JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTHAPURAMU
COLLEGE OF ENGINEERING (AUTONOMOUS) :: PULIVENDULA
Regulation –R15

I. Course Overview

Course Code	**:**	15ACS21			
Course Title	**:**	**Software Engineering**			
Course Structure	**:**	Lectures	Tutorials	Practical's	Credits
		4	0	0	4
Course Coordinator	**:**	Miss S. GHOUHAR TAJ			
Team of instructor	**:**	Mr. G. Murali			

The main objective of the course is to help the students in understanding the process of developing a software system from scratch and guiding them through the development process by giving them the fundamental principles of system development. The course will initiate students to the different software process models, project management, software requirements engineering process, systems analysis and design as a problem-solving activity, key elements of analysis and design, and the place of the analysis and design phases within the system development life cycle.

II. Prerequisite(s):

Level	Credits	Periods/Week	Prerequisites
UG	4	4	Awareness of different System Software's and application Software's

III. Assessment:

FORMATIVE ASSESSMENT	
Mid Semester Subjective Test I for 20 Marks in first 2 units is conducted at8 the end of 9th week. (Subjective paper shall contain 5 questions of 2marks and 3 question of 5marks,student has to answer 2 questions)	**20 Marks**
Mid Semester Objective Test I for 10 Marks in first 2 units is conducted at8 the end of 9th week	**10 Marks**

(Objective paper is set for 20 bits) **Total**	 **30 Marks**
Mid Semester Subjective Test II for 20 Marks in first 2 units is conducted at the end of the course (Subjective paper shall contain 5 questions of 2marks and 3 question of 5marks,student has to answer 2 questions) **Mid Semester Objective Test II for 10 Marks in first 2 units is conducted at the end of the course** **(**Objective paper is set for 20 bits) **Total**	 **20 Marks** **10 Marks** **30 Marks**
Final Internal marks for a total of 30marks shall be arrived at by considering the marks secured by the student in both the mid examinations with **80% weight-age to the better mid exam and 20% to the other.**	

SUMMATIVE ASSESSMENT	
End Semester Examination in all units is conducted for 70 Marks	**70 Marks**
Grand Total	**100 Marks**

IV. Course Objectives

1. To understand the software life cycle models.
2. To understand the software requirements and SRS document.
3. To understand the importance of modeling and modeling languages.
4. To design and develop correct and robust software products.
5. To understand the quality control and how to ensure good quality software.
6. To understand the planning and estimation of software projects.
7. To understand the implementation issues, validation and verification procedures.
8. To understand the maintenance of software

V. Course Outcomes

1. Acquire Knowledge about different fundamental software process models.
2. Define and develop a software project from requirement gathering to implementation.
3. Obtain knowledge about design principles and practices of software engineering.
4. Focus on the fundamentals of modeling and testing a software project.
5. Gain knowledge about estimation and maintenance of software systems

VI. Program Outcomes:

a. An ability to apply knowledge of computing, mathematical foundations, algorithmic principles, and computer science and engineering theory in the modeling and design of computer-based systems to real-world problems (fundamental engineering analysis skills)
b. An ability to design and conduct experiments, as well as to analyze and interpret data (information retrieval skills)
c. An ability to design , implement, and evaluate a computer-based system, process, component, or program to meet desired needs, within realistic constraints such as economic, health and safety, manufacturability, and sustainability (Creative Skills)
d. An ability to function effectively on multi-disciplinary teams (team work)
e. An ability to analyze a problem, identify, formulate and use the appropriate computing and engineering skills for obtaining its solution (engineering problem solving skills)
f. Obtaining the knowledge of algorithmic skills regarding data structures. (program oriented skills)
g. An ability to communicate effectively both in writing and orally (speaking / writing skills)
h. The broad education necessary to analyze the local and global impact of computing and engineering solutions on individuals, organizations, and society (engineering impact assessment skills)

i. Recognition of the need for, and an ability to engage in continuing professional development and life-long learning (continuing education awareness)
j. A Knowledge of structural skills which are related to theoretical skills for programming (detailed subject oriented skills).
k. An ability to use current techniques, skills, and tools necessary for computing and engineering practice (practical engineering analysis skills)
l. An ability to apply design and development principles in the construction of software and hardware systems of varying complexity (software hardware interface)
m. An ability to recognize the importance of professional development by pursuing postgraduate studies or face competitive examinations that offer challenging and rewarding careers in computing (successful career and immediate employment)

Syllabus

JAWAHARLAL NEHRU TECHNOLOGICAL UNIVERSITY ANANTHAPURAMU

COLLEGE OF ENGINEERING (AUTONOMOUS) :: PULIVENDULA

Regulation –R15

B.Tech. III Year –I Sem (C.S.E) **L T P C**

4 0 0 4

SOFTWARE ENGINEERING

UNIT I

Software and Software Engineering: The Nature of Software, The Unique Nature of Web Apps, Software Engineering, Software Process, Software Engineering Practice, Software Myths.
Process Models: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models, Specialized Process Models, The Unified Process, Personal and Team Process Models, Process Terminology, Product and Process, Agile Process model, Extreme Programming, A comparison of different process models.

UNIT II

Software Project Planning and Management: Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation, Project Estimation Techniques, Empirical Estimation Techniques, COCOMO-A Heuristic Estimation Technique, Halstead‘s Software Science-An Analytical Technique, Staffing Level Estimation, Scheduling, Organization and Team Structures, Staffing, Risk Management, Software Configuration Management

UNIT III

Understanding Requirements: Requirements Engineering, Establishing the Groundwork, Eliciting Requirements, Requirements Analysis, Structured Analysis, Data Oriented Analysis, Object oriented AnalysisDeveloping Use Cases, Building the Requirements Model, NegotiatingRequirements, and Validating Requirements.
Requirements Modeling: Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case, Data Modeling Concepts, Class-Based Modeling.

UNIT IV

Design Concepts: Design within the Context of Software Engineering, Design Process, Design Concepts, The Design Model, Function oriented software design, Object oriented software development.
Architectural Design: Software Architecture, Architectural Genres, Architectural Styles,Architectural Design.
Component-Level Design: What is a Component, Designing Class-Based Components, Conducting Component-Level Design, Component-Level Design for WebApps

UNIT V

User Interface Design: The Golden Rules, User Interface Analysis and Design, InterfaceAnalysis, Interface Design Steps, Design Evaluation.

Coding and Testing: Coding, Code Review, Software Documentation, Testing, Testing in the Large versus Testing in the Small, Unit Testing, Black-Box Testing, White-Box Testing, Debugging, Program Analysis Tools, Integration Testing, Testing Object-Oriented Programs, System Testing, Some General Issues Associated with Testing.

Software Maintenance: Characteristics of Software Maintenance, Software Reverse Engineering, Software Maintenance Process Models, Estimation of Maintenance cost.

TEXT BOOKS :

1. Software Engineering A practitioner's Approach, Roger S. Pressman, Seventh Edition, 2009, McGrawHill International Edition.
2. Fundamentals of Software Engineering, Rajib Mall, Third Edition, 2009, PHI

REFERENCE BOOKS:

1. Software Engineering, Ian Sommerville, Ninth edition, Pearson education.
2. Software Engineering : A Primer, Waman S Jawadekar, Tata McGraw-Hill, 2008
3. Software Engineering, A Precise Approach, Pankaj Jalote, Wiley India, 2010.
4. Software Engineering, Principles and Practices, Deepak Jain, Oxford University Press.
5. Software Engineering1: Abstraction and modeling, Diner Bjorner, Springer International edition, 2006.
6. Software Engineering2: Specification of systems and languages, Diner Bjorner, Springer International edition , 2006.
7. Software Engineering Foundations, Yingxu Wang, Auerbach Publications, 2008.
8. Software Engineering Principles and Practice, Hans Van Vliet, 3rd edition, John Wiley &Sons Ltd.
9. Software Engineering 3: Domains, Requirements,and Software Design, D.Bjorner, Springer International Edition.
10. Introduction to Software Engineering, R.J.Leach, CRC Press.

VII. Course Plan:

Lecture No	Course Outcomes	Topics to be Covered	Reference
		UNIT – I	
1-2	Acquire Knowledge on the fundamental characteristics of Software	**Software and Software Engineering**: The Nature of Software, The Unique Nature of Web Apps, Software Engineering,	**T1:** Chapter 1, 1.1 -1.3 R1: chapter 1, 1.1, 1.2
3-4	An Understanding of software process and Myths	Software Process, Software Engineering Practice, Software Myths**.**	**T1**: Chapter 1, 1.4 -1.6 R4: chapter 1, 1.8,
5-6	Obtain Knowledge on various Perspective process models	**Process Models**: A Generic Process Model, Process Assessment and Improvement, Prescriptive Process Models (incremental Process models)	**T1:** Chapter 2, 2.1 -2.3.2 **T2:** Chapter 2, 2.2 R1:chapter 2,2.1 R4 : chapter 2, 2.2, 2.3
7-8	An Understanding of Specialized and Unified process models	Prescriptive Process Models (evolutionary Process models), Specialized Process Models, The Unified Process	**T1:** Chapter 2, 2.3.3 -2.5 **T2:**Chapter 2, 2.3-2.5
9-10	Ability to demonstrate team work	Personal and Team Process Models, Process Terminology, Product and Process.	**T1:** Chapter 2, 2.5 -2.8
		UNIT – II	
11-12	Obtain Knowledge on the roles and responsibilities of software project manager	**Software Project Management:** Responsibilities of a Software Project Manager, Project Planning, Metrics for Project Size Estimation	**T2:** Chapter 3, 3.1-3.4 **T1:** Chapter 25, 25.2 **R1:**chapter 22 22.1, 22.2 R2: chapter 5
13-14	Ability to apply project estimation techniques on real world problems	Project Estimation Techniques, Empirical Estimation Techniques, COCOMO-A Heuristic Estimation Technique,	**T2:** Chapter 3, 3.5-3.7 **T1:** Chapter 26, 26.7
15-16	Ability to assess the resource requirement and schedules of the project	Halstead‘s Software Science-An Analytical Technique, Staffing Level Estimation, Scheduling	**T2:** Chapter 3, 3.8-3.10 **T1:** Chapter 26, 26.9
17-18	Acquire knowledge on	Organization and Team Structures, Staffing, Risk	**T2:** Chapter 3, 3.11-3.14

	the software configuration management	Management, Software Configuration Management	
UNIT – III			
19-20	Gain Knowledge on the process of Requirement Engineering	**Understanding Requirements:** Requirements Engineering, Establishing the Groundwork, Eliciting Requirements,	**T1:** Chapter 5, 5.1 -5.3 **T2:** Chapter 4, 4.1
21-22	Apply Creative skills in developing Requirement Engineering Models	Developing Use Cases, Building the Requirements Model,	**T1**: Chapter 5, 5.4 -5.5
23-24	Ability to Validate Requirements	Negotiating Requirements, Validating Requirements	**T1**: Chapter 5, 5.6 -5.7
25-26	Analyze Requirements for modeling use cases	**Requirements Modeling:** Requirements Analysis, Scenario-Based Modeling, UML Models That Supplement the Use Case	**T1**: Chapter 6, 6.1 -6.3 **T2**: Chapter 7, 7.4
27-28	An Understanding of data based modeling and class based modeling	Data Modeling Concepts, Class-Based Modeling	**T1**: Chapter 6, 6.4 -6.5
UNIT - IV			
29-30	Ability to Understand Basic Design Process	**Design Concepts:** Design within the Context of Software Engineering, Design Process, Design Concepts(modularity)	**T1:** Chapter 8, 8.1 -8.3.5
31-32	Obtain Knowledge on basic design concepts and models	Design Concepts, The Design Model	**T1:** Chapter 8, 8.3.5 -8.4
33-34	An Understanding of various architectural types	**Architectural Design:** Software Architecture, Architectural Genres	**T1:** Chapter 9, 9.1 -9.2 **T2:** Chapter 5, 5.1-5.31 R1: chapter 6,6.1, 6.2
35-36	Ability to design software architecture based on client needs	Architectural Styles, Architectural Design.	**T1:** Chapter 9, 9.3 -9.4
37-38	Gain Knowledge on	**Component-Level Design:**	**T1:** Chapter 10 , 10.1-

	component based software development process	What is a Component, Designing Class-Based Components	10.2 **T2:** Chapter 5, 5.6
39-40	Ability to design architecture for web applications	Conducting Component-Level Design, Component-Level Design for WebApps	**T1:** Chapter 10 , 10.3-10.4
UNIT – V			
41-42	An understanding of rules for User Interface Design	**User Interface Design:** The Golden Rules, User Interface Analysis and Design	**T1:** Chapter 11 , 11.1-11.2 **T2:** Chapter 9, 9.2
43-44	Ability to design and user interface	Interface Analysis, Interface Design Steps	**T1:** Chapter 11 , 11.3-11.4 **T2:** Chapter 9, 9.5
45-47	Obtain Knowledge on design evaluation process	Interface Design Steps (cntd), Design Evaluation	**T1:** Chapter 11 , 11.4 - 11.6
48-51	Ability to conduct code review	**Coding and Testing:** Coding, Code Review, Software Documentation, Testing	**T2:** Chapter 10, 10.1-10.3
52-54	Obtain skills to conduct testing on software	Testing in the Large versus Testing in the Small, Unit Testing, Black-Box Testing, White-Box Testing	**T2:** Chapter 10, 10.4-10.7 **T1**: Chapter 18,18.3
55-57	Ability to analyze and debug a program	Debugging, Program Analysis Tools, Integration Testing,	**T2:** Chapter 10, 10.8-10.10 **T1**:Chapter 17, 17.3
58-60	Ability to test Object oriented Programs	Object-Oriented Programs, System Testing, Some General Issues Associated with Testing.	**T2:** Chapter 10, 10.11-10.13 **T1**: Chapter 19,19.2

VIII. Mapping Course Outcomes leading to the achievement of Program Outcomes:

Course Outcomes	Program Outcomes(PO's)												
	A	B	c	d	e	F	g	h	i	J	k	l	M
(CO's)													
1	H		H		H		S			S	H		S
2		H	S		S	S						H	S
3		S	H		S					S	S	H	S
4	S		S								H	S	
5							S	H					

S = Supportive H = Highly Related

Justification of Course Syllabus Covering Course Outcomes:

- By covering the syllabus, student would be able to acquire knowledge about different conventional software process models and could be able to design and develop software by following certain design principles for real world problems.

Justification of Course Outcomes and Program Outcomes Mapping Table:

- CO-1 is highly related to PO's - a, c, e and k and acts as a supportive for PO's - g, j and m such that Student can be able to gain knowledge on different software process models which helps them in developing the software for real world problems. It also helps them to communicate effectively in team meetings.
- CO-2 is highly related to PO's - b, l and acts as a supportive for PO's - c, e, f and m such that the student will be able to analyze the requirements in solving the various problems that helps in the development of efficient software.
- CO-3 is highly related to PO's - c, l and acts as a supportive for PO's - b, e, j, k and m such that the student gains the knowledge regarding the thumb rules of designing software and will be able to apply those principles in software development.
- CO-4 is highly related to PO - k and acts as a supportive for PO's - a, c and l such that the student will be able to apply various testing techniques on a software to analyze the quality of the software and validate the requirements

- CO-5 is highly related to PO - H and acts as a supportive for PO's - G such that the student will be able to assess the cost of software development and take necessary measures in maintaining software at Organizational level