

<b>SOFTWARE ENGINEERING</b> <b>[As per Choice Based Credit System (CBCS) scheme]</b> <b>(Effective from the academic year 2016 -2017)</b> <b>SEMESTER – IV</b>			
Subject Code	15CS42	IA Marks	20
Number of Lecture Hours/Week	04	Exam Marks	80
Total Number of Lecture Hours	50	Exam Hours	03
<b>CREDITS – 04</b>			
<b>Course objectives:</b> This course will enable students to <ul style="list-style-type: none"> <li>Outline software engineering principles and activities involved in building large software programs.</li> <li>Identify ethical and professional issues and explain why they are of concern to software engineers.</li> <li>Describe the process of requirements gathering, requirements classification, requirements specification and requirements validation.</li> <li>Differentiate system models, use UML diagrams and apply design patterns.</li> <li>Discuss the distinctions between validation testing and defect testing.</li> <li>Recognize the importance of software maintenance and describe the intricacies involved in software evolution.</li> <li>Apply estimation techniques, schedule project activities and compute pricing.</li> <li>Identify software quality parameters and quantify software using measurements and metrics.</li> <li>List software quality standards and outline the practices involved.</li> <li>Recognize the need for agile software development, describe agile methods, apply agile practices and plan for agility.</li> </ul>			
<b>Module 1</b>			<b>Teaching Hours</b>
<b>Introduction:</b> Software Crisis, Need for Software Engineering. Professional Software Development, Software Engineering Ethics. Case Studies. <b>Software Processes:</b> Models: Waterfall Model (Sec 2.1.1), Incremental Model (Sec 2.1.2) and Spiral Model (Sec 2.1.3). Process activities. <b>Requirements Engineering:</b> Requirements Engineering Processes (Chap 4). Requirements Elicitation and Analysis (Sec 4.5). Functional and non-functional requirements (Sec 4.1). The software Requirements Document (Sec 4.2). Requirements Specification (Sec 4.3). Requirements validation (Sec 4.6). Requirements Management (Sec 4.7).			<b>12 Hours</b>
<b>Module 2</b>			
<b>System Models:</b> Context models (Sec 5.1). Interaction models (Sec 5.2). Structural models (Sec 5.3). Behavioral models (Sec 5.4). Model-driven engineering (Sec 5.5). <b>Design and Implementation:</b> Introduction to RUP (Sec 2.4), Design Principles (Chap 17). Object-oriented design using the UML (Sec 7.1). Design patterns (Sec 7.2). Implementation issues (Sec 7.3). Open source development (Sec 7.4).			<b>11 Hours</b>
<b>Module 3</b>			
<b>Software Testing:</b> Development testing (Sec 8.1), Test-driven development (Sec 8.2), Release testing (Sec 8.3), User testing (Sec 8.4). Test Automation (Page no 42, 70,212, 231,444,695). <b>Software Evolution:</b> Evolution processes (Sec 9.1). Program evolution dynamics (Sec			<b>9 Hours</b>

<b>9.2). Software maintenance (Sec 9.3). Legacy system management (Sec 9.4).</b>	
<b>Module 4</b>	
<b>Project Planning:</b> Software pricing (Sec 23.1). Plan-driven development (Sec 23.2). Project scheduling (Sec 23.3): Estimation techniques (Sec 23.5). <b>Quality management:</b> Software quality (Sec 24.1). Reviews and inspections (Sec 24.3). Software measurement and metrics (Sec 24.4). Software standards (Sec 24.2)	<b>10 Hours</b>
<b>Module 5</b>	
<b>Agile Software Development:</b> Coping with Change (Sec 2.3), The Agile Manifesto: Values and Principles. Agile methods: SCRUM (Ref “ <b>The SCRUM Primer, Ver 2.0</b> ”) and Extreme Programming (Sec 3.3). Plan-driven and agile development (Sec 3.2). Agile project management (Sec 3.4), Scaling agile methods (Sec 3.5):	<b>8 Hours</b>
<b>Course Outcomes:</b> After studying this course, students will be able to:	
<ul style="list-style-type: none"> <li>• Design a software system, component, or process to meet desired needs within realistic constraints.</li> <li>• Assess professional and ethical responsibility</li> <li>• Function on multi-disciplinary teams</li> <li>• Use the techniques, skills, and modern engineering tools necessary for engineering practice</li> <li>• Analyze, design, implement, verify, validate, implement, apply, and maintain software systems or parts of software systems.</li> </ul>	
<b>Graduate Attributes</b>	
<ul style="list-style-type: none"> <li>• Project Management and Finance</li> <li>• Conduct Investigations of Complex Problems</li> <li>• Modern Tool Usage</li> <li>• Ethics</li> </ul>	
<b>Question paper pattern:</b>	
<p>The question paper will have ten questions.  There will be 2 questions from each module.  Each question will have questions covering all the topics under a module.  The students will have to answer 5 full questions, selecting one full question from each module.</p>	
<b>Text Books:</b>	
<ol style="list-style-type: none"> <li>1. Ian Sommerville: Software Engineering, 9th Edition, Pearson Education, 2012. (Listed topics only from Chapters 1,2,3,4, 5, 7, 8, 9, 23, and 24)</li> <li>2. The SCRUM Primer, Ver 2.0, <a href="http://www.goodagile.com/scrumpriemer/scrumpriemer20.pdf">http://www.goodagile.com/scrumpriemer/scrumpriemer20.pdf</a></li> </ol>	
<b>Reference Books:</b>	
<ol style="list-style-type: none"> <li>1. Roger S. Pressman: Software Engineering-A Practitioners approach, 7th Edition, Tata McGraw Hill.</li> <li>2. Pankaj Jalote: An Integrated Approach to Software Engineering, Wiley India</li> </ol>	
<b>Web Reference for eBooks on Agile:</b>	
<ol style="list-style-type: none"> <li>1. <a href="http://agilemanifesto.org/">http://agilemanifesto.org/</a></li> <li>2. <a href="http://www.jamesshore.com/Agile-Book/">http://www.jamesshore.com/Agile-Book/</a></li> </ol>	