

Course Code	Course Title	Credits	Lectures /Week
USCS403	Software Engineering	2	3
About the Course: This course covers a collection of methods which embody an "engineering" approach to the development of software. It discusses the nature of software and software projects, software development models, software process maturity, project planning, management, and estimations. It also underlines the topics on software testing and quality assurance.			
Course Objectives: <ul style="list-style-type: none"> To learn and understand the Concepts of Software Engineering To learn and understand Software Development Life Cycle To apply the project management and analysis principles to software project development. To apply the design & testing principles to software project development. 			
Learning Outcomes: After successful completion of this course, students would be able to <ul style="list-style-type: none"> Plan a software engineering process life cycle, including the specification, design, implementation, and testing of software systems that meet specification, performance, maintenance and quality requirements Analyze and translate a specification into a design, and then realize that design practically, using an appropriate software engineering methodology. Know how to develop the code from the design and effectively apply relevant standards and perform testing, and quality management and practice Able to use modern engineering tools necessary for software project management, time management and software reuse. 			
Unit	Topics	No of Lectures	
I	Introduction: The Nature of Software, Software Engineering, Professional Software Development, Layered Technology, Process framework, CMM, Process Patterns and Assessment Prescriptive Models: Waterfall Model, Incremental, RAD Models Evolutionary Process Models: Prototyping, Spiral and Concurrent Development Model Specialized Models: Component based, Aspect Oriented development, The Unified Process Phases, Agile Development- Agility, Agile Process, Extreme Programming Requirement Analysis and System Modeling: Requirements Engineering, Eliciting Requirements, SRS Validation, Components of SRS, Characteristics of SRS, Object-oriented design using the UML - Class diagram, Object diagram, Use case diagram, Sequence diagram, Collaboration diagram, State chart diagram, Activity diagram, Component diagram, Deployment diagram	15	

II	<p>System Design: System/Software Design, Architectural Design, Low-Level Design Coupling and Cohesion, Functional-Oriented Versus Object-Oriented Approach, Design Specifications, Verification for Design, Monitoring and Control for Design</p> <p>Software Measurement and Metrics: Process Metrics and Project Metrics, Software Measurement, Object Oriented Metrics, Software Project Estimation, Decomposition Techniques, LOC based, FP based and Use case based estimations, Empirical estimation Models</p> <p>Software Project Management: Estimation in Project Planning Process</p> <p>–Software Scope and Feasibility, Resource Estimation, Empirical Estimation Models – COCOMO II, Estimation for Agile Development, The Make/Buy Decision</p> <p>Project Scheduling - Basic Principles, Relationship Between People and Effort, Effort Distribution, Time-Line Charts</p>	15
III	<p>Risk Management - Risk strategies, Software risks, Risk Identification, projection, RMMM Quality Concepts</p> <p>Software Quality Assurance SQA activities, Software reviews, FTR, Software reliability and measures, SQA plan Software Configuration Management, elements of SCM, SCM Process, Change Control Capability Maturity Model</p> <p>Software Testing : Verification and Validation, Introduction to Testing, Testing Principles, Testing Objectives, Test Oracles, Levels of Testing, White-Box Testing/Structural Testing, Functional/Black-Box Testing, Test Plan, Test-Case Design</p>	15
<p>Textbooks:</p> <ol style="list-style-type: none"> 1. Software Engineering, A Practitioner’s Approach, Roger S, Pressman, 2019 2. Software Engineering: principles and Practices, Deepak Jain, OXFORD University Press, 2008 <p>Additional References:</p> <ol style="list-style-type: none"> 1. Software Engineering, Ian Sommerville, Pearson Education, 2017 2. Fundamentals of Software Engineering, Fourth Edition, Rajib Mall, PHI, 2018 3. Software Engineering: Principles and Practices, Hans Van Vliet, John Wiley & Sons, 2010 4. A Concise Introduction to Software Engineering, Pankaj Jalote, Springer 		