Program	ADP Data Science	
Course Code	CC-303	
Course Title	Software Engineering	
Credit Hours	**Theory**	**Lab**
	3	0
Lecture Duration	90 minutes (1.5 Hours), 2 lectures per week	
Semester	4	
Pre-requisites	**Courses**	**Knowledge**
	Nil	Nil
Follow Up Courses		
Aims and Objectives	Upon completion of this course, the student will be able to: 1. Understanding Grasp modeling concepts with emphasis performance analysis. on 2. Planning of software 3. Designing Software	

Course Learning Outcomes (CLOs):

CLO No	Course Learning Outcome	Bloom Taxonomy
CLO-1	Describe various software engineering processes and activates techniques for the design of digital electronic circuits	C1 (Describe)
CLO-2	Apply the system modeling techniques to model a medium size software systems	C3 (Apply)
CLO-3	Apply software quality assurance and testing principles to medium size software systems	C4 (Apply)
CLO-4	Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis	C2 (Discuss)

Learning Outcomes	At the end of the course, you should be able to: • Describe various software engineering processes and activates
	• Apply the system modeling techniques to model a medium size software systems • Apply software quality assurance and testing principles to medium size software systems • Discuss key principles and common methods for software project management such as scheduling, size estimation, cost estimation and risk analysis

Contents	1. Describe various software engineering processes and activates 1.1. Nature of Software 1.2. Overview of Software Engineering 1.3. Professional software development 2. Software engineering practice 2.1. Software process structure 2.2. Software process models 2.3. Agile software Development 2.4. Agile process models 2.5. Agile development techniques 3. Requirements engineering process 3.1. Functional requirements 3.2. Non-functional requirements 4. Model driven engineering 4.1. Context models 4.2. Interaction models 4.3. Structural models 4.4. Behavioral models 5. Architectural design 5.1. Design and implementation 6. UML diagrams 6.1. Design patterns 6.2. Software testing and quality assurance
	6.3. Software evolution 6.4. Project management 6.5. Project planning 7. Configuration management 7.1. Software Process improvement
Teaching-learning Strategies	• Interactive class session • Hands on practices in class • Brainstorming and Group discussion sessions
Assignments	• Paper based written assignments 3 • Project 2 • Quiz 4
Textbooks &	• Software Engineering, Sommerville I., 10th Edition, Pearson Inc., 2014
Reference material	• Software Engineering, A Practitioner's Approach, Pressman R. S.& Maxim B. R., 8th Edition, McGraw-Hill, 2015.
Notes	• The instructor reserves the right to modify the grading scheme/marks division and course outline during the semester.