Key aspects of Scrum - roles - Product Owner - Scrum Master - Team and product backlog Scrum process flow - product backlog - sprints backlog - scrum meetings –demos - How sprint works - Sprint Planning - Daily scrum meeting - updating sprint backlog - Burn down chart - sprint review - sprint retrospective - Scrum Metrics – velocity - burn down - defects carried over.

#### Unit 2

Traditional process Models – Waterfall – incremental - Requirements Engineering - Tasks Initiation – Elicitation- Developing Use Cases-Building the analysis Model – Negotiation – Validation - Requirements Modelling - building the analysis model - Scenario based methods - UML Models.

#### Unit 3

Design engineering Design concepts - Design models - software architecture - architectural styles and patterns - Performing user interface Design-Golden Rules-User Interface Analysis and Design- Interface Analysis -Interface design steps - Testing strategies and tactics - Unit testing - integration testing - validation and system testing.

**Text Book(s)**

Pressman R S, Bruce R. Maxim, "Software engineering - A Practitioner's Approach", Eighth Edition, Tata McGraw-Hill, 2014.

**Reference(s)**

Crowder JA, Friess S, "Agile project management: managing for success", Cham: Springer International Publishing; 2015.

Stellman A, Greene J. Learning agil, "Understanding scrum, XP, lean, and kanban", O'Reilly Media, Inc."; 2014.

Gregory J, Crispin L., "More agile testing: learning journeys for the whole team", Addison-Wesley Professional; 2014.

Rubin KS., "Essential Scrum: a practical guide to the most popular agile process", Addison-Wesley; 2012.

Cohn M. User stories applied: For agile software development. Addison-Wesley Professional; 2004.

**Evaluation Pattern**

Assessment	Internal	External
Periodical 1 (P1)	15	
Periodical 2 (P2)	15	
*Continuous Assessment (CA)	20	
End Semester		50

\*CA – Can be Quizzes, Assignment, Projects, and Reports