DSE-2A(1)

Course code: CS-521 T **Course Title: A1. Software Engineering**

Total Credit: 2 Marks: 50 (UA: 40 + IA: 10)

Periods: 3 per week (50 Minutes each)

Prerequisites

1. Object Oriented Programming fundamental
2. UML

Course Objectives

- To study Software Development Life Cycle, Development models and Agile Software development.
- To study fundamental concepts in software testing, including software testing objectives, process, criteria, strategies, and methods.
- To discuss various software testing issues and solutions in software unit test; integration,
- regression, and system testing.
- To learn the process of improving the quality of software work products.
- To gain the techniques and skills on how to use modern software testing tools to support software testing projects.
- To expose Software Process Improvement and Reengineering.

Learning Outcomes

After learning the course the students should be able to:

- Prepare SRS (Software Requirement Specification) document and SPMP (Software Project Management Plan) document.
- Apply the concept of Functional Oriented and Object Oriented Approach for Software Design.
- Recognize how to ensure the quality of software product, different quality standards and software review techniques.
- Apply various testing techniques and test plan in.
- Able to understand modern Agile Development

Course Outline

Unit – 1 Introduction to Software and Software Engineering

The Evolving Role of Software, Software: A Crisis on the Horizon and Software Myths, Software Engineering: A Layered Technology, Software Process Models, The Linear Sequential Model, The Prototyping Model, The RAD Model, Evolutionary Process Models, Agile Process Model, Component - Based Development, Process, Product and Process.Agility and Agile Process model, Extreme Programming, Other process models of Agile Development and Tools.

Unit - 2 Managing Software Project

Software Metrics (Process, Product and Project Metrics), Software Project Estimations, Software Project Planning (MS Project Tool), Project Scheduling & Tracking, Risk Analysis & Management (Risk Identification, Risk Projection, Risk Refinement , Risk Mitigation). Understanding the Requirement, Requirement Modelling, Requirement Specification (SRS), Requirement Analysis and Requirement Elicitation, Requirement Engineering. Design Concepts and Design Principal, Architectural Design,Component Level Design, User Interface Design, Web Application Design.

Unit - 3 Software Coding & Testing

Coding Standard and coding Guidelines, Code Review, Software Documentation, Testing Strategies, Testing Techniques and Test Case, Test Suites Design, Testing Conventional Applications, Testing Object Oriented Applications, Testing Web and Mobile Applications, Testing Tools (Win runner, Load runner). Quality Concepts and Software Quality Assurance, Software Reviews (Formal Technical Reviews), Software Reliability, The
Quality Standards: ISO 9000, CMM, Six Sigma for SE, SQA Plan.

Unit - 4 Software Maintenance and Configuration Management

Types of Software Maintenance,The SCM Process,Identification of Objects in the Software Configuration, **DevOps:** Overview, Problem Case Definition, Benefits of Fixing Application Development Challenges, DevOps Adoption Approach through Assessment, Solution Dimensions, What is DevOps?, DevOps Importance and Benefits, DevOps Principles and Practices, 7 C's of DevOps Lifecycle for Business Agility, DevOps and Continuous Testing, How to Choose Right DevOps Tools, Challenges with DevOps Implementation, Must Do Things for DevOps, Mapping My App to DevOps -

Unit - 5 Test and Tutorial

Reference Books:-

1. Roger S.Pressman, Software Engineering- A practitioner's Approach, McGraw-Hill International Editions
2. Ian Sommerville, Software engineering, Pearson education Asia
3. Pankaj Jalote, Software Engineering – A Precise Approach Wiley
4. Behhforoz& Frederick Hudson, Software Engineering Fundamentals, OXFORD
5. Rajib Mall, Fundamentals of software Engineering, Prentice Hall of India.
6. Deepak Gaikwad, Viral Thakkar, DevOps Tools from Practitioner's ViewPoint, Wiley
7. Merlin Dorfman (Editor), Richard H. Thayer (Editor) ,Software Engineering
8. Robert C. "Uncle Bob" Martin , Clean Architecture: A Craftsman's Guide to Software Structure and Design