

COURSE TITLE: SOFTWARE ENGINEERING**COURSE OBJECTIVES:**

- The basic objective of software engineering is to develop methods and procedures for software development that can scale up for large systems.
- It can be used consistently to produce high-quality software at low cost and with a small cycle of time.

LEARNING OUTCOMES:

Students will be able to:

- Apply use of knowledge of Software Life Cycle to successfully implement the projects in the corporate world
- Identify the Inputs, Tools and techniques to get the required Project deliverable and Product deliverable using knowledge areas of Project Management.

Code	Course	Teaching Period / Week		Credit			Duration of Theory Exam (in Hrs.)
		L	Pr./ Tu	Int.	Ext.	Total	
MCS103	Software Engineering	4	-	2	2	4	2

Module No	Objective	Content	Evaluation
1	The objective of this module is to introduce the student to the basic foundations of software development using software engineering principles.	Introduction to software engineering and project management Introduction to Software Engineering, Software Components, Software Characteristics, Software Crisis, Software Engineering Processes, Similarity and Differences from Conventional, Engineering Processes, Software Quality Attributes. Software Development Life Cycle (SDLC), Models: Water Fall Model, Prototype Model, Spiral Model, Evolutionary Development Models, Iterative Enhancement Models.	Unit Test-1 (Marks-25)
2	To introduce students to Software Requirement elicitation techniques	Software Requirement Analysis and Specification Requirement Engineering Process: Elicitation, Analysis, Documentation, Review and Management of User Needs, Feasibility Study, Information Modeling, Data Flow Diagrams, Entity Relationship Diagrams, Data Dictionary Decision Tables, SRS Document, IEEE Standards for SRS. Requirement Elicitation: Interviews, Questionnaire, Brainstorming, Facilitated Application Specification Technique (FAST), Use Case Approach. SRS Case study.	Online Test (Marks-15)

3	This will introduce the students to the basic concepts of software project scheduling & design	<p>Software Project Planning and Scheduling Business Case, Project selection and Approval, Project charter, Project Scope management: Scope definition and Project Scope management, Creating the Work Breakdown Structures, Scope Verification, Scope Control. Staffing Level Estimation, Effect of schedule Change on Cost, Degree of Rigor & Task set selector, Project Schedule, Schedule Control</p> <p>Software Design Basic Concept of Software Design, Architectural Design, Low Level Design: Modularization, Design Structure Charts, Pseudo Codes, Flow Charts, Coupling and Cohesion Measures, Design Strategies: Function Oriented Design, Object Oriented Design, Top-Down and Bottom-Up Design. Software Measurement and Metrics: Various Size Oriented Measures: Halstead's Software Science, Function Point (FP) Based Measures, Cyclomatic Complexity Measures: Control Flow Graphs.</p>	
4	To understand the importance of Software Testing strategies and Quality Assurance during the software development process.	<p>Software Testing and Quality Assurance Testing Objectives, Unit Testing, Integration Testing, Acceptance Testing, Regression Testing, Testing for Functionality and Testing for Performance, Top-Down and Bottom-Up Testing Strategies: Test Drivers and Test Stubs, Structural Testing (White Box Testing), Functional Testing (Black Box Testing), Test Data Suit Preparation, Alpha and Beta Testing of Products.</p> <p>Static Testing Strategies: Formal Technical Reviews (Peer Reviews), Walk Through, Code Inspection, Compliance with Design and Coding Standards Software Quality Assurance (SQA): Verification and Validation, SQA Plans, Software Quality Frameworks, ISO 9000 Models, SEI-CMM Model.</p>	Assignment (Marks-5)
5	The objectives of this module is to introduce the fundamentals of software costing and maintenance To describe three metrics for software productivity assessment.	<p>Software Maintenance and Software Project Management Software as an Evolutionary Entity, Need for Maintenance, Categories of Maintenance: Preventive, Corrective and Perfective Maintenance, Cost of Maintenance, Software Re- Engineering, Reverse Engineering. Software Configuration Management Activities, Change Control Process, Software Version Control, An Overview of CASE Tools. Software Estimation: Size Estimation: Function Point (Numericals). Cost Estimation: COCOMO (Numericals), COCOMO-II (Numericals). Software Risk Analysis and Management.</p>	Assignment (Marks-5)

EVALUATION:

Evaluation	Details	Marks
(* please give details of assessment in terms of Unit test/ Project/ quiz /or other assignments and marks allotted for it)		
Internal	<ul style="list-style-type: none"> • Unit test • Online Test • Assignment 	50 Marks
External	Final Examination	50 Marks
Total marks		100 Marks

TEXT BOOKS:

- 1) Roger S Pressman, *Software Engineering*, 5th and 7th edition, (2018) McGraw Hill publication.
- 2) Kathy Schwalbe, *Managing Information Technology Project*, 6edition, (2016) Cengage Learning publication.

REFERENCE BOOKS:

- 1) Jack T Marchewka, *Information Technology Project Management*, (2010) Wiley India publication.
 - 2) KK Agrawal, Yogesh Singh, *Software Engineering* 3rd edition (2015) by New Age International publication.
 - 3) Richard H. Thayer, *Software Engineering Project Management*, (2016) Wiley India Publication.
 - 4) Douglas Bell, *Software Engineering for students: A Programming Approach*, (2018) Pearson publication.
-