

19CS302 SOFTWARE ENGINEERING



Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P	CS	WA/RA	SSH	SA	S	BS
45	-	30	5	5	30	20	5	5

Source:

[https://
www.computer
hope.com](https://www.computerhope.com)

PREREQUISITE COURSES: Data Base Management Systems, Oops through Java?

COURSE DESCRIPTION AND OBJECTIVES:

This course focuses on the concepts of software development 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 defect free and efficient software as per the given requirements.

COURSE OUTCOMES:

Upon completion of the course, the student will be able to achieve the following outcomes:

COs	Course Outcomes	POs
1	Understand the basic concepts of software engineering.	1, 11
2	Compare different process models and identify appropriate process model based on project requirements.	2,4
3	Build Software Requirement Specification (SRS) document for any software product.	3, 5
4	Design of solution using UML diagrams like use case, sequence diagrams etc.	3,4,5
5	Design suitable architectural that meets all non functional requirements.	5
6	Apply different testing techniques to ensure bug free software and measure metrics such as software size and quality of the product.	4,5,11

SKILLS:

- ✓ Understand 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.

UNIT– I **L- 9**

INTRODUCTION TO SOFTWARE ENGINEERING: The evolving role of software; Software; Changing nature of software; Software myths.

GENERIC VIEW OF PROCESS: Software Engineering - a layered technology; A process framework; Software Development Life Cycle(SDLC); The Capability Maturity Model Integration(CMMI); Process assessment.

PROCESS MODELS: Waterfall model; Incremental process models; Evolutionary process models (Spiral and Prototype models).

UNIT – II **L- 9**

AN AGILE VIEW OF PROCESS: Agile process models - unified process model, extreme programming; Scrum.

REQUIREMENTS ENGINEERING: Inception, Elicitation, Elaboration, Negotiation, Specification (SRS document; IEEE standards for SRS), Validation, Requirements management; Feasibility study.

BUILDING THE ANALYSIS MODEL: Data modeling - data objects, attributes, relationships, cardinality and modality; Class based modeling - identify analysis classes, specify attributes, define operations.

UNIT – III **L- 9**

DESIGN ENGINEERING: Design concepts; The design model.

CREATING AN ARCHITECTURAL DESIGN: Software architecture; Data design; Architectural styles and patterns; Architectural design.

PERFORMING USER INTERFACE DESIGN: Golden rules; User interface analysis and design; Interface analysis; Interface design steps; Design evaluation.

UNIT – IV **L- 9**

PRODUCT METRICS: Software quality; Metrics for analysis model; Metrics for design model; Architecture metrics and Object oriented metrics; Metrics for source code; Metrics for testing; Metrics for maintenance.

TESTING STRATEGIES: A strategic approach to software testing; Test strategies for conventional software; Validation testing; System testing.

TESTING TACTICS: Black-Box and White-Box testing.

UNIT - V **L- 9**

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.

LABORATORY EXPERIMENTS

LIST OF EXPERIMENTS

TOTAL HOURS: 30

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.

1. Identifying the Requirements from Problem Statements.
2. Estimation of Project Metrics.
3. Modeling UML Use Case Diagrams and Capturing Use Case Scenarios.
4. Statechart and Activity Modeling.
5. Modeling UML Class Diagrams and Sequence diagrams.
6. Estimation of Test Coverage Metrics and Structural Complexity.
7. Design of Test Cases.

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 BOOK:

1. Simon Sennet, Steve McRobb and Ray Farmer, "Object Oriented Systems Analysis and Design, 2nd edition, 2004.

LIST OF PROJECTS

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.
2. **Online Bookshop Example:** Following the model of amazon.com or bn.com, design and implement an online bookstore.
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.
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 taxis to the runway and then takes off.
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.
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.
7. **An Auction Application:** Several commerce models exist and are the basis for several companies like eBay.com, priceline.com etc. Design and implement an auction application that provides auctioning services. It should clearly model the various auctioneers, the bidding process, auctioning etc.
8. **A Notes And File Management System:** During one’s student years and professional career one produces a lot of personal notes, documents. All these documents are usually kept 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 project is to build a distrib-

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.

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