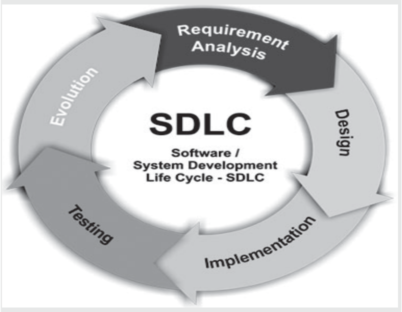
16CS301 SOFTWARE ENGINEERING



Hours Per Week :

L	T	P	C
3	-	2	4

Total Hours :

L	T	P
45	-	30

WA/RA	SSH/HSH	CS	SA	S	BS
5	40	5	8	5	2

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.

Course Outcomes:

The student will be able to:

- prepare a Software Requirement Specification (SRS) document for any software project.
- identify the importance of system analysis and design in solving complex problems.
- distinguish between object-oriented approach and traditional approach in system analysis and design.
- construct various UML models such as use case diagrams, class diagrams, interaction diagrams, state chart diagrams, activity diagrams and implementation diagrams, using appropriate notation.
- understand various metrics to measure software product size and complexity.

SKILLS:

✓ *Define a process for developing/completing different kinds of projects on time with expected quality.*

✓ *Understand the software requirements and find out various ways to gather them and specifying them.*

✓ *Analyze and model (Diagramatical/Representations) a software product.*

✓ *Design an effective, user-friendly interface for a given software product.*

✓ *Find and fix the bugs in a software product.*

UNIT - 1 **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, The Capability Maturity Model Integration (CMMI), Process Assessment.

PROCESS MODELS: The Waterfall Model, Incremental Process Models, Evolutionary Process Models.

UNIT - 2 **L-9**

AN AGILE VIEW OF PROCESS: Agile process models - The Unified process, Extreme Programming, Scrum.

REQUIREMENTS ENGINEERING: Inception, Elicitation, Elaboration, Negotiation, Specification, Validation, Requirements management.

BUILDING THE ANALYSIS MODEL: Data modeling - Data objects, Attributes, Relationship, Cardinality and modality; Class based modeling - Identify analysis classes, Specify attributes and Define operations.

UNIT - 3 **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 - 4 **L-9**

PRODUCT METRICS: Software Quality, Metrics for Analysis Model, Metrics for Design Model, 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 - 5 **L-9**

RISK MANAGEMENT: Reactive vs Proactive Risk strategies, Software risks, Risk identification, Risk projection, Risk refinement, RMMM, RMMM Plan.

QUALITY MANGEMENT: Quality concepts, Software quality assurance, Software Reviews, Formal technical reviews, Statistical Software quality Assurance, Software reliability, The ISO 9000 quality standards.

ACTIVITIES:

- *Identify the customer requirements and define the process of developing the product.*
- *Draft the system and software requirements and prepare SRS document.*
- *Design the project using UML diagrams.*
- *Prepare test cases for testing the software product to ensure that it is defect-free.*
- *Make a list of possible risks and prepare a mitigation plan.*

LABORATORY EXPERIMENTS

Course Outcomes:

The student will be able to:

- Construct various UML models (including use case diagrams, class diagrams, interaction diagrams, statechart diagrams, activity diagrams, and implementation diagrams) using the appropriate notation
- Understand various indirect metrics to measure software product size and complexity and find & fix the bugs in any software product.

List of Experiments:

Total Hours:30

Laboratory session of this course is designed in such a way that the student completes three tasks of the type given below. Ten tasks are given below for the sake of providing guidelines to faculty members and students, on the type of tasks that can be implemented. They are expected to add/modify them to implement their own.

TASK - 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.

TASK - 2

ONLINE BOOKSHOP EXAMPLE: Following the model of amazon.com or bn.com, design and implement an online bookstore.

TASK - 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.

TASK - 4

A MULTI-THREADED AIRPORT SIMULATION: Simulate the operations in an airport. Your application shouldsupport 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.

TASK- 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. The software is released under a proprietary license, and will have the following features: Remote, unattended moderation of discussions However, it will have powerful discovery and business intelligence features, and be infinitely extendable, owing to a powerful API and adherence to Java platform standards. Encourages peer review and indicates for management potential leaders, strong team players and reinforces enterprise and team goals seamlessly and with zero administration.

TASK - 6

A 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

TASK - 7

AN AUCTION APPLICATION: Several commerce models exist and are the basis for a number of 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.

TASK- 8

A NOTES AND FILE MANAGEMENT SYSTEM: In the course of 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 distributed 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.

TASK- 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.

TASK - 10

HOSPITAL MANAGEMENT SYSTEM: Simulate to show and explain hospital structure, staff, and relationships with patients, and patient treatment terminology.

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, "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 using UML", 2nd edition, McGraw Hill, 2002.
2. Sommervelle, "Software Engineering", 7th edition, Pearson education, 2008.
3. Shely Cashman Rosenblatt, "Systems Analysis and Design", 1st edition, Thomson Publications, 2006.