Subject Code: CSE3001	SOFTWARE ENGINEERING		L,T,P,J,C 2,0,2,4,4
Course Description	This course covers engineering activities involved in software product development with the focus on managing the projects and processes throughout the scope and life time of the software product.		
Objectives	1. To introduce the essential software engineering concepts involved in developing software products and components 2. To impart skills in the design and implementation of efficient software systems across disciplines and also ensure engineering practices and standards.		
Expected Outcome	On completion of this course, the student will be able to A. Explain the principles of the engineering processes in software development B. Implement the software development processes activities from Requirements to Validation & Verification. C. Manage software projects through activities of Estimations, Scheduling, Quality and Software Maintenance		
SLO	1. Having an ability to apply mathematics and science in engineering applications 5. Having design thinking capability 6. Having an ability to design a component or a product applying all the relevant standards and with realistic constraints		

Module	**Topics**	**L Hrs**	**SLO**
1	**OVERVIEW OF SOFTWARE ENGINEERING** Nature of Software, Software Engineering, Software – process, project, product, Process Models – Classical & Evolutionary models, Overview of System Engineering	5	1
2	**INTRODUCTION TO SOFTWARE PROJECT MANAGEMENT** Planning – scope, milestones & deliverables, Risk Management, Metrics & Measurements	3	5, 6
3	**MODELLING – REQUIREMENTS** Requirements Engineering process– Requirement Elicitation, System Modelling - Requirements Specification and Requirement Validation	6	5, 6

4	**SOFTWARE DESIGN** Design concepts and principles - Abstraction - Refinement - Modularity – Cohesion & coupling, Architectural design, Detailed Design – Transaction & Transformation, Refactoring of designs, Object-oriented Design User-Interface Design.	4	6
5	**VALIDATION & VERIFICATION** Strategic Approach to Software Testing, Testing Fundamentals – Test Plan, Test Design, Test Execution, Reviews, Inspection & Auditing	4	5
6	**SOFTWARE EVOLUTION** Software Maintenance, Types of Maintenance, Software Configuration Management, Overview of RE-engineering & Reverse Engineering	4	1
7	**QUALITY ASSURANCE** Product & Process Metrics, Quality Standards & Models –ISO, TQM, Six-Sigma	2	1
8	**RECENT TRENDS** Recent Trends in Software Design/Specialized Software Testing, Related Tools and Standards	2	
Lecture Hours		**30hrs**	
Lab (Indicative List of Experiments (in the areas of) 1. Work Break-down Structure (Process Based, Product Based, Geographic Based and Role Based) 2. Estimations – Cost & Schedule 3. Entity Relationship Diagram, Context flow diagram, DFD (Structural Modelling and Functional Modelling) 4. State Transition Diagrams (Behavioural Modelling) 5. System Requirements Specification		**30hrs**	1,5,6

6. UML diagrams for OO Design 7. Tools for Version Control 8. Black-box, White-box testing 9. Non-functional testing Sample Lab Experiments: **1.** Prepare a WBS for developing a customized social networking portal for your institution **2.** Using the WBS estimate the effort that will be needed to finish the product. Also give a detailed cost estimation & budget for completing this project **3.** Identify the Actors involved, modularize the problem, context of the modules. Draw refined structures of DFD and make a functional model of the system **4.** Impart dynamism to the functional model, so that the system behaves in states and transition according to the requirements **5.** Prepare the complete SRS **6.** Detail the functional model of the system using UML diagrams in the context of OO development **7.** Prepare separate version of the design and the code, and use tools for change management **8.** Validate the functionality of the developed system in conformance with the SRS **9.** Evaluate the performance of the system in terms of load, stress, endurance and scalability		
Project # Generally a team project [3 members] Projects may be given as group projects A software product in any of the following category should be developed 1. Native platform-based application	**60** [Non Contact hrs]	1, 5, 6

2. Web-based Application
3. Mobile App
4. Web-service
5. Software component

- Practice the processes in Requirements phase
- Based on the nature of the product implement the design phase
- Coding & construction based on a suitable language/platform
- Validate and Evaluate the software product
- Prepare a complete documentation for the product (SRS, TRS, Maintenance etc)
- Use DevOps or Bluemix in the entire SDLC for your project

Text Books

1. Roger Pressman, Software Engineering: A Practitioner's Approach, 7th Edition, McGraw-Hill, 2010.

Reference Books

1) Ian Sommerville,Software Engineering, 9th Edition, Addision-Wesley, 2016
2) Pankaj Jalote, A Concise Introduction to Software Engineering, Springer, 2008
3) William E. Lewis , "Software Testing and Continuous Quality Improvement", Third Edition, Auerbach Publications, 2008

Software Engineering

Knowledge Areas that contain topics and learning outcomes covered in the course

Knowledge Area	**Total Hours of Coverage**
CS: SE(Software Engineering) CE: SWE	21
CS: SE(Software Testing)	4
CS: SE(Software Project Management)	5

Body of Knowledge coverage

KA	Knowledge Unit	Topics Covered in CE-BoK	Topics Covered	Hours
CS: SWE	Software Engineering	SWE1-Software Processes SWE2- Software requirements and specifications SWE3- Software design SWE5-Software Evolution	Process Models Modelling - Requirements Software Design Software Evolution	21
CS: SWE	Software Testing	SWE4 - Software testing and validation	Test plan, design & execution Reviews, Inspection & Auditing	4
CS: SWE	Software Project Management	SWE8- Software project management SWE6- Software tools and environments	WBS, Estimation, Scheduling, Software Quality, Software Maintenance	5
			Total hours	**30**

Where does the course fit in the curriculum?

This course is a

- Core Course.
- Suitable from 5th semester onwards.
- Knowledge in any one programming language and concepts of DBMS is desirable.

What is covered in the course?

Part 1: Software Engineering Concepts
Introducing the fundamentals of different process models involved in software development. It covers the system study through requirements gathering and analysis, which leads to modelling the system. The concepts on designing the software product the engineering activities like abstraction, cohesion, Entity-Relations, dataflow and context flow designs are covered. Software maintenance, configuration, reverse and re-engineering processes are completely dealt.

Part II: Software Testing
This section covers the testing fundamentals and the need for testing. Drafting a test plan, designing a complete test cycle and test executions are covered. Basic exposure to software review and auditing are also dealt.

Part III: Software Project Management
This section deals with planning for the software project, risk management with focus on software metrics and measurements.

What is the format of the course?

This Course is designed with 100 minutes of in-classroom sessions per week, 60 minutes of video/reading instructional material per week, 100 minutes of lab hours per week, as well as 200 minutes of non-contact time spent on implementing course related project. Generally this course should have the combination of lectures, in-class discussion, case studies, guest-lectures, mandatory off-class reading material, quizzes.

How are students assessed?

- Students are assessed on a combination group activities, classroom discussion, projects, and continuous, final assessment tests.
- Additional weightage will be given based on their rank in crowd sourced projects/ Kaggle like competitions.
- Students can earn additional weightage based on certificate of completion of a related MOOC course.

Session wise plan

Class Hour	Lab Hour	Topic Covered	levels of mastery	Text/Reference Book	Remarks
2		Nature of Software, Introduction to Software Engineering, Software Product, Project	Usage	1	
3		Software Process Models, Overview of all the engineering activities	Usage	1	
1		Project Planning, milestones & deliverables	Usage	1	
	4	Work Break-down Structure	Usage		LAB Component
	4	Estimations – Cost & Schedule	Usage		LAB Component
1		Risk Management	Usage	1	
1		Metrics & Measurements	Usage	1	
2		Intro to Requirements Engineering, Elicitation.	Usage	1	
2		System Modeling,	Usage	1	
2		SRS, Req. Validation	Usage	1	
2		Design Concepts & Principles - Abstraction - Refinement - Modularity – Cohesion & coupling concepts,	Usage	1	

		Architectural design, Detailed Design			
	4	Entity Relationship Diagram, Context flow diagram Structured and functional modelling	Usage		LAB Component
1		Transaction & Transformation, Refactoring of designs	Usage	1, 2	
	4	State Transition Diagrams Software – Behavioral Modelling Requirements Specification	Usage		LAB Component
1		Object-oriented Design User-Interface Design.	Usage	1	
	6	UML diagrams for OO Design	Usage		LAB Component
1		Strategic Approach to Software Testing, Testing Fundamentals	Usage	1,2,3	
	2	Tools for Version Control	Usage		LAB Component
	2	Black-box, White-box testing	Usage		LAB Component
2		Test Plan, Test Design	Usage	1,2,3	
2		Test Execution, Reviews, Inspection & Auditing	Usage	4	
	4	Functional & Non-	Usage	1,2,3	LAB

		functional testing			Component
2		Software Maintenance, Types of Maintenance	Usage	4	
1		Software Configuration Maintenance,	Usage	4	
1		Overview of RE-engineering & Reverse Engineering	Familiarity	1	
1		Product & Process Metrics	Usage	1, 2	
1		Quality Standards & Models	Familiarity	1	
2		Emerging Trends, Tools & Standards	Familiarity		
30 Hours (2 Credit hours /week □ 15 Weeks schedule)	30 Hours (2 Credit hours / week)				