6.3 Course Name: Certificate Course in Software Analysis

Course Objective

The objective of this course is to provide the student with the detailed knowledge of techniques for the analysis and design of complex software intensive systems. This course also introduces the students with the best methods for QA and the testing of the software.

Course Prerequisite

Education: Any Engineering /Science graduate with mathematics up to 10+2 level 2+ Year of experience in IT Domain

Course Outcome

These candidates will be provided an overview of software Engineering methodology, Project development and Management skills. After completion of this course, they can start career as software Quality Assurance Engineer, Tester and leads to project manager and entrepreneur after having relevant experience

Course Duration: 80 Hrs (4 hours/ day for 4 Weeks)

Course Outline:

S. No.	Course Modules	Duration (Hrs)
1	Introduction and Overview	3
2	Life Cycle Models	5
3	Integrated Development Environment	2
4	Version Control Systems	4
5	Requirements Engineering	3
6	Software Requirements Analysis and Specification (Technical Documentation)	10
7	OO Software and UML	4
8	Software Architecture	3
9	Black-Box Testing	3
10	White-Box Testing	2
11	Agile Development Methods	6
12	Mastering Agile Scrum Project Management	14
13	Agile Scrum Management using Jira Tool	4
14	Software Refactoring	2
15	DevOps intro	2
16	Case Study	3
17	Project Work	10
	Total	**80**

1. Introduction and Overview

- Importance of Software Engineering
- Discipline of Software Engineering

- The Software Crisis
- Software Phases

2. **Life Cycle Models**
    - Requirements Engineering
    - Design
    - Maintenance
    - Software Process Model Introduction
    - Waterfall Process
    - Spiral Process
    - Evolutionary Prototyping Process
    - Rational Unified Process
    - Agile Process
    - Choosing a Model
    - Lifecycle Documents
3. **Integrated Development Environment**
    - Eclipse Introduction
    - IDE Overview
    - Plug-Ins
    - Eclipse Demo: Create Java Project
    - Eclipse Demo: Create a Class
    - Eclipse Demo: Run Configuration
    - Eclipse Demo: Debugging
4. **Version Control Systems**
    - Version Control System Introduction
    - Two Main Types of VCS
    - Introduction to Git
    - Git Workflow
    - Git Demo: Intro to Git
    - Git Demo: Git + Eclipse
    - Git Demo: Github
    - Git Recap: Local Repositories
    - Git Recap: Remote Repositories
5. **Requirements Engineering**
    - General RE Definition
    - Software Intensive Systems
    - Functional and Non-functional Requirements
    - User and System Requirements

- Modeling Requirements
- Analyzing Requirements
- Requirements Prioritization
- Requirements Engineering Process

6. **Software Requirements Analysis and Specification (Technical Documentation)**
   - Requirements Gathering and Analysis
   - Why Spend Time and Resource to Develop an SRS Document
   - Value of a Good SRS
   - Requirement Process
   - Requirements Specification
     - Desirable Characteristics of an SRS
     - Components of an SRS.
     - Structure of a Requirements Document
   - Functional & Non-Functional Specification with Use Cases.
     - Basics.
     - Examples
     - Extensions
   - Developing Use Cases
- Other Approaches for Analysis
     - Data Flow Diagrams
     - Entity Relationship Diagrams
   - How to Document the Requirements & Organised.
   - Validation.
   - UX Design
   - Small Case Study

7. **OO Software and UML**
   - Object Orientation Introduction
   - UML Structural Diagrams: Class Diagrams
   - Class Diagram: Creation Tips
   - UML Structural Diagrams: Component Diagram
   - UML Structural Diagram: Deployment Diagram
   - UML Behavioural Diagram: Use Case
   - Use Case Diagram: Creation Tips
   - UML Behavioural Diagrams: Sequence
   - UML Behavioural Diagrams: State Transition Diagram

8. **Software Architecture**
   - What is Software Architecture?

- Prescriptive vs. Descriptive Architecture
- Architectural Evolution
- Architectural Elements
- Components, Connectors, and Configuration
- Deployment Architectural Perspective

9. Black-Box Testing

- Intro to Selenium (use Eclipse IDE)*
- Systematic Functional Testing Approach
- Test Data Selection
- Category Partition Method
- Produce and Evaluate Test Case Specifications
- Generate Test Cases from Test Case Specifications
- Model Based Testing
- Finite State Machines

10. White-Box Testing

- Coverage Criteria Intro
- Statement Coverage
- Control Flow Graphs
- Test Criteria Subsumption

11. Agile Development Methods

- Cost of Change
- Agile Software Development Process
- Extreme Programming (XP)
- XP’s Values and Principles
- Test First Development
- Refactoring
- Pair Programming
- Continuous Integration
- Testing Strategy
- High Level Scrum Process

12. Agile Scrum Management using Jira Tool

- Jira Features
- Jira Installation
- Project Implementation in Jira
- Create and configure Scrum board
- System Dashboard, project navigation
- Issue Types

- Backlog, Sprint, Sprint Planning
- Sprint Tracking
- Defect Reporting and tracking in JIRA
- Sprint Report-Burndown Chart
- Sprint Report-Velocity

13. Mastering Agile Scrum Project Management

- Traditional vs. Agile Project Management
- Agile Manifesto Principles
- SCRUM and XP Methods
- Other Agile Methods
- Value Driven Development
- Prioritization in Agile Project Management
- Planning Agile Projects
- Estimation
- Managing Projects
- Adaptive Planning and Design
- Soft Skills and Leadership
- Team Formation and Boosting Team Performance
- Stakeholder Engagement
- Communication in Projects
- Problem Detection, Metrics and Resolution
- Quality and Earned Value Management
- Continual Improvement

14. Software Refactoring

- Reasons to Refactor
- Refactoring Demo
- Refactoring Risks

15. Introduction of DevOps

16. Case Study:

17. Project

===== END OF PROPOSAL ==========================