DISCIPLINE SPECIFIC CORE COURSE– 15 (DSC-15/DSC-A7): Software Engineering

Credit distribution, Eligibility and Pre-requisites of the Course

Course title & Code	Credits	Credit distribution of the course			Eligibility criteria	Pre-requisite of the course
		Lecture	Tutorial	Practical/ Practice		
DSC 15/DSC-A7 Software Engineering		**3**	**0**	**1**	Pass in Class XII	DSC01 Programming using Python/ DSC04 Object Oriented Programming with C++/A course in C/C++ or Python at plus 2 level

Learning Objectives

This course will acquaint the student with different approaches and techniques used to develop good quality software. The course includes learning of various software development process frameworks, requirement analysis, design modeling, qualitative and quantitative software metrics, risk management, and testing techniques.

Learning outcomes

On successful completion of the course, a student will be able to:

- describe the software development models.
- analyse and model customer requirements and build design models.
- estimate and prepare schedule for software projects.
- analyse the impact of risks involved in software development.
- design and build test cases, and perform software testing.

SYLLABUS OF DSC 15

Unit 1 (9 hours)

Introduction: Software Engineering - A Layered Approach; Software Process – Process Framework, Umbrella Activities; Process Models – Waterfall Model, Incremental Model, and Evolutionary process Model (Prototyping, Spiral Model); Introduction to Agile, Agile Model – Scrum.

Unit 2 (6 hours)

Software Requirements Analysis and Specification: Use Case Approach, Software Requirement Specification Document, Flow-oriented Model, Data Flow Model

Unit 3 (8 hours)

Design Modeling: Translating the Requirements model into the Design Model, The Design Process, Design Concepts - Abstraction, Modularity and Functional Independence; Structure Charts.

Unit 4 (7 hours)

Software Metrics and Project Estimation: Function based Metrics, Software Measurement, Metrics for Software Quality; Software Project Estimation (FP based estimations); Project Scheduling (Timeline charts, tracking the schedule).

Unit 5 (5 hours)

Quality Control and Risk Management: Quality Control and Quality Assurance, Software Process Assessment and Improvement; Software Risks, Risk Identification, Risk Projection, Risk Mitigation, Monitoring and Management.

Unit 6 (10 hours)

Software Testing: Strategic Approach to Software Testing, Unit Testing, Integration Testing, Validation Testing, System Testing; Black-Box and White Box Testing, Basis Path Testing.

Essential/recommended readings

1. Pressman, R.S. *Software Engineering: A Practitioner's Approach*, 9th edition, McGraw-Hill, 2020.
2. Aggarwal, K.K., Singh, Y. *Software Engineering*, 3rd edition, New Age International Publishers, 2007.
3. Jalote, P. An Integrated Approach to Software Engineering, 3rd Edition, Narosa Publishing House, 2005.

Additional References

1. Sommerville, I. *Software Engineering*, 9th edition, Addison Wesley, 2011.
2. *The Definitive Guide to Scrum: The Rules of the Game*, Ken Schwaber, Jeff Sutherland, July 2016.

Suggested Practical List :(30 Hours)

Practical exercises such as

The students document, design and code a module of a Software Project using an appropriate Software Process model. The Software Project should include the use of software engineering tools and include.

1. Problem Statement, Process Model
2. Requirement Analysis: Create Data Flow, Data Dictionary, Use Cases, Sequence Diagram, Software Requirement Specification Document
3. Project Management: Timeline Chart, Compute FP, Effort, Cost, Risk Table.
4. Design Engineering: Architectural Design, Pseudocode of a small module.
5. Coding: Develop at least a single module using any programming Language
6. Testing: Compute Basic path set for at least one module from a project, Generate test cases.

Some of the sample projects are given below:

1. Criminal Record Management: Implement a criminal record management system for jailers, police officers and CBI officers
2. DTC Route Information: Online information about the bus routes and their frequency and fares.
3. Car Pooling: To maintain a web-based intranet application that enables the corporate employees within an organization to avail the facility of carpooling effectively.
4. Patient Appointment and Prescription Management System
5. Organized Retail Shopping Management Software
6. Online Hotel Reservation Service System
7. Examination and Result computation System
8. Automatic Internal Assessment System
9. Parking Allocation System