

## **B001 : Introduction to Software Development Methodologies**

### **Overview:**

Software development is the set of activities and processes that will eventually result in a software product. This may include inventing, improving, selecting among alternative solutions, and then describing computer programs that meet user requirements within the constraints of the environment. Adopting a good software development methodology will eventually lead to a software product that meets most, if not all of the agreed upon requirements, within the given constraints that, which includes cost, and schedule.

### **Objective:**

The objective of this introductory course is to provide participants with the basics of common software development methodologies with emphasis on concepts and techniques of software development that will form the base for subsequent courses.

### **Topics:**

- Introduction to Systems Engineering.
- Introduction to Information Engineering.
- Basics of Software Engineering.
- Software Process Models
- The Software Development Life Cycle
- Basics of Requirements Engineering
- Basics of Software Design
- Software Development Tools
- Automating Software Development through CASE Tools
- Managing Software Development Projects

### **Duration:**

3 Days

### **Pre-Requisites:**

None

## **B002: Software Requirements**

- **Objectives:**

The main objective of this course is to provide trainees with the basics of software requirements engineering including requirements elicitation techniques, analysis models and methodologies, tools and management issues.

- **Outcomes:**

Upon successful completion of this course, trainees will be able to:

- Develop & document software requirements at different levels following standard techniques.
- Analyze requirements to identify quality issues
- Develop software analysis models following state of the art technologies (using UML standard notations)
- Understand issues related to software requirements management
- Develop selection criteria for Software Requirements Support Tools

- **Contents:**

- Introduction
- Characteristics of Good Requirements
- Developing and Documenting Software Requirements
- Use Cases for documenting software requirements
- issues in requirements management
- Software requirements tools.

- **Duration:**

2 Days

## B003 : Process Modeling

- Objectives :

The goal for this course is to give the trainee a clear picture on function-based process modeling. This includes a discussion of what process modeling is, the goals of process modeling, and a comparison of the different methods used for model building with emphasis on graphical (visual) representation. Detailed information on how to collect requirements and to construct appropriate models should be covered. Case studies that illustrate examples may be presented and examples solved as exercises using graphical notation.

- Outcomes :

The participants will receive detailed training on modeling a business process. They will learn not only the technical aspects of the covered techniques, but also the methodology they should follow in using them.

Business analysts, designers or modelers will gain a set of core techniques that they can apply regardless of the tool or notation chosen.

The course lays the foundation for building business process models that are understandable and shareable so that one can be confident that he/she will make the right modeling choices. When reviewing someone else's models, it will help them understand and ask the right questions.

- Contents :

- Introduction to Process Modeling
- Business Process Management (BPM)
- Requirements Collection for Process Modeling
- Business Modeling Languages (e.g. UML, BPEL, BPMN, WS-CDL)
- Defining Process Models
- Business Process Modeling Techniques
- Data Flow Diagrams
- Work Flows
- UML Activity Diagrams
- Measuring and Analyzing Processes
- Case Studies

- Duration :

3 Days

## B004 : Data Modeling

- Objectives :

As it is becoming more essential for every IT professional to have a basic understanding of data modeling, this course aims to provide an opportunity for the trainees to gain fundamental data modeling skills that all developers are expected to possess. By getting exposed to a number of examples and involved in the creation of data models, through a series of exercises, the trainees will be able to read an existing data model, create a new data model, and appreciate fundamental database design techniques.

- Outcomes :

The trainees will gain an understanding of relational database concepts and database implementation issues. Upon completion of the course, the trainees are expected to be well acquainted with entity relationship (ER) data modeling and data normalization, the creation and management of conceptual data models, generation of physical data models, databases and implementation details.

- Contents :

- Introduction and Basic Concept
- Overview of Data Modeling Using The Entity Relationship Diagram
- Relationships
- Relationship Degrees
- Complex Entities
- ERD Design Guidelines and Quality Checks
- Object Modeling and Class Diagrams
- Mapping Entity Relationship Diagram Into Tables
- Data Normalization
- Designing Databases

- Duration :

3 Days

## **B005 : Software Design I**

- **Objectives:**

The main objective of this course is to provide trainees with the basics of software architecture including Architectural models, Quality attributes, tools and new technologies.

- **Outcomes:**

Upon successful completion of this course, trainees will be able to:

- Develop & document software architecture following standard techniques.
- Critique software architecture based on industry de facto quality attributes
- Develop software following state of the art architectural models; e.g.
  - Service Oriented Architecture
  - Software Product Lines
  - Aspect Oriented Architecture
- Understand new trends in software architecture development
  - Model-Driven Architecture
  - Middleware for distributed systems
- Develop selection criteria for Software Design Automated Tools

- **Contents:**

- Introduction
- Characteristics of Good Design
- Developing and Documenting Software Architecture
- Software Product Lines
- Aspect Oriented Architecture
- Model Driven Architecture
- Service Oriented Architecture and Technologies
- Middleware for Distributed Systems
- Automated Tools for Software Design

- **Duration:**

3 Days

## **B006 : Software Design II**

- **Objectives:**

The main objective of this course is to provide trainees with the basics of software components design including key technologies such as UML and design patterns. The course also aims giving hands on experience on tools supporting software design and migration from design to code.

- **Outcomes:**

Upon successful completion of this course, trainees will be able to:

- Develop & document software detailed design using UML 2.0.
- Critique software designs based on industry de facto standards.
- Develop software detailed designs using design patterns
- Understand how to move from architecture to design following standard methods.
- Understand hot to move from design to code following standard methods.
- Develop selection criteria for software design automated tools

- **Contents:**

- Introduction
- UML 2.0
- Software design tactics
- Design patterns
- Specifying component interfaces
- Optimizing object model to address performance requirements
- Realization of class model components
- Mapping class model to a storage schema.

- **Duration:**

3 Days

## **B007 : Software Quality Assurance and Testing**

### **Overview:**

Software Quality is the successful satisfaction of explicitly stated functional and performance requirements, explicitly documented development standards, and implicit characteristics that are expected of all professionally developed software. Software testing on the other hand is the process and activities associated with confirming the proper operation of a software system according to the agreed upon requirements.

### **Objective:**

The main objectives of this course are to provide participants with the basic concepts, techniques, tools, and methodologies for software Quality assurance and testing.

### **Topics:**

- Introduction to Software Quality
- Software Dependability
- Software Quality Assurance and Testing
- Software Quality Metrics
- Standards in Software Quality
- Software Quality Assurance Plan and Activities
- Software Reviews
- Verification and Validation
- Software Testing
- Test Documentation (Plan, Procedure, etc.)
- Checklists and Operational Profiles
- Different Types of Tests
- Introduction to Models for Software Testing
- Beyond Testing: Process Improvement

### **Duration:**

3 Days

### **Pre-Requisites:**

Introduction to Software Development Methodologies.