22-382–0104	SOFTWARE ENGINEERING	CATEGORY	L	T	P	CREDIT
		CORE	3	1	0	4

Prerequisite: NIL

Course Outcomes:
After completion of this course, students will be able to

CO1	Choose suitable life cycle models to be used in a particular context.	(Cognitive level : Apply)
CO2	Develop a Software Requirement Specification.	(Cognitive level : Apply)
CO3	Develop a UML Diagram from data.	(Cognitive level : Apply)
CO4	Compare different testing strategies.	(Cognitive level : Understand)
CO5	Summarize Software Quality Assurance	(Cognitive level : Understand)

Mapping of Course Outcomes with Programme Outcomes - Low=1, Medium=2, High=3

	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12
CO1	2	1		1			2	1				
CO2	2	2	2	1	2		2	1				
CO3	2	2	2	1	2		2	1				
CO4	2	1	2				2	1				
CO5	2	1	2				2					

22-382-0104 SOFTWARE ENGINEERING

UNIT I (12 Hours)

Introduction to Software Engineering - Professional software development, Software engineering ethics. Software process models - Software process models- Waterfall Model, V-process model, Spiral Model, Prototyping Model, Software Iterative and Incremental Method. Agile software development - Agile methods, agile manifesto - values and principles. Agile development techniques- Scrum, Lean(LN),Extreme Programming (XP), Agile Unified Process (AUP). Agile Project Management. Overview of DevOps and Code Management – Code management, DevOps automation, Continuous Integration, Delivery, and Deployment (CI/CD/CD).

UNIT II(8 Hours)

Functional and non-functional requirements, Requirements engineering processes. Requirements elicitation, Requirements validation, Requirements change, Traceability Matrix. Developing use cases, Software Requirements Specification Template, Software Design- Overview of Software Design, How to characterize a good software design, Cohesion and coupling, Layered arrangement of modules, Approaches to Software Design.

UNIT III(8 Hours)

Software Maintenance – Characteristics of Software Maintenance, Software Reverse Engineering, Software maintenance process models, Estimation of maintenance cost. Object Modeling using UML – Basic object Orientation concepts, Unified Modeling Language, UML diagrams, Use Case Model, Class Diagrams, Interaction Diagrams, Activity Diagram, State Chart diagram.

UNIT IV(8 Hours)

Coding and Testing – Coding, code review, Software Documentation, Testing, Software testing strategies - Unit Testing, Integration Testing, Validation testing, System testing, Debugging, White box testing, Path testing, Control Structure testing, Black box testing, Testing Documentation.

[

UNIT V(9 Hours)

Software Quality, Software Quality Dilemma, Achieving Software Quality Elements of Software Quality Assurance, SQA Tasks, Software measurement and metrics. Software Process Improvement (SPI), SPI Process CMMI process improvement framework, ISO 9001:2000 for Software. Cloud-based Software - Virtualisation and containers, Everything as a service (IaaS, PaaS), Software as a service. Microservices Architecture - Microservices, Microservice deployment.

Textbook:

- “Fundamentals of Software Engineering”,byRajib Mall, Edition 5, February 2019,PHI Learning Pvt. Ltd.,
- “Software Engineering–a Practitioner’s approach” by Roger S Pressman, Edition 7,2017,McGraw Hill.
- “Software Engineering” by Ian Sommerville, Edition 10,October 2018,PEARSON INDIA.
- “Software Engineering A Precise Approach” by Pankaj Jalote, 2010, WILEY INDIA
- “Software Testing- Principles, Techniques and Tools”, M G Limaye
- Software Quality Assurance from theory to implementation, Daniel Galin.