

Course Code:	Course Title	Credit
ITC504	Software Engineering	3
<b>1)Prerequisite:</b> Basic programming of knowledge.		
<b>2)Course Objectives:</b>  The course aims:		
1	To provide the knowledge of software engineering discipline	
2	To understand Requirements and analyze it	
3	To do planning and apply scheduling	
4	To apply analysis, and develop software solutions	
5	To demonstrate and evaluate real time projects with respect to software engineering principles	
6	Apply testing and assure quality in software solution	
<b>3)Course Outcomes:</b>  On successful completion, of course, learner/student will be able to:		
1	Understand and use basic knowledge in software engineering	
2	Identify requirements, analyze and prepare models	
3	Plan, schedule and track the progress of the projects.	
4	Design & develop the software solutions for the growth of society	
5	To demonstrate and evaluate real time projects with respect to software engineering principles	
6	Apply testing and assure quality in software solution	

#### 4) syllabus

Module		Content	Hrs
<b>Module 1</b>	<b>Introduction to Software Engineering</b>	Nature of Software, Software Engineering, Software Process, Capability Maturity Model (CMM) Generic Process Model, Prescriptive Process Models: The Waterfall Model, V-model, Incremental Process Models, Evolutionary Process Models, Concurrent Models, Agile process, Agility Principles, Extreme Programming (XP), Scrum, Kanban model Self-learning Topics: Personal and Team Process Models	07
<b>Module 2</b>	<b>Requirement Analysis</b>	Software Requirements: Functional & non-functional – user-system requirement engineering process – feasibility studies – elicitation – validation & management – software prototyping – S/W documentation – Analysis and modelling Requirement Elicitation, Software requirement specification (SRS), Self-learning Topics: prioritizing requirements (Kano diagram) - real life application case study	09
<b>Module 3</b>	<b>Software Estimation and Scheduling</b>	Management Spectrum, 3Ps (people, product and process) Process and Project metrics Software Project Estimation: LOC, FP, Empirical Estimation Models - COCOMO II Model, Specialized Estimation Techniques, Object based estimation, use-case based estimation Project scheduling: Defining a Task Set for the Software Project, Timeline charts, Tracking the Schedule, Earned Value Analysis Self-learning Topics: Cost Estimation Tools	04

		and Techniques, Typical Problems with IT Cost Estimates.	
<b>Module 4</b>	<b>Design Engineering</b>	Design Process & quality, Design Concepts, The design Model, Pattern-based Software Design. 4.2 Architectural Design :Design Decisions, Views, Patterns, Application Architectures, Modeling Component level Design: component, Designing class based components, conducting component-level design, User Interface Design: The golden rules, Interface Design steps & Analysis, Design Evaluation Self-learning Topics: Refinement, Aspects, Refactoring	07
<b>Module 5</b>	<b>Software Risk, Configuration Management</b>	Risk Identification, Risk Assessment, Risk Projection, RMMM Software Configuration management, SCM repositories, SCM process Software Quality Assurance Task and Plan, Metrics, Software Reliability, Formal Technical Review (FTR), Walkthrough Self-learning Topics:: Configuration management for WebApps	06
<b>Module 6</b>	<b>Software Testing and Maintenance</b>	Testing: Software Quality, Testing: Strategic Approach, Strategic Issues- Testing: Strategies for Conventional Software, Object oriented software, Web Apps Validating Testing- System Testing- Art of Debugging. Maintenance : Software Maintenance-Software Supportability- Reengineering- Business Process Reengineering- Software Reengineering- Reverse Engineering- Restructuring- Forward Engineering Self-learning Topics: Test Strategies for WebApps	06
		<b>Total</b>	<b>39</b>

<b>5) Textbooks:</b>	
1	1 Roger S. Pressman, Software Engineering: A practitioner's approach, McGraw Hill
2	2 Rajib Mall, Fundamentals of Software Engineering, Prentice Hall India
3	3 PankajJalote, An integrated approach to Software Engineering, Springer/Narosa.
4	4 Ian Sommerville, Software Engineering, Addison-Wesley. William Stallings, Cryptography and Network Security, Principles and Practice, 6th Edition, Pearson Education, March 2013.
<b>6) References</b>	
1	<a href="https://www.youtube.com/watch?v=wEr6mwquPLY">https://www.youtube.com/watch?v=wEr6mwquPLY</a>

**7) Internal Assessment:**

Assessment consists of one )Mid Term Test of 20 marks and Continuous Assessment of 20 marks.(Total 40

Mid Term test is to be conducted when approx. 50% syllabus is completed Duration of the midterm test shall be one hour.

**8) Continuous Assessment:-**

Continuous Assessment **is of 20 marks.** The rubrics for assessment will be considered on approval by the subject teachers. The rubrics can be any 2 or max 4 of the following:-

Sr.no	Rubrics	Marks
1.	*Certificate course for 4 weeks or more:- NPTEL/ Coursera/ Udemy/any MOOC	10 marks
2	Case studies + Assignment	10 marks

\*For sr.no.1, the date of the certification exam should be within the term and in case a student is unable to complete the certification , the grading has to be done accordingly.

**9)Rubrics for slow learners:-**

1.) Case study, Presentation, group discussion, technical debate on recent trends in the said course (10 marks)

2. Project based Learning and evaluation / Extra assignment / Question paper solution (10 marks)

<b>11)End Semester Theory Examination:</b>	
--	--

1	Question paper will be of 60 marks
2	Question paper will comprise a total of five questions
3	All question carry 20 marks
4	Any three questions out of five need to be solved.