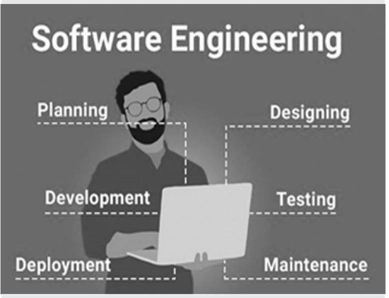

Source: https://artoftesting.com/software-engineering

22CS307 SOFTWARE ENGINEERING

Hours Per Week :

L	T	P	C
2	0	2	3

PREREQUISITE KNOWLEDGE: Data base management systems, OOPs through Java.

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the concepts of software life cycle, role of process models and methods to prepare software requirement specification document. In addition to that, it also imparts knowledge of design, development and testing of software. The objective of this course is to enable the student to develop efficient, cost effective, feasible software as per user requirements.

MODULE - 1

UNIT-1 **8L+0T+8P=16 Hours**

INTRODUCTION

Introduction to Software Engineering: Introduction to Software and Software engineering, Software characteristics, Software project, Software myths, Project Planning, Scheduling and Management.

Generic View of Process: Software Engineering - A layered technology, A process framework, Software Development Life Cycle (SDLC), The Capability Maturity Model Integration (CMMI).

Process Models: Conventional Model, Agile process models - Unified process model, Extreme Programming, Scrum.

UNIT-2 **8L+0T+8P=16 Hours**

REQUIREMENTS ENGINEERING

Requirements Engineering: Functional and Non-functional requirements, User requirements, System requirements, Requirement engineering tasks, formal requirements specification and verification, Feasibility Study, Social impact of software on society.

Building the Analysis Model: Data modeling - Data objects, Attributes, Relationships, Cardinality and modality. Class based modeling - Identify analysis classes, specify attributes and Define operations.

Design Engineering: Design model, Design concepts. Creating an Architectural Design-Architectural styles and patterns.

Performing User Interface Design: Golden rules; User interface analysis and design.

PRACTICES:

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Development of software requirements specification using Mind-Map tool.
- Project planning using Gantt charts.
- Project estimation using metrics.
- Capture Use Case Scenarios and model UML Use Case Diagrams.
- Model the UML state chart and Activity diagrams.
- Model the UML Class and Sequence diagrams.

MODULE - 2

UNIT-1 **8L+0T+8P=16 Hours**

TESTING

Testing Strategies: A strategic approach to software testing, Unit testing, Integration testing, Validation testing, System testing,

Testing Tactics: Black-Box and White-Box testing techniques, Art of debugging.

Product Metrics: Metrics for analysis model; Metrics for design model, Metrics for source code; Metrics for testing; Metrics for maintenance.

UNIT–2 **8L+0T+8P=16 Hours**

RISK AND QUALITY MANAGEMENT

Risk Management: Software risks, Risk identification; Risk projection; Risk refinement, Reactive vs Proactive risk strategies, RMMM.

Quality Management: Quality concepts, Formal technical reviews, Statistical Software Quality Assurance.

Computer-Aided Software Engineering (CASE): Use of appropriate CASE tools- Requirement engineering tools, Project planning tools, Testing tools.

Piracy and Intellectual Property: Privacy concerns in software design, development, and data handling, **Professionalism and Codes of Ethics:** Adhering to professional codes of ethics in software engineering.

PRACTICES:

Laboratory session of this course is designed in such a way that the student should complete three projects of the given type by performing the below experiments.

- Estimate the test coverage and Structural complexity of product using metrics.
- Develop the test cases for all the functional requirements of projects selected.
- Perform the functional testing using Selenium tool.

LIST OF PROJECTS:

Project-1: A Point-Of-Sale (PoS) System: A POS system is a computerized application used to record sales and handle payments; it is typically used in a retail store, it includes hardware components such as a computer and bar code scanner, and software to run the system. It interfaces to various service applications, such as a third-party tax calculator and inventory control. These systems must be relatively fault tolerant; that is, even if remote services are temporarily unavailable they must still be of capturing sales and handling at least cash payments. A POS system must support multiple and varied client-side terminals and interfaces such as browser, PDAs, touch-screens.

Project-2: Online Bookshop Example: Following the model of amazon.com or bn.com, design and implement an online bookstore.

Project-3: A Simulated Company: Simulate a small manufacturing company. The resulting application will enable the user to take out a loan, purchase a machine, and over a series of monthly production runs, follow the performance of their company.

Project-4: A Multi-Threaded Airport Simulation: Simulate the operations in an airport. Your application should support multiple aircrafts using several runways and gates avoiding collisions/ conflicts. Landing: an aircraft uses the runway, lands, and then taxis over to the terminal. Take-Off: an aircraft taxies to the runway and then takes off.

Project-5: An Automated Community Portal: Business in the 21st Century is above all BUSY. Distractions are everywhere. The current crop of “enterprise intranet portals” is often high noise and low value, despite the large capital expenditures it takes to stand them up. Email takes up 30 - 70% of an employee’s time. Chat and Instant Messaging are either in the enterprise or just around the corner. Meanwhile, management is tasked with unforeseen and unfunded leadership and change-agent roles as well as leadership development and succession management. What is needed is a simplified, repeatable process that enhances communications within an enterprise, while allowing management and peers to self-select future leaders and easily recognize high performance team members in a dynamic way. Additionally, the system should function as a general-purpose content management, business intelligence and peer-review application. Glasscode’s goal is to build that system.

Project-6: Content Management System: The goal is to enable non-technical end users to easily publish, access, and share information over the web, while giving administrators and managers complete control over the presentation, style, security, and permissions. Features: Robust Permissions System, Templates for easy custom site designs, Total control over the content, Search engine friendly URL’s, Role based publishing system, Versioning control, Visitor profiling.

Project-7: An Auction Application: Several commerce models exist and are the basis for several

SKILLS:

- ✓ *Know the software requirements and find out various ways to gather and specify them.*
- ✓ *Choose a process model for developing software solutions without schedule/ effort overruns and good quality.*
- ✓ *Analyse and model (diagrammatical representations) a software product.*

companies like eBay.com, pricellne.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.

Project-8: A Notes And File Management System: During one's student years and professional career one produces a 1 lot of personal notes, documents. All these documents are usually kept 1 on papers or individual files on the computer. Either way the bulk of the information is often erased corrupted and eventually lost. The goal of this 1 project is to build a distrib- VFSTR 106 III Year I Semester uted software application that addresses this problem. The system will provide an interface to create, organize and manage personal notes through the Internet for multiple users. The system will also allow users to collaborate by assigning permissions for multiple users to view and edit notes.

Project-9: Library Management System(LMS): The goal is to enable students and librarians to easily access and manage the library and run it smoothly. Each physical library item - book, tape cassette, CD, DVD, etc. could have its own item number. To support it, the items may be barcoded. The purpose of barcoding is to provide a unique and scannable identifier that links the barcoded physical item to the electronic record in the catalog. Barcode must be physically attached to the item, and barcode number is entered into the corresponding field in the electronic item record. Barcodes on library items could be replaced by RFID tags. The RFID tag can contain item's identifier, title, material type, etc. It is read by an RFID reader, without the need to open a book cover or CD/DVD case to scan it with barcode reader.

Project-10: Hospital Management System: Simulate to show and explain hospital structure, staff, and relationships with patients, and patient treatment terminology

Project-11: Draft Software Requirement Analysis for the following Problem Statement: Fuel Delivery System: An unattended petrol (gas) pump system that includes a credit card reader. The customer swipes the card through the reader and then specifies the amount of fuel required. The fuel is delivered, and the customer's account debited.

COURSE OUTCOMES:

Upon successful completion of this course, students will have the ability to:

CO No.	Course Outcomes	Blooms Level	Module No.	Mapping with POs
1	Use basic concepts of software engineering for designing software product.	Usage	1	1, 11
2	Compare different process models and identify appropriate process model based on project requirements.	Evalua-tion	1	2, 4
3	Build Software Requirement Specification (SRS) document for any software product.	Design	1	3, 5
4	Design of solutions using UML diagrams like Use case, Sequence diagrams etc.	Design	1	3, 4, 5
5	Create an appropriate architecture for a given project that meets all quality constraints.	Create	2	5
6	Apply different testing techniques to ensure bug free software and metrics to measure the software size, complexity, and budget etc.	Apply	2	4, 5, 11

TEXT BOOKS:

1. Roger S. Pressman, "Software Engineering, A practitioner's Approach", 6th Edition, McGrawHill International Edition, 2008.
2. Booch G., Rumbaugh J. and Jacobsons I, "The Unified Modeling Language User Guide", 2nd Edition, Addison Wesley, 2005.

REFERENCE BOOKS:

1. Simon Sennet, Steve McRobb and Ray Farmer, "Object Oriented Systems Analysis and Design, 2nd edition, 2004.
2. Dr. Pankaj Jalote "Software Engineering: A Precise Approach"-edition 2010.
3. Yuan Hong and Xiaokui Shu "Privacy Engineering: A Dataflow and Ontological Approach".
4. ACM and IEEE-CS, "Software Engineering Code of Ethics and Professional Practice".