

## “Software Engineering Package” Training Program (English Material)

**Package Length:** 92 Hours @ 23 days

### Package Overview

Software development requires the collaborative effort of a diverse team with varied skills. To be most effective, team members need to understand the activities performed at each stage in the development cycle. In this PTC package, the attendees will be able to analyze, design, implement and test applications that meet the customer (internal / external) requirements through a simulated case study. Also, attendees gain the management techniques to handle and control the software development project. Moreover, gain the understanding of business process re-engineering techniques with simulated case studies.

### Package Content

#	Training Program Name	Hrs	Days
1	Introduction to Software Development	8	2
2	System requirements	8	2
3	System Analysis and Design	12	3
4	Relational Database – SQL	12	3
5	Software Testing (Quality Assurance)	8	2
6	Programming	12	3
7	Project Management for IT specialists	12	3
8	Ms-project – Project Management tool (Optional)	8	2
9	Business process re-engineering	8	2
10	Introduction to CMMI	4	1

### Learn what?

- **Introduction to Software Engineering**
  - Analyze, design, program and test software projects
  - Draw system requirements and write user stories (case diagrams )
  - Draft mock-up user interfaces
  - Program using basic language concepts and syntax
  - Simplify complex systems using modern object-oriented analysis and design techniques
  - Test user requirements and application logic with both manual and automated testing techniques

- **System Requirement**
  - Develop software requirements for software-intensive systems using proven methodologies
  - Apply use cases to software development initiatives
  - Enhance and refine use cases using an iterative approach
  - Model user interface using mock-ups and rapid prototypes
  - Develop a requirements data model
  - Validate requirements and generate use case tests
  
- **System Analysis and Design**
  - Analyze user requirements and design robust
  - Select the right software architecture for your evolving business needs
  - Design a robust core of stored information for new or existing legacy requirements
  - Control complex behavior for effective decision making and user interaction
  - Adopt a development process that ensures robust database and Web-enabled systems
  
- **Relational Database - SQL**
  - Design, build and query a relational database
  - Develop a data model to describe an application's data
  - Apply normalization to data for effective, stable database design
  - Build a relational database from the logical database design
  - Access data in a relational database using simple SQL queries
  
- **Software Testing (Quality Assurance)**
  - Apply general software testing principles and fundamental test processes
  - Implement test levels and types to various software development models
  - Conduct static techniques using proper roles, responsibilities and tools
  - Perform specification- and structure-based test design techniques
  - Manage tests including planning, estimating, monitoring and controlling
  
- **Programming**
  - Deliver adaptable software iterations based on Agile methodologies
  - Minimize bugs and maximize productivity with Test-Driven Development and unit testing
  - Enhancing existing code for easier maintenance and improved design

- Achieve quality design by adopting established coding principles
- Simplify complex coding problems with design patterns
  
- **Project Management for IT specialists**
  - Scope management – define the project and its main activities
  - Time management – scheduling, PDM and PERT
  - Cost management - optimizing and controlling
  - Resource management – leveling technique
  - Communication management - team roles and project manager duties
  - Risk Management – define risk and its actions
  
- **Ms-project – Project Management tool**
  - Create a project with activities and resources
  - Calculate a schedule
  - Analyze resource/cost data in projects
  - Determine resource allocation through leveling
  - Monitoring and controlling the project life
  
- **Business Process Re-engineering (BPR)**
  - BPR Definition and its driving forces
  - BPR characteristics, process definition and QDF approach
  - BPR and TQM
  - Objectives of applying BPR
  - BPR Assessment Scheme
  - Building a BPR model , tool
  - Case Study
  
- **CMMI**
  - Apply the Capability Maturity Model Integration (CMMI) methodology for process improvement
  - Select the CMMI implementation that best meets the needs of your organization
  - Perform key appraisal activities related to the CMMI
  - Prepare a Summary Report that includes recommendations for action
  - Develop a practical and effective implementation schedule
  - Apply CMMI principles within your organization