INSTITUTE OF AERONAUTICAL ENGINEERING

(Autonomous)
Dundigal, Hyderabad - 500 043

INFORMATION TECHNOLOGY

COURSE DESCRIPTION FORM

Course Title	**SOFTWARE ENGINEERING**			
Course Code	A50518			
Regulation	**R15- JNTUH**			
Course Structure	Lectures	Tutorials	Practical's	Credits
	5	-	-	4
Course Coordinator	Dr. I Surya Prabha , Professor			

I. COURSE OVERVIEW:

Software Engineering comprises the core principles consistent in software construction and maintenance: fundamental software processes and life-cycles, mathematical foundations of software engineering, requirements analysis, software engineering methodologies and standard notations, principles of software architecture and re-use, software quality frameworks and validation, software development, and maintenance environments and tools. An introduction to object-oriented software development process and design.

II. PREREQUISITE(S):

Level	Credits	Periods/ Week	Prerequisites
UG	4	5	OOAD

III. MARKS DISTRIBUTION:

Sessional Marks	University End Exam marks	Total marks
Midterm Test There shall be two midterm examinations. Each midterm examination consists of essay paper, objective paper and assignment. The essay paper is for 10 marks of 60 minutes duration and shall contain 4 questions. The student has to answer 2 questions, each carrying 5 marks. The objective paper is for 10 marks of 20 minutes duration. It consists of 10 multiple choice and 10 fill-in-the blank questions, the student has to answer all the questions and each carries half mark. First midterm examination shall be conducted for the first two and half units of syllabus and second midterm examination shall be conducted for the remaining portion. Five marks are earmarked for assignments. There shall be two assignments in every theory course. Assignments are usually issued at the time of commencement of the semester. These are of problem solving in nature with	75	100

Sessional Marks	University End Exam marks	Total marks
critical thinking. Marks shall be awarded considering the average of two midterm tests in each course.		

IV. EVALUATION SCHEME:

S. No	Component	Duration	Marks
1.	I Mid Examination	80 minutes	20
2.	I Assignment	-	5
3.	II Mid Examination	80 minutes	20
4.	II Assignment	-	5
5.	External Examination	3 hours	75

V. COURSE OBJECTIVES:

I. Understand the basic Software engineering methods and practices, and its applications.
II. Demonstrate the necessary understanding of methods and techniques for software management, and also to be able to use these in various development situations,
III. Analyze and understand software measurement and software risks.
V. Understanding the role of project management including planning, scheduling, risk management.
VI. Understand the implementation of different software architectural styles.

VI. COURSE OUTCOMES:

At the end of the course the students are able to:

1. **Explain** fundamental knowledge in mathematics, programming and computer systems.
2. **Apply** basic knowledge and understanding of the analysis, synthesis and design of complex systems.
3. **Apply** software engineering principles and techniques.
4. **Design** and evaluate large-scale software systems.
5. **Apply** the notations used to analyze the performance of algorithms.
6. **Demonstrate** ethical standards and legal responsibilities.
7. **Explain** to communicate and coordinate competently by listening, speaking, reading and writing.
8. **Explain** the principles, tools and practices of IT project management.
9. **Illustrate** the managing time, processes and resources effectively by prioritizing competing demands.
10. **Apply** the fundamental knowledge of science in emerging technologies.
11. **Develop** as an effective member or leader of software engineering teams.
12. **Experiment** different testing methods.

VII. HOW PROGRAM OUTCOMES ARE ASSESSED:

Program Outcomes		Level	Proficiency assessed by
PO1	**Engineering knowledge**: Apply the knowledge of mathematics, science, engineering fundamentals, and an engineering specialization to The solution of complex engineering problems.	H	Assignments, Tutorials
PO2	**Problem analysis**: Identify, formulate, review research literature, and analyze complex engineering problems reaching substantiated conclusions using first principles of mathematics, natural sciences, and Engineering sciences.	H	Assignments
PO3	**Design/development of solutions**: Design solutions for complex engineering problems and design system components or processes that meet the specified needs with appropriate consideration for the public health and safety, and the cultural, societal, and environmental Considerations.	S	Mini Projects
PO4	**Conduct investigations of complex problems**: Use research-based knowledge and research methods including design of experiments, analysis and interpretation of data, and synthesis of the information to Provide valid conclusions.	S	Projects
PO5	**Modern tool usage**: Create, select, and apply appropriate techniques, resources, and modern engineering and IT tools including prediction and modeling to complex engineering activities with an understanding of the limitations.	S	Mini Projects
PO6	**The engineer and society**: Apply reasoning informed by the contextual knowledge to assess societal, health, safety, legal and cultural issues and the consequent responsibilities relevant to the professional Engineering practice.	N	--
PO7	**Environment and sustainability**: Understand the impact of the professional engineering solutions in societal and environmental contexts, and demonstrate the knowledge of, and need for sustainable Development.	N	--
PO8	**Ethics**: Apply ethical principles and commit to professional ethics and Responsibilities and norms of the engineering practice.	N	--
PO9	**Individual and team work**: Function effectively as an individual, and as a member or leader in diverse teams, and in multidisciplinary settings.	S	Mini Projects
PO10	**Communication**: Communicate effectively on complex engineering activities with the engineering community and with society at large, such as, being able to comprehend and write effective reports and design documentation, make effective presentations, and give and receive clear instructions.	N	--
PO11	**Project management and finance**: Demonstrate knowledge and understanding of the engineering and management principles and apply these to one's own work, as a member and leader in a team, to manage Projects and in multidisciplinary environments.	H	Mini Projects
PO12	**Life-long learning**: Recognize the need for, and have the preparation and ability to engage in independent and life-long learning in the Broadest context of technological change.	S	Projects

N - None **S - Supportive** **H - Highly Related**

VIII. HOW PROGRAM SPECIFIC OUTCOMES ARE ASSESSED:

Program Specific Outcomes		Level	Proficiency assessed by
PSO1	**Professional Skills:** The ability to understand, analyze and develop computer programs in the areas related to algorithms, system software, multimedia, web design, big data analytics, and networking for efficient analysis and design of computer - based systems of varying complexity.	H	Lectures, Assignments
PSO2	**Software Engineering Practices:** The ability to apply standard practices and strategies in software service management using open-ended programming environments with agility to deliver a quality service for business success.	H	Projects
PSO3	**Successful Career and Entrepreneurship:** The ability to employ modern computer languages, environments, and platforms in creating innovative career paths, to be an entrepreneur, and a zest for higher studies.	S	Guest Lectures

N - None S - Supportive H - Highly Related

IX. SYLLABUS:

UNIT-I

Introduction to Software Engineering: The evolving role of software, Changing Nature of Software, legacy software, Software Myths.

A Generic View of Process: Software engineering-A layered technology, a process framework, The Capability Maturity Model Integration (CMMI), Process Patterns, Process Assessment, personal and team process models.

Process Models: The Waterfall model, Incremental process models, Evolutionary Process Models, Specialized Process Models, The Unified Process.

UNIT-II

Software Requirements: Functional and non-Functional Requirements, User Requirements, System Requirements, Interface Specification, the software requirement document.

Requirement engineering process: Feasibility studies, Requirements elicitation and analysis, requirements validation, Requirements management.

System models: Context Models, behavioral models, Data models, object models, structured method.

UNIT-III

Design Engineering: Design process and design quality, Design concepts, the design model, pattern based software design.

Creating an Architectural Design: Software architecture, Data design, Architectural Styles and patterns, Architectural design, assessing alternative architectural designs, mapping data flow into software architecture.

Modeling Component-level design: designing class –based components, conducting component-level design, object constraint language, designing conventional components.

Performing User interface design: Golden rules, User interface analysis and design, interface analysis, interface design steps, Design evaluation.

UNIT-IV

Testing Strategies: A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, system testing, the art of debugging. **Product metrics**: Software Quality, Frame work for product metrics, Metrics for Analysis Model, Metrics for Design Model, metrics for source code, metrics for testing, metrics for maintenance. Metrics for process and products: Software Measurement, Metrics for software quality.

UNIT-V

Risk management: Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM plan.

Quality Management: Quality concepts, software quality assurance, Software Reviews, Formal technical Reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

Text books:

1. Software engineering a practitioner's approach, Roger S Pressman, sixth edition Mc Graw Hill International Edition.
2. Software Engineering, Ian Sommerville, seventh edition, Pearson education.

References:

1. Software Engineering, a Precise Approach, PankajJalote, Wiley India, 2010.
2. Software Engineering: A primer, Waman S Jawadekar, Tata McGraw-Hill, 2008.
3. Fundamentals of Software Engineering, Rajib Mall, PHI, 2005.

IX. COURSE PLAN:

At the end of the course, the students are able to achieve the following course learning outcomes.

Lecture No.	Topics to be covered	Course Learning Outcomes	Reference
1-4	The evolving role of software, Changing Nature of Software. Legacy software, Software Myths.	**Explain** the evolution of software	T1:1.1 T1:1.4
5	Software engineering-A layered technology, a process framework.	**Explain** process frame work	T1:2.1
6	The Capability Maturity Model Integration (CMMI).	**Illustrate** CMMI	T1:2.3
7-9	Process Patterns, Process Assessment, personal and team process models.	**Explain** process patterns	T1:2.4
10-15	The Waterfall model, Incremental process models, Evolutionary Process Models, Specialized Process Models, The Unified Process.	**Demonstrate** waterfall model, incremental evolutionary, specialized models	T1:3.2
16-18	Functional and non-Functional Requirements	**Distinguish** between Functional and non-Functional Requirements	T2:6.1
19-22	User Requirements, System Requirements. Interface Specification, the software requirement document	**Discuss** user and system requirements, **Explain** software requirement document	T2:6.2
23-25	Requirements elicitation and analysis, requirements validation, Requirements management	**Demonstrate** requirement management	T2:7.2
26-32	Context Models, behavioral models, Data models, object models, structured method	**Demonstrate** Design Engineering	T2:8.1
33-35	Design process and design quality, Design concepts	**Explain** design concepts	T1:9.2
36-37	The design model, pattern based software design	**Explain** software design	T1:9.4

38-41	Software architecture, Data design, Architectural Styles and patterns, Architectural design	**Demonstrate** Architectural Styles and patterns	T1:10.1
42-44	Assessing alternative architectural designs, mapping data flow into software architecture.	**Illustrate** data flow in software architecture	T1:10.5
45-46	Designing class –based components, conducting component-level design, object constraint language, designing conventional components	**Explain** component level design	T1:9.3
47	Golden rules, User interface analysis and design, interface analysis	**Summarize** golden rules	T1:12.1
48	Interface design steps, Design evaluation.	**Explain** interface design	T1:12.3.4
49-50	A strategic approach to software testing, test strategies for conventional software, Black-Box and White-Box testing, Validation testing, system testing, the art of debugging.	**Demonstrate** testing techniques	T1:13.1
51-52	Software Quality, Frame work for product metrics, Metrics for Analysis Model, Metrics for Design Model	**Explain** software quality	T1:15.1
53	Metrics for source code, metrics for testing, metrics for maintenance	**Demonstrate** metrics for testing	T1:15.5
54	Metrics for process and products: Software Measurement, Metrics for software quality	**Explain** metrics of software quality	T1:22.1
55	Reactive vs. Proactive Risk strategies, software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM plan.	**Demonstrate** RMMM	T1:25.1
56-58	Quality concepts, software quality assurance, Software Reviews, Formal technical Reviews	**Explain** quality concepts	T1:26.1
59-60	Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards	**Demonstrate** quality standards	T1:26.6

X. MAPPING COURSE OBJECTIVES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course	Program Outcomes												Program Specific Outcomes		
Objectives	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
I	H	H	S	S					S		H	S	H	S	H
II	S		H						S			H	S	H	H
III			S						H		S	H	S	H	S
IV	H	S		H					S		S		H		H
V	S	S		S					H		H		H	S	S
VI	H	S		H					S		S		S	H	

S - Supportive **H - Highly Related**

XI. MAPPING COURSE OUTCOMES LEADING TO THE ACHIEVEMENT OF PROGRAM OUTCOMES AND PROGRAM SPECIFIC OUTCOMES:

Course	Program Outcomes												Program Specific Outcomes		
Outcomes	PO1	PO2	PO3	PO4	PO5	PO6	PO7	PO8	PO9	PO10	PO11	PO12	PSO1	PSO2	PSO3
1	H	H	S	S					S		H	S	H	S	H
2	S		H						S			H	S	H	H
3			H						H		S	H	H	H	S
4	H	S		S					S		S		S		H
5	S	S		S					H		H		H	S	S
6	H	S		H					S		S		S	H	
7	S		H						S			H	S	H	H
8			H						H		S	H	H	H	S
9	H	H	S	S					S		H	S	H	S	H
10	S	S		S					H		H		H	S	S
11	H	H	S	S					S		H	S	H	S	H
12	S	S		S					H		H		H	S	S

S - Supportive **H - Highly Related**

Prepared by: Dr. I Surya Prabha , Professor

HOD, IT