



Arab Republic of Egypt  
**Ministry of Communications  
and Information Technology**



**Information Technology  
Industry Development Agency**

# Course Catalog 2008-2010



**Software Engineering Competence Center**

---

# Introduction

Software Engineering Competence Center (SECC) mission is to support the development of the software industry in Egypt by improving the software Engineering practices to higher maturity levels and achieve strong global market presence.

**Providing efficient and quality training is an essential tool to achieve SECC goals, where the main objectives of training are the following:**

- Train software engineers to appreciate applying software standards as a part of the system life cycle
- Provide training in metrics , collection and analysis
- Provide training on strategic planning & business models
- Provide specific courses i.e. Personal Software Processes (PSP) and Team Software Processes (TSP)
- Provide training on specific models i.e. SEI **Capability Maturity Model Integration (CMMI)**
- Provide training on software technology transfer and offshore development
- Influence academic curriculums in computer science and software Engineering to meet the future trends

**This document describes SECC training courses that cover the following tracks:**

- A. Executive and Top Management
- B. Software Development Methodologies
- C. Object Oriented Software Development
- D. Software Quality Engineering
- E. Systems Management Improvement
- F. Software Process Improvement
- G. SPI for SME's Training
- H. Software Testing
- I. SEI Official Training
- J. Rational Training
- K. Embedded Systems
- L. Soft Skills
- M. Collective Track

**The target audience includes, but not limited to, the following:**

- Executives and Top Management
- Software Engineering Process Group Members
- Personnel Responsible for an Organization's Software Process Improvement Program
- Project Managers and Middle Level Management
- Software Engineers
- Software Engineering Educators
- Software Quality Assurance Specialists

## SECC Training Tracks, Courses, Duration

### A. Executive And Top Management Track

Code	Course Name	Duration (Days)
A001	Executive Briefing	1
A002	Overview of the Capability and Maturity Model Integration	1
A003	Software Process Improvement overview	1

### B. Software Development Methodologies Track

Code	Course Name	Duration (Days)
B001	Introduction to Software Development Methodologies	3
B002	Requirement Engineering	3
B003	Process Modeling: Concepts and Tools	3
B004	Data Modeling: Concepts and Tools	3
B005	Software Design (I)	3
B006	Software Design (II)	3
B007	Software Quality Assurance and Testing	3

### C. Object Oriented Software Development

Code	Course Name	Duration (Days)
C001	Introduction to Object Orientation	2
C002	UML ( Unified Modeling Language )	3
C003	O.O. Analysis & Design	4
C004	O.O. Implementation	3

### D. Software Quality Engineering Track

Code	Course Name	Duration (Days)
D001	Introduction to Software Quality Engineering	2
D002	Software Quality Management	2
D003	Software Processes	2
D004	Software Project Management	2
D005	Software Metrics and Measurements	2
D006	Software Inspection and Testing	2
D007	Software Audit	2

## SECC Training Tracks, Courses, Duration

### E. Systems Management Improvement Track

Code	Course Name	Duration (Days)
E001	Project Management Techniques and Tools	2
E002	Total Quality Management	2
E003	Business Process Re-engineering	2

### F. Software Process Improvement Track

Code	Course Name	Duration (Days)
F001	Software Sizing and Estimation	3
F002	Managing Software Projects with Metrics	3
F003	Risk Management	2
F004	Software Configuration Management	1
F005	Decision Analysis & Resolution	1
F006	Measurement & Analysis	1
F007	Organizational Transition using CMMI	3
F008	Product & Process Quality Assurance	1
F009	Project Planning , Monitoring and Control	1
F010	Essential Requirements Management	2
F011	Supplier Agreement Management	1
F012	Technical Solution and Product Integration	1
F013	Verification & Validation	2

### G. SPI for SME's Training Track

Code	Course Name	Duration (Days)
G001	Project Management	3
G002	Product Development I: Requirements Management & Development	2
G003	Product Development II: Design & Implementation	3
G004	Configuration Management	2
G005	Peer Review	2
G006	Quality Assurance	2
G007	Software Internal Auditor Training	1
G008	Software Metrics	1
G009	Software Estimation	2

## SECC Training Tracks, Courses, Duration

### H. Software Testing Track

Code	Course Name	Duration (Days)
H001	Software Test Planning and Design	2
H002	Writing Testable Requirements	2
H003	Software Inspections and Reviews	2
H004	Managing Testing Process	2
H005	Principles Architectures and Strategies of Test Automation	2

### I. SEI Training Track

Code	Course Name	Duration (Days)
I1	CMMI Based Courses	
I101	Introduction to CMMI	3
I102	CMMI Executive Workshop	2
I103	Leading and Managing CMMI Based Process Improvement	2
I104	CMMI Based Implementation Essentials	2
I105	Intermediate Concepts of CMMI	5
I106	Mastering Process Improvement	5
I107	Defining Software Process	2
I108	High Maturity with Statistics Outlines	2
I2	TSP/PSP Training Courses	
I201	Introduction to PSP	2
I202	Leading a Development Team	3
I203	PSP for Engineers: Planning & Quality	10
I204	TSP Executive Strategy	2

### J. Rational Training Track

Code	Course Name	Duration (Days)
J001	Essentials of Rational Requisite Pro	1
J002	Essentials of Rational ClearQuest	1
J003	Essentials of Clear Case	2
J004	Essentials of Rational ClearCase UCM for Windows	1
J005	Essentials of Configuration Management with Rational ClearCase	2
J006	Essentials of Rational Project Console	1
J007	Essentials of Rational SoDA for Microsoft Word	1
J008	Essentials of Test Management with Rational Testing Manager	1

## SECC Training Tracks, Courses, Duration

### K. Embedded Systems Track

Code	Course Name	Duration (Days)
K1	Embedded Linux Track	3
K2	FPGA Track	10
K3	ARM Track	4
K4	Embedded Multimedia Track	4
K5	RTOS Track	4
K6	Embedded Support Track	2

### L. Soft Skills Track

Code	Course Name	Duration (Days)
L001	Presentation Skills	3
L002	Communication Skills	2
L003	Business Writing Essentials	2
L004	Technical Writing Skills	2
L005	Negotiation Skills	3
L006	Marketing Essentials	2
L007	Essentials of Sales	2
L008	Sales Management Course	2
L009	Train the Trainer	3

### M. Collective Track

Code	Course Name	Duration (Days)
<b>B</b>	<b>Software Development Methodologies Track</b>	
B001	Introduction to Software Development Methodologies	3
B002	Process Modeling: Concepts and Tools	3
B003	Data Modeling: Concepts and Tools	3
B004	Software Design (I)	3
B006	Software Quality Assurance and Testing	3
<b>C</b>	<b>Object Oriented Software Development</b>	
C001	Introduction to Object Orientation	2
C002	UML ( Unified Modeling Language )	3
C003	O.O. Analysis & Design	4
C004	O.O. Implementation	3
<b>L</b>	<b>Soft Skills Track</b>	
L001	Presentation Skills	3
L002	Communication Skills	2
L003	Business Writing Essentials	2
L004	Technical Writing Skills	3
L005	Negotiation Skills	3
L006	Marketing Essentials	2
L007	Essentials of Sales	2

## **A. Executive and Top Management**

*This track is designed to offer a general overview for software leaders who want to understand concepts and approaches related to software process improvement and Capability Maturity Model Integration(CMMI).*

## Executive and Top Management

---

<b>A001</b> Executive Briefing	10
<b>A002</b> Overview of the Capability and Maturity Model Integration	11
<b>A003</b> Software Process Improvement overview	12

## Executive Briefing

### *Overview*

The course is intended to offer answers about the benefits of technology adoption or improvement. It gives information about efforts needed to support software acquisition, development and management.

### *Topics to be covered*

- Technology transition: a frame work to approach technology alternative, select candidate solutions, apply models and heuristics to successful implementation
- Process improvement: a frame work of SEI/IDEAL (Initiating, Diagnosis, Establishing, Acting, Leveraging) approach
- Capability Maturity Model Integration (CMMI): describe how the organization can use CMM to define, implement, measure, control and improve its software processes
- Software Capability Evaluation (SCE): explains the relationship of SCE method to CMMI. It also explains the relationship between the site visit process and the use of SCE
- CMMI Based Appraisal for Internal Process improvement (CBA-IPI): explains SEI method for software organizations to self assess their maturity

### *Target Audience*

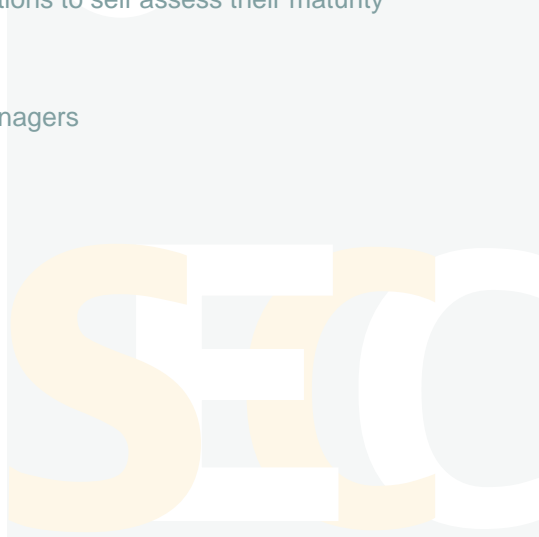
Top level and executive managers

### *Duration*

1 day

### *Pre-Requisites*

None



## Overview of the Capability and Maturity Model Integration (CMMI)

### *Overview*

This course is designed to offer general overview of CMMI concepts. Each maturity level contains key practices which the organization can use to define, implement, measure, control and improve its business processes. These capability models are the underlying foundation for several appraisals and process improvement.

### *Topics to be covered*

- Overview of Software Capability Maturity Model Integration (CMMI) : Describe the model for practices. How an organization can use it to define, implement, measure, control and improve its software processes
- Awareness of The Components of The CMMI Models and how They May be Applied
- Awareness of behavioral Characteristics Associated with each Maturity level
- Reinforcement of key concepts about model structure and function, and important model characteristics

### *Target Audience*

Top level and executive managers.

### *Duration*

1 day

### *Pre-Requisites*

None



## Software Process Improvement (SPI) Overview

### *Overview*

It is an overview for software leaders who want to understand the rationale , language and approach for Capability Maturity Model Integration - based software process improvement.

Through a combination of lectures and discussions participants learn what software process improvement (SPI) can give them. They can gain an appreciation for the business value.

### *Topics to be covered*

- Background of software process improvement
- Benefits of software process improvement
- Overview of CMMI
- Overview of Initiating, Diagnosis, Establishing, Acting Learning (IDEAL) model such as appraisals, software process definition, metrics, etc. IDEAL serves as a roadmap for initiating, planning and implementing improvement activities. It is the founding strategy employed in developing many SEI products and services and is used to effectively integrate them into a comprehensive method managing and improving software capability

### *Target Audience*

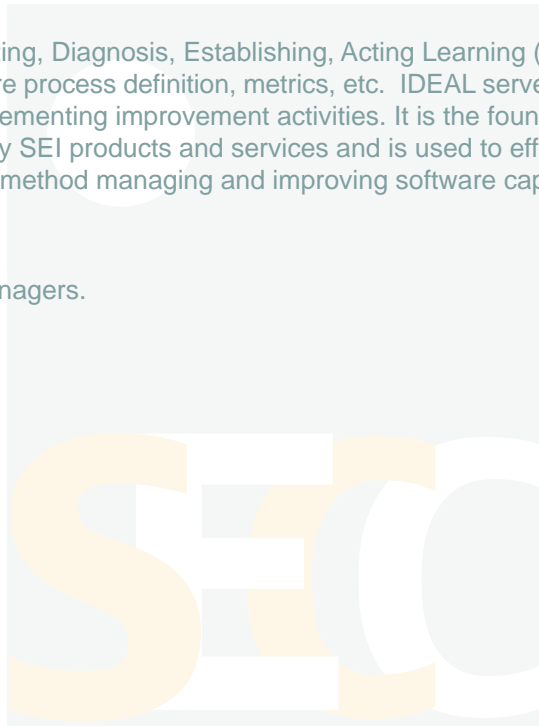
Top level and executive managers.

### *Duration*

1 day

### *Pre-Requisites*

None





# Table of Content

Introduction .....	2
SECC Training Tracks, Courses, Duration .....	4
<b>A.</b> Executive and Top Management Track .....	8
<b>B.</b> Software Development Methodologies Track .....	14
<b>C.</b> Object Oriented Software Development Track .....	24
<b>D.</b> Software Quality Engineering Track .....	30
<b>E.</b> Systems Management Improvement Track .....	40
<b>F.</b> Software Process Improvement Track .....	46
<b>G.</b> SPI for SME's Training Track .....	62
<b>H.</b> Software Testing Track .....	74
<b>I.</b> SEI Training Track .....	82
<b>J.</b> Rational Training Track .....	98
<b>K.</b> Embedded Systems Track .....	108
<b>L.</b> Soft Skills Track .....	136
<b>M.</b> Collective Track .....	148



## **B. Software Development Methodologies**

*This track is designed to offer participants the concepts and details of software development methodologies with emphasis on process modeling, data modeling, software design, and quality assurance & testing.*

## Software Development Methodologies

---

<b>B001</b>	Introduction to Software Development Methodologies	16
<b>B002</b>	Requirement Engineering	17
<b>B003</b>	Process Modeling: Concepts and Tools	18
<b>B004</b>	Data Modeling: Concepts and Tools	19
<b>B005</b>	Software Design (I)	20
<b>B006</b>	Software Design (II)	21
<b>B007</b>	Software Quality Assurance and Testing	22

## Introduction to Software Development Methodologies

### *Overview*

Software development is a process of inventing, improving and selecting among alternative solutions, and then describing computer programs that meet users requirements within the constraints of the environment and based on relevant criteria. A good software design is one that describes a system that meets all its requirements. The best software design is one that describes a system that meets and most appropriately balances the requirements, constraints and criteria. The purpose of this introductory course is to provide participants with the basics of software development methodologies with emphasis on concepts and techniques of software development forming base for subsequent courses.

### *Topics to be covered*

- Software development environment
- Basics of software engineering
- Introduction to information engineering
- Software development life cycle
- Basic software design issues
- Software development tools
- Automating software development through CASE tools
- Managing software development projects
- Exercises

### *Target Audience*

Software development / design engineers

### *Duration*

3 days

### *Pre-Requisite*

None

## Requirement Engineering

### *Overview*

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.

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 and develop selection criteria for Software Requirements Support Tools.

### *Topics to be covered*

- Characteristics of Good Requirements
- Developing and Documenting Software Requirements
- Use Cases for documenting software requirements
- Issues in requirements management
- Software requirements tools
- Exercises

### *Target Audience*

Software development / Design engineers.

### *Duration*

3 days

### *Pre-requisite*

Introduction to Software Development Methodologies.

## Process Modeling: Concepts and Tools

### *Overview*

Process modeling involves graphically representing the functions, or processes, which capture, manipulate, store, and distribute data between a system and its environment and between components within a system.

The course lays the foundation for building business process models that are understandable and shareable. When reviewing someone else's models, it will help them understand and ask the right questions.

### *Topics to be covered*

- 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

### *Target Audience*

Software development / Design engineers

### *Duration*

3 days

### *Pre-requisite*

Introduction to Software Development Methodologies.

## Data Modeling: Concepts and Tools

### *Overview*

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.

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.

### *Topics to be covered*

- Overview of Data Modeling
- 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
- Case studies

### *Target Audience*

Software development / design engineers

### *Duration*

3 days

### *Pre-requisite*

Introduction to software development methodologies

## Software Design (I)

### *Overview*

The purpose of design is to create a solution that best meets the criteria of the software. The purpose of a method is to help the designer achieve a good design. A complete software design method includes criteria for choosing between various designs alternatives. These criteria help designers to meet software design objectives.

This course focuses on the logical design aspects of software design with emphasis on designing forms, reports, interfaces and logical databases.

### *Topics to be covered*

- 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
- Exercises

### *Target Audience*

Software development / design engineers.

### *Duration*

3 days

### *Pre-Requisite*

Introduction to Software Development methodologies

## Software Design (II)

### *Overview*

This course is a continuation to the course entitled "**B005 : software Design (I)**". This course focuses on the physical design aspects of software development with emphasis on designing: physical files and databases; system and program structures and distributed processing strategies.

### *Topics to be covered*

- Basic concepts of software physical design
- Physical design deliverables and outcomes
- Designing physical files and databases
- Designing the internals
- Program and process design
- Designing distributed systems
- Designing systems for client/server architecture
- Designing systems for Multi-tiers architecture
- Managing data in distributed systems
- Web enabled design

### *Target Audience*

Software development / design engineers.

### *Duration*

3 days

### *Pre-Requisite*

Software Design (I)



## 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. 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 to be covered

- 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
- Exercises

### Target Audience

Software development / design engineers

### Duration

3 days

### Pre-Requisites

Introduction to Software Development Methodologies



Acronyms and Abbreviations

- All acronyms used in this document and throughout the project must be defined in the first instance in which they are used in the document.
- Abbreviations should be used only when necessary.
- The use of acronyms and abbreviations should be avoided where possible.

Software Engineering & Compliance Center

100

## **C. Object Oriented Software Development**

*The main focus of this track is to provide participants with fundamentals and details of object oriented paradigm, including Unified Modeling Language (UML), object oriented analysis, design and implementation.*

## Object Oriented Software Development

---

<b>C001</b>	Introduction to Object Orientation	26
<b>C002</b>	UML (Unified Modeling Language)	27
<b>C003</b>	Object Oriented Analysis & Design	28
<b>C004</b>	Object Oriented Implementation	29

## Introduction to Object Orientation

### *Overview*

The aim of this course is to provide participants with basics and fundamental concepts of Object Oriented paradigm.

### *Topics to be covered*

- The Main Building Blocks of object-oriented modeling
- Principles of object-oriented design
- OO concepts
- Data models
- Levels of abstraction
- Class diagrams
- Relationship modeling
- Class
- Specifying classes in a typical object-oriented programming language (C++)
- Mapping abstract data types to classes
- Encapsulation
- Information hiding
- Message passing
- Inheritance
- Polymorphism

### *Target Audience*

Systems and/or software Engineering practitioners and developers.

### *Duration*

2 days

### *Pre-Requisites*

None

## UML (Unified Modeling Language)

### *Overview*

Unified Modeling Language (UML) is an emerging standard. UML is designed to facilitate capturing of the static structure of a system as well as its dynamic behavior. As is the case with graphical modeling languages and tools, modeling a system is based on a collection of discrete objects with their interactions to represent the system under study from different perspectives. UML is a nonproprietary, discrete, universal and general-purpose modeling language.

### *Topics to be covered*

- An introduction to object oriented approach
- The concepts of models, modeling, and modeling languages
- The Unified Modeling Language including its elements and diagrams
- Class diagram
- Object diagram
- Use case diagram
- Sequence diagram
- Collaboration diagram
- State chart diagram
- Activity diagram
- Component diagram
- Deployment diagram
- A hands-on experience for designing UML models based on one of the UML development tools

### *Target Audience*

Systems and/or software Engineering practitioners and developers.

### *Duration*

3 days

### *Pre-Requisites*

Introduction to Object Orientation

## Object Oriented Analysis & Design

### *Overview*

This course presents the concepts and techniques of object-oriented analysis and design using UML. The course views the software development process, an iterative software lifecycle, which uses UML throughout to capture and communicate analysis and design information.

### *Topics to be covered*

- Object oriented software Engineering
- OO process and project
- OO analysis Vs OO design
- Requirements Engineering
- Requirements elicitation
- Requirements analysis
- Analysis modeling by example
- Requirements specifications case Studies
- Software architecture
- Object oriented design
- Design case studies
- OO design patterns

### *Target Audience*

Systems and/or software Engineering practitioners and developers.

### *Duration*

4 days

### *Pre-Requisites*

UML (Unified Modeling Language)

## Object Oriented Implementation

### *Overview*

This course presents the techniques and practices of transforming object-oriented design artifacts into code. The course is designed to attendees who are familiar with at least one Object-Oriented programming language who wish to gain experience/ familiarization in coding of an object oriented design.

The main focus of this course is to teach participants techniques and different options to mapping an object-oriented design, developed using UML, to an implementation. Participants will gain knowledge and practical hands-on to implement basic classes, static behavior, dynamic behavior, and complex relationships (for example inheritance, aggregation and other complex relationships)

### *Topics to be covered*

- Implementing classes
- Implementing static behavior
- Implementing dynamic behavior
- Instantiating and deleting objects
- Implementing Inheritance
- Implementing aggregation
- Implementing relationships
- Case studies

### *Target Audience*

Systems and/or software Engineering practitioners and developers.

### *Duration*

3 days

### *Pre-Requisites*

Object Oriented Analysis & Design

## D. Software Quality Engineering

*The main objective of this track is to provide participants with the basic concepts, techniques and models for software quality Engineering with emphasis on software quality management, software project management, software metrics and measurements and software inspection, testing and auditing.*

## Software Quality Engineering

---

<b>D001</b>	Introduction to Software Quality Engineering	32
<b>D002</b>	Software Quality Management	33
<b>D003</b>	Software Processes	34
<b>D004</b>	Software Project Management	35
<b>D005</b>	Software Metrics and Measurements	36
<b>D006</b>	Software Inspection and Testing	37
<b>D007</b>	Software Audits	38

## Introduction to Software Quality Engineering

### *Overview*

The main objectives of this course are to provide participants with the basics of software quality Engineering with emphasis on software quality standards, specifications, models and techniques.

### *Topics to be covered*

- Basic concepts of software quality Engineering
- Software quality profile
- Software standards and specifications
- Software total quality management
- Concepts of prevention and detection
- Organizational and interpersonal techniques
- Software quality models
- Risk management

### *Target Audience*

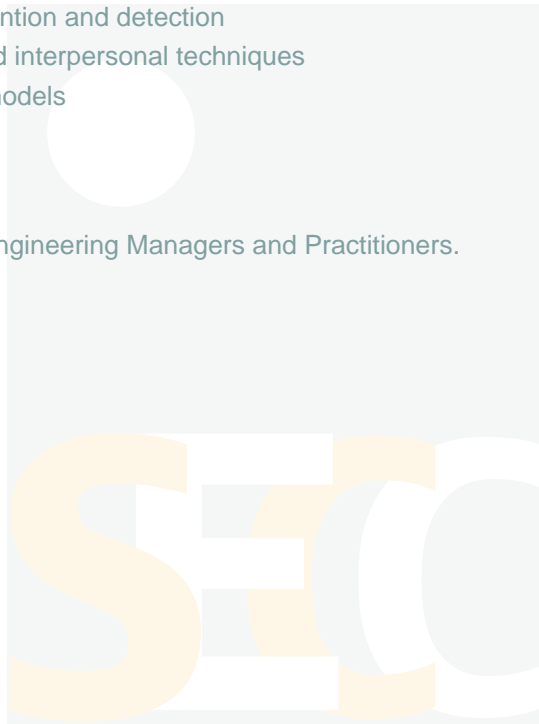
Systems and/or software Engineering Managers and Practitioners.

### *Duration*

2 days

### *Pre-Requisites*

None



## Software Quality Management

### *Overview*

The main objectives of this course are to provide participants with the basics concepts, techniques and models for software quality management.

### *Topics to be covered*

- Basic of software quality management
- Customer requirement and satisfaction for quality
- Customer oriented lifecycle techniques
- People management and development process
- Competence model and role classification

### *Target Audience*

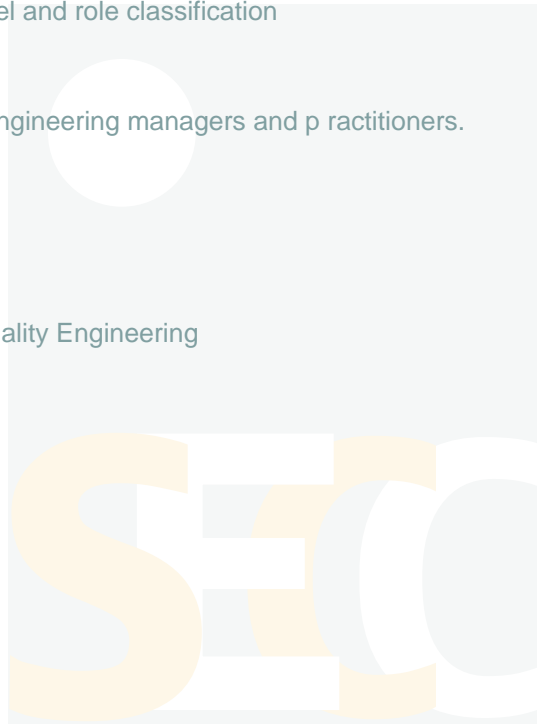
Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering



## Software Processes

### *Overview*

This course provides participants with the basics of software processes with emphasis on modeling, procedures, assessment and environmental aspects.

### *Topics to be covered*

- Software development procedures
- Defect prevention, detection and removal methods
- Software process assessment and evaluation techniques
- Software process modeling
- Software environments
- Quality improvement efforts and quality systems

### *Target Audience*

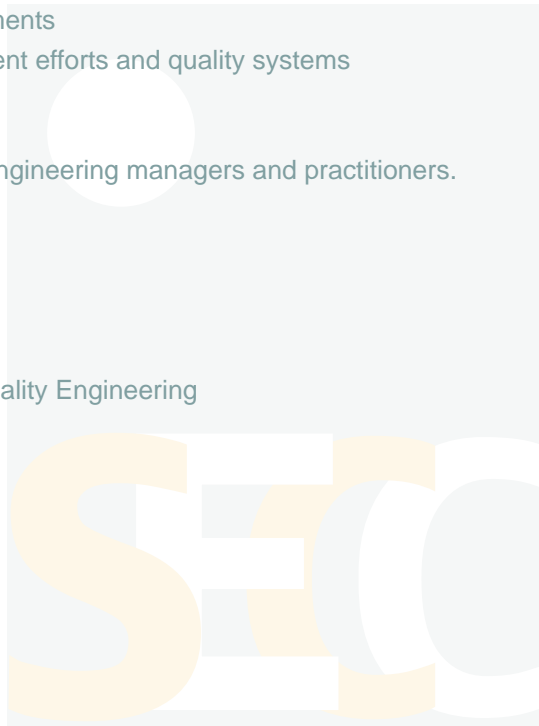
Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering



## Software Project Management

### *Overview*

The main objectives of this course are to provide participants with the basic concepts, procedures, tools and models for software project management.

### *Topics to be covered*

- Concepts of software project management
- Project planning factors
- Project planning methods and tools
- Software size estimation
- Software cost and effort estimation
- Maintenance types
- Project scheduling and tracking
- Tools for project management
- Risk management

### *Target Audience*

Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering

## Software Metrics and Measurements

### *Overview*

This course provides the participants with the basics of software metrics and measurements with emphasis on analytical techniques and tools.

### *Topics to be covered*

- Definition of metrics and measures
- Designing metrics and measures
- Analytical techniques
- Quality analysis tools
- Sampling theory and techniques
- Prediction techniques
- Software quality attributes

### *Target Audience*

Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering

## Software Inspection and Testing

### *Overview*

The main objectives of this course are to provide participants with the basic concepts, techniques and tools for software inspection and testing verification & validation.

### *Topics to be covered*

- Concepts of software inspection and testing
- Inspection types
- Inspection process
- Types of tests
- Test strategies and design
- Test environments and tools
- Test management
- Verification and validation

### *Target Audience*

Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering

## Software Audits

### *Overview*

This course provides participants with the basics of software auditing including audit planning and standards.

### *Topics to be covered*

- Basic aspects of software auditing functionality
- Audit types
- Audit methodology
- Audit process
- Audit planning
- Audit standardization

### *Target Audience*

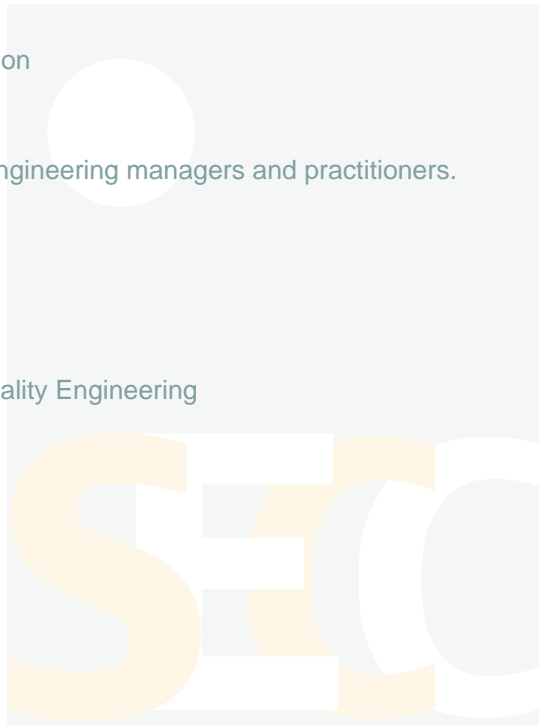
Systems and/or software Engineering managers and practitioners.

### *Duration*

2 days

### *Pre-Requisites*

Introduction to Software Quality Engineering





## **E. Systems Management Improvement**

*This track is designed to develop and improve the skills required to plan and manage the implementation of software projects including total quality management and business process Re-engineering.*

## Systems Management Improvement

---

<b>E001</b> Project Management Techniques and Tools	42
<b>E002</b> Total Quality Management	43
<b>E003</b> Business Process Re-engineering	44

## Project Management Techniques and Tools

### *Overview*

Projects' management are very essential towards fulfilling project objectives. Currently many techniques and tools are available for managing projects. The main objectives of this course are to develop and improve the skills required to plan and manage implementation of projects successfully with emphasis on software development projects, as well as providing a comprehensive and practical set of management tools that can be implemented and used for software project management.

### *Topics to be covered*

- Techniques of projects' management
- Manage software development projects
- Increase productivity and cut costs
- Current available tools for projects management
- Case studies and practical training

### *Target Audience*

Software project managers and other individuals involved in software project planning, monitoring and control.

### *Duration*

2 days

### *Pre-Requisites*

None

## Total Quality Management

### *Overview*

Total Quality Management (TQM) promotes a process of continuous improvements and innovation, where control is embedded within and is driven by the culture of the company as a whole. TQM means the use and application of methods and procedures for continually satisfying customer's requirements at the lowest price. The main objectives of this course are to provide participants with the general framework of TQM with emphasis on different methods and procedures.

### *Topics to be covered*

- Basics of TQM
- TQM framework
- Quality policy deployment
- Quality function deployment
- Quality circles
- Total quality control
- TQM standards

### *Target Audience*

Software project managers and other individuals involved in software project planning, monitoring and control.

### *Duration*

2 days

### *Pre-Requisites*

None



## Business Process Re-engineering

### *Overview*

Business Process Re-Engineering (BPR) is a methodology whereby organizations can take a good look at the way they do business and reevaluate their effectiveness. BPR combines methodologies and techniques from information systems analysis, management, organizational behavior and communication disciplines to analyze business processes in order to determine whether to automate, integrate, or obliterate each process.

BPR is the process that addresses the concerns of whether certain systems or processes should even exist. It extends beyond the scope of software reEngineering in all its flavors, as well as information Engineering.

Various levels of BPR exist from applying technology to an existing business process to redefining an organization's line of business. The main levels are: technology applications, work-flow analysis, business process redesign, business reEngineering.

### *Topics to be covered*

- Basic principles of BPR
- Basic concepts in process modeling
- Modeling large processes
- Analyzing a process Model
- Business reEngineering guidelines
- Techniques to help redesign processes
- Tools to help redesign processes

### *Target Audience*

Software project managers and other individuals involved in software project planning, monitoring and control.

### *Duration*

2 days

### *Pre-Requisites*

None



## F. Software Process Improvement

*This track is designed to provide participants with an in depth understanding of the different techniques and methods commonly used for software process improvement, with emphasis on: sizing and estimation, risk management, configuration management, requirements managements, quality assurance, project planning, product integration and organizational transition.*

## Software Process Improvement

<b>F001</b>	Software Sizing and Estimation	48
<b>F002</b>	Managing Software Projects with Metrics	49
<b>F003</b>	Risk Management	50
<b>F004</b>	Software Configuration Management	51
<b>F005</b>	Decision Analysis & Resolution	52
<b>F006</b>	Measurement & Analysis	53
<b>F007</b>	Organizational Transition using CMMI	54
<b>F008</b>	Product & Process Quality Assurance	55
<b>F009</b>	Project Planning, Monitoring and Control	56
<b>F010</b>	Essential Requirements Management	57
<b>F011</b>	Supplier Agreement Management	58
<b>F012</b>	Technical Solution and Product Integration	59
<b>F013</b>	Verification & Validation	60

## Software Sizing and Estimation

### *Overview*

The course provides an in-depth understanding of methods commonly used in the software industry to estimate software projects. It also provides an overview of the features of sizing and estimation tools that are commonly used in the software industry. Emphasis is given to training and hand-on understanding of estimation methods that may be implemented within organizations processes.

### *Topics to be Covered*

- Software estimation foundations :definitions of terms, benefits, barriers to estimation
- Estimating software Size: lines of Code, function points
- Estimation methods: activity, based costing, Delphi technique
- Common estimation tools: COCOMO II, SLIM, Price-S, SEER-SM, PQM-Plus
- Estimation, selection, evaluation, and Implementation: selection criteria, tool evaluate procedure, Implementing an estimation tool
- Estimation procedures within project process: training needs, estimation checklist(s), validating estimation data, auditing the estimation procedures

### *Target Audience*

Managers, team leaders and senior developers who are responsible for and perform software estimation activities.

### *Duration*

3 days

### *Prerequisite*

None

## Managing Software Projects with Metrics

### *Overview*

The metrics course builds on fundamental project concepts and consists of the training and hand-on experience in the managing of projects through the use of a core metrics set that may be implemented within organization's processes.

### *Topics to be covered*

- Using Goal, question, metric methodology to design a set of metrics
- Designing and implementing a work breakdown structure
- Software size: Lines of code, other physical size measures, function points, Other functional measures
- Product metrics: schedule, cost, effort
- Process metrics: performance, process improvement
- Quality metrics: defect density, reliability, maintainability
- Metrics associated with higher levels of maturity: improving productivity, understanding process capability, analyzing the variance of metric values using statistical techniques, quantifying goals and project performance
- Metrics Collection
- Metrics reporting: process company templates for management and Engineering metrics
- Criteria to evaluate and select metric tools

### *Target Audience*

- Managers
- Team leads
- Senior developers who are responsible for and perform project planning, monitoring, and control activities

### *Duration*

3 days

### *Prerequisite*

None

## Risk Management

### *Overview*

This course provides a survey of the risk management technology and the major approaches that have been developed within the software Engineering field. Some of the major concepts and frameworks for risk management will be explored during the course with exercises used to guide the participants to apply the concepts in their current situation.

### *Topics to be Covered*

- Basic concepts of risk management
- Framework for risk management
- Risk taxonomies
- Risk management within maturity models
- Risk patterns
- Selection of development life cycle models based on risk patterns
- Risk management in Agile methods
- Risk management in project planning
- Contingency planning
- System approach to risk management
- Technologies and tools for risk management
- Strategic risk management at the enterprise level
- Practical approaches to risk management

### *Target Audience*

- Enterprise executives
- Organization managers
- Project leaders
- Key technical leaders

### *Duration*

2 days

### *Prerequisite*

None

## Software Configuration Management

### *Overview*

This course explains the underlying concepts and provides practical knowledge of techniques and tools for establishing and maintaining the integrity of the body of documentation and code and other items that constitute a software product. These configuration management techniques apply equally to the development and support of information systems, commercial software products and embedded applications.

### *Topics to be Covered*

- Introduction and overview
- The identification function: a basis for change
- The control function: managing change
- Version and release management
- Organizing teams
- Preparing and updating a SCM plan
- SCM tools
- SCM in SEI-CMM®
- Conclusion

### *Target Audience*

This course targets a wide range of professionals grappling with complex or evolving systems. This includes project and business managers, quality representatives and technical personnel involved in the production and/or control of business critical information

### *Course Duration*

1 day

### *Prerequisite*

None

## Decision Analysis & Resolution

### *Overview*

The course provides deep insight to decision analysis and resolution with emphasis on decision requirements, evaluation methods, data collection and analysis.

### *Topics to be Covered*

- Introduction
  - Decision requirement
  - Benefits
- CMMI Model
  - Generic practices
  - Specific practices
  - Linkages to PA's
- Scope
  - Process definition
  - Evaluation methods
  - Resource requirements
  - Data collection & analysis
  - Continuous improvement

### *Target Audience*

Systems and / or software Engineering managers and practitioners.

### *Duration*

1 day

### *Prerequisite*

None

## Measurement & Analysis

### *Overview*

This course provides a conceptual and practical framework for designing and implementing a software metrics program. Participants will be able to understand the usage of Metrics in identifying and solving problems, leading to a more effective management of software projects and products. The course also covers concepts of Metrics in CMMI® implementation.

### *Topics to be Covered*

- Concept of measurement
- Managing projects using metrics
- Managing processes using metrics
- Managing products quality and performance using GQM (Goal Question Metric) based metrics program
- Metrics and process models
- Using statistical techniques in metrics analysis
- Setting up metrics program
- Case studies

### *Target Audience*

- Software developers
- Software process engineers
- Software quality practitioners
- Software quality managers

### *Course Duration*

1 day

### *Prerequisite*

None

## Organizational Transition using CMMI

### Overview

The course provides the basics for organizational transition based on CMMI. Emphasis will be given to organizational process focus, process definition, as well as training supply management.

### Topics to be covered

- Transition
  - Moving from project centric to organization centric
  - Benefits
  - Judgmental factors
  - Model perspectives
- CMMI model
  - Generic practices
  - Specific practices
  - OPF & OPD & linkages
  - OT & linkages
  - IPM & linkages
- Definition
  - Process definition
  - Process architecture
  - Tailoring & PI
  - Resources & role owners
  - Data Analysis
  - Capability baselines
  - Tying it up - QMS
- Summary
  - Process asset library
  - Continuous improvements methods

### Target Audience

Systems and/or software Engineering managers and practitioners.

### Course Duration

3 days

### Prerequisite

None

## Product & Process Quality Assurance

### *Overview*

This course is designed to train professionals on the principles and practices of auditing the organization's quality system also called process audits.

The course shall prepare potential internal quality auditors to conduct, report and audit for compliance to predefined QMS and a standard or a model which is being followed in the organization. The course will help to devise, plan and maintain an effective internal audit program. It encourages the acquisition of sound and constructive audit skills and initiates process improvements through correction of non-conformities and preventative actions.

### *Topics to be Covered*

- Quality concepts
- Audit systems
- Audit planning
- Checklists
- Opening meetings
- Audit investigations
- Record findings
- Audit reporting
- Corrective actions & follow-ups

### *Target Audience*

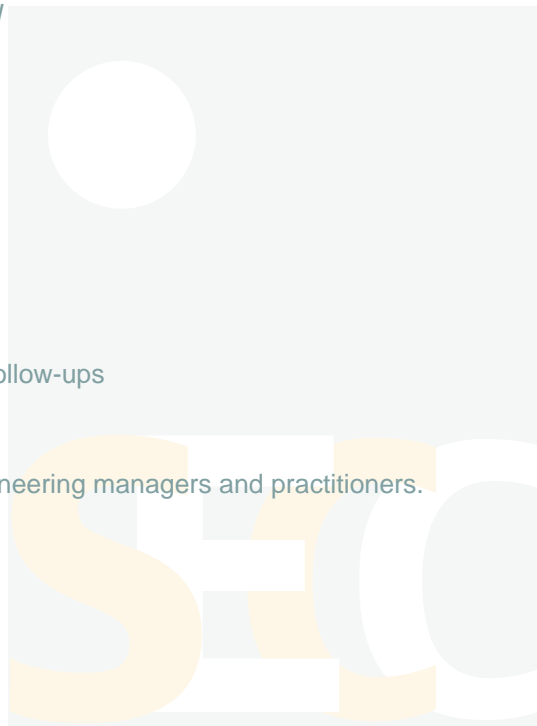
Systems and software engineering managers and practitioners.

### *Course Duration*

1 day

### *Prerequisite*

None



## Project Planning Monitoring and Control

### *Overview*

This course provides project managers and team leaders with the basic concepts required for effective project tracking, as well as the techniques required to prepare an effective project plan. The course also connects the principles of project planning to the CMMI context.

### *Topics to be covered*

- Basics of project planning
- Delphi estimation technique planning
- Work breakdown structure
- Scheduling using PERT / CPM /GANTT
- Resource leveling
- Basics of project monitoring and control
- Monitoring and control
- Status reporting
- Project monitoring and control and CMMI

### *Target Audience*

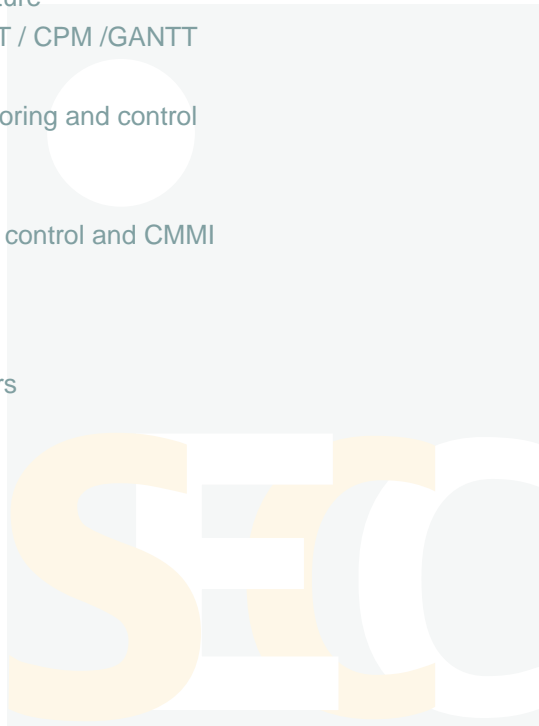
- Project managers
- Module/team leaders

### *Course Duration*

1 day

### *Prerequisite*

None



## Essential Requirements Management

### *Overview*

The course focuses on the fundamentals of requirements management and how they relate to the system development life cycle from writing the requirements to testing. The objective of this course is to improve the likelihood that products being constructed in time and satisfies customer needs. The course examines key elements of the software requirements management program and identifies pitfalls where many programs have failed.

Emphasis will be given to the techniques that have proved successful in real-world programs, methods for implementing them and strategies for determining whether the techniques are being effectively implemented.

### *Topics to be covered*

- Why do requirements management?
- Defining requirements
- The requirements management process
- Scope and terms
- Introduction to requirements activities
- Requirements characteristics
- Requirements elicitation
- Requirements triage
- Requirements specification
- Manage system scope
- Refine the system definition
- Manage changing requirements
- Requirements across the product lifecycle
- Institution aliasing requirement management
- Future directions: use-case modeling
- Managing software requirements

### *Target Audience*

Software engineers; business development personnel, project managers and practitioners interested in upgrading their skill set on requirement Management.

### *Course Duration*

2 days

### *Prerequisite*

None

## Supplier Agreement Management

### *Overview*

The course provides the basics for supplier agreement management with emphasis on supplier needs, acquisition, evaluation, process and agreement.

### *Topics to be covered*

- Introduction
  - Supplier, types of Supplier
- Establishing supplier need
  - Preparing requirements for supplier
- Explaining acquisition
  - Types of acquisition
  - Selection criteria for acquisition
- Identification and selection of supplier
  - Evaluation process
  - Factors for evaluating supplier
  - Negotiation stages
- Activities to be considered for various acquisition types
- Establishing supplier agreement
  - Types of agreement
  - Contents of supplier agreement
- Monitoring supplier
- Explanation of methods of monitoring suppliers
- Accepting deliverables from supplier
- Using deliverable
- Revising supplier agreement
- Interaction of other process area with supplier management
- Supplier management-SEI-CMMI

### *Target Audience*

System and/or software Engineering managers and practitioners.

### *Course Duration*

1 day

### *Prerequisite*

None

## Technical Solution and Product Integration

### *Overview*

The course aims at providing participants with the basics and fundamental of technical solution and product integration.

### *Topics to be covered*

- Introduction
  - Context setting with an example
  - Overview to software Engineering process
- Brief on software Engineering
  - Analysis modeling
  - Design Engineering
  - Architectural design
  - Component level design
  - User interface design
- Technical solution
  - Purpose
  - Introduction
  - CMMI perspective
  - Case study
- Product integration
  - Purpose
  - Introduction
  - CMMI perspective
  - Case study
- Technical solution and product integration
  - Service Industry
- Case studies used during the course
  - Accounting package
  - ATM
  - Travel agency

### *Target Audience*

Project managers and leaders, programmers, testers , designers, analysts, quality assurance, personnel and software quality controllers.

### *Course Duration*

1 day

### *Prerequisite*

None

## Verification and Validation

### *Overview*

This course provides participants with: a clear understanding of the inspection technique differentiating it from other review techniques; an understanding of the issues involved in implementing inspections; as well as a comprehensive process for test planning, test execution and result reporting. The course also imparts skills for participating in an inspection as a leader, recorder and author

### *Topics to be covered*

- Verification as part of CMMI framework
- Quality concepts
- Reviews
- Introduction to Fagan's inspection
- Technique
- Fagan's inspection technique - process
- Planning
- Fagan's inspection techniques - roles
- Automation of testing activities
- Testing
- Unit testing
- System testing
- Standards & testing

### *Target Audience*

Project managers and leaders, programmers, testers, designers, analysts, Quality Assurance, personnel and software quality controllers.

### *Course Duration*

2 days

### *Prerequisite*

None



## G. SPI for SME's Training

*The main focus of this track is to introduce to participants the skills and procedures necessary to help SME's improve their software processes including: project management, requirements management, product Engineering, configuration management, quality assurance and peer review.*

## SPI for SME's Training

<b>G001</b> Project Management	64
<b>G002</b> Product Development I: Requirements Management & Development	65
<b>G003</b> Product Development II: Design & Implementation	66
<b>G004</b> Configuration Management	67
<b>G005</b> Peer Review	68
<b>G006</b> Quality Assurance	69
<b>G007</b> Software Internal Auditor Training	70
<b>G008</b> Software Metrics	71
<b>G009</b> Software Estimation	72

## Project Management

### *Overview*

The main objectives of this course is to provide participants with the basic requirements and fundamentals of project management including project Planning, monitoring and control.

### *Topics to be Covered*

- Project management introduction
- Project initiation and planning
- Project estimation
- Project management plan
- Project execution and control (monitoring & metrics)
- Project closure
- SPIG

### *Target Audience*

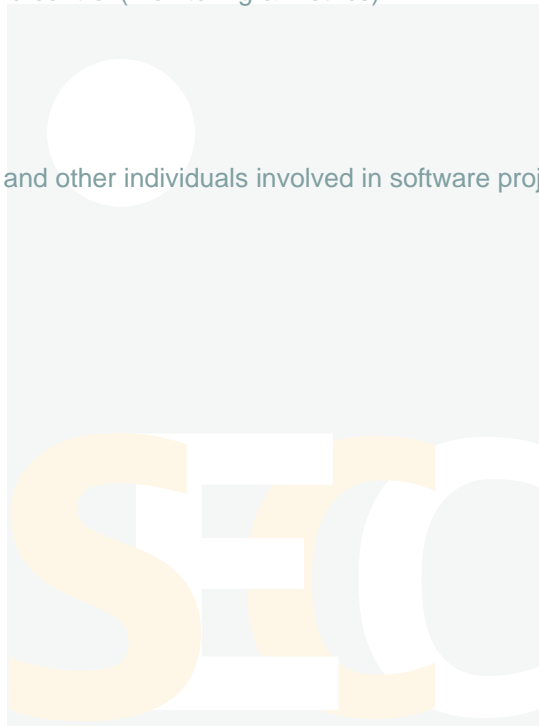
Software project managers and other individuals involved in software project planning, monitoring and control.

### *Duration*

3 days

### *Pre-Requisites*

None



## Product Development I: Requirements Management & Development

### *Overview*

Successful software projects are built on a foundation of well-understood and well documented requirements. However, many organizations get caught in the process details preventing them from collecting, documenting and managing their requirements effectively. The objective of this course is to introduce its participants to the skills and procedures fundamental to requirements elicitation and analysis, development and validation and requirements management throughout the product development cycle. The course lays the foundation for actually implementing improved requirements Engineering and management practices not just hear about them.

### *Topics to be covered*

- Elicit and manage requirements
- The requirements management process
- Customize the requirements management process defined in the SPIG for SMEs
- Software Requirements Specification document (SRS)
- Overview on supportive processes for the requirements process

### *Target Audience*

Anyone involved with defining, understanding or analyzing requirements for a software product will find useful information in this workshop. The primary audience is those individuals directly involved in software development. A secondary audience is marketers, product managers who are charged with specifying the features and attributes of a software product.

### *Duration*

2 days

### *Pre-Requisites*

None

## Product Development II: Design & Implementation

### *Overview*

To provide the basic concepts of the software product engineering process as described in the software process implementation guide for small and medium enterprises (SPIG). The Course provides a first step towards a long-term journey to produce a quality product through improving the software development process including the design, implementation, testing and delivery to the customer.

### *Topics to be Covered*

- Standards, methodologies and techniques to promote producing quality product
- Standard processes for developing the product
- Appropriate product life cycle to guide software development process
- Integration strategy for the assembly of the product components
- Verifications & validation throughout the SDLC
- Standards for testing
- Standard documentation for the software development work products

### *Target Audience*

- Software development / design engineers
- Software testing engineers

### *Duration*

3 days

### *Pre-Requisites*

None

## Configuration Management

### *Overview*

The main objective of this course is to provide participants with the basics and fundamentals of configuration management with emphasis on developing CM Plan and its implementation.

### *Topics to be Covered*

- Configuration management terminology
- CM organizational elements and responsibilities
- CM policy, process and procedures for SME
- CM activities
- Establishing the configuration management plan for a project

### *Target Audience*

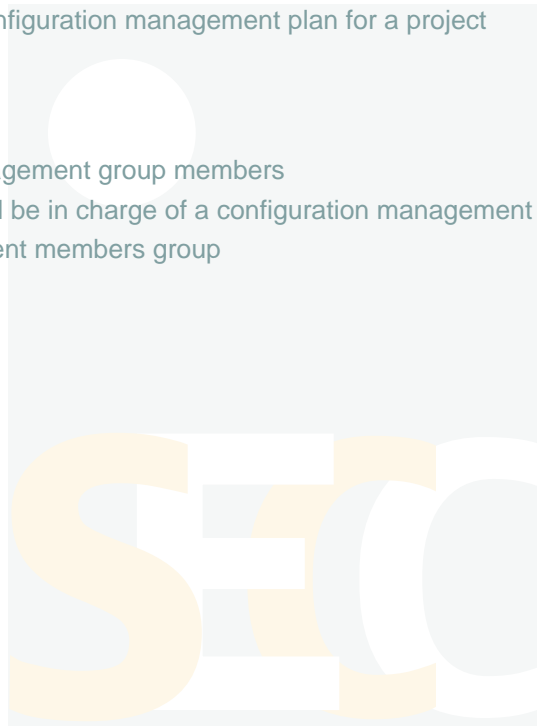
- Project managers
- Configuration management group members
- Any person who will be in charge of a configuration management role
- Process improvement members group

### *Duration*

2 days

### *Pre-Requisites*

None



## Peer Review

### *Overview*

To provide the basic concepts of the Peer Review Processes as described in the software process improvement guide (SPIG) for small and medium enterprises. The course provides a first step to develop and improve the skills of the participants about peer reviews to lead a software project to a successful conclusion.

### *Topics to be covered*

- Standards, methodologies and techniques to promote producing Quality product
- The phases and components of the processes in the SPIG
- The objectives, principles and methods of peer reviews
- Standard techniques in planning and organizing peer reviews
- The roles of reviewers
- Techniques how to evaluate readiness and completion criteria for peer reviews
- Build in capabilities of conducting, facilitating and reporting peer reviews
- Standard techniques in tracking and confirming actions coming out of peer reviews
- Peer reviews during software development project life cycle using real-life case studies

### *Target Audience*

Any member who will participate in peer reviews as well as quality members and process improvement members.

### *Duration*

2 days

### *Pre-Requisites*

None

## Quality Assurance

### *Overview*

The aim of the course is to introduce the trainees to software quality assurance (SQA), in which it is the concern of every software engineer to improve quality, reduce cost and improve product time-to-market. The course aims to cover the software quality assurance fundamentals, basic concepts, terminologies, standards as well as QA measurements with Emphasis on simplified guidelines and templates, all of which are included in the SQA Process within the SPIG **Handbook**.

### *Topics to be covered*

- Basic QA concepts in the framework of SPIG
- Terminology and standards for QA
- SQA group, roles and responsibilities
- Develop QA plan
- QA activities performed on every software project
- Measurements and metrics
- Sources of risk
- Quality checks on software products and processes
- Guidelines and templates

### *Target Audience*

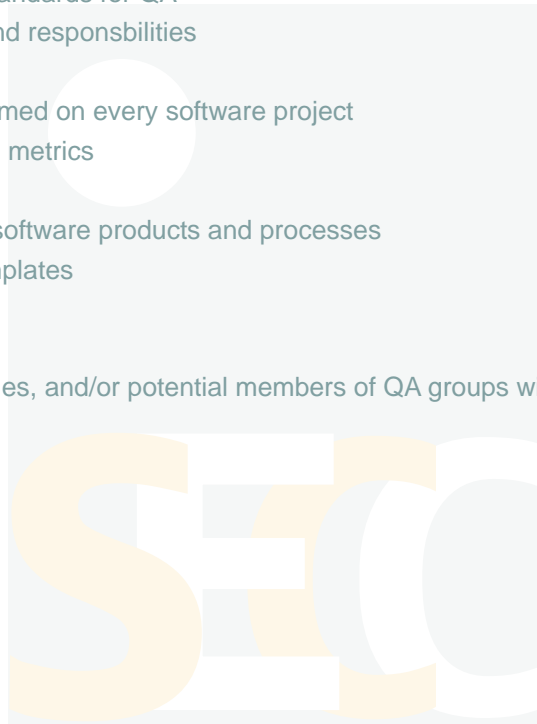
Managers of small companies, and/or potential members of QA groups within small and medium-sized companies.

### *Duration*

2 days

### *Pre-Requisites*

None



## Software Internal Auditor Training

### Overview

This course has been designed to train software professionals in the principles and practices of auditing the organization's quality system also called process audits.

The course shall prepare potential Internal quality auditors to conduct, report and audit for compliance to - Predefined QMS and a standard or a model like CMMI<sup>®</sup>, ISO 9001-2000, which is being followed in the organizations.

The course will help to devise, plan and maintain an effective internal audit program in a software organization. It shall encourage the acquisition of sound and constructive audit skills and initiate process improvements through correction of non-conformities and preventative actions.

### Target Audience

- Quality concepts
- Quick recap of CMMI<sup>®</sup>
- Quick recap of ISO
- The audit system
- Audit Planning
- Checklists
- The opening meeting
- Audit investigation
- Recording findings
- Audit reporting
- Corrective action and follow-up

### Target Audience

The course has been designed for those wishing to :

- Become internal quality auditors (for auditing of software process compliance)
- Acquire sufficient appreciation of internal auditing of processes in a software company
- Enhance quality through and effective internal audit process
- Extend and develop their previous knowledge of quality, systems and auditing skills
- Use the auditing skills for ISO 9001-2000 or CMMI<sup>®</sup> implementation

### Duration

1 day

### Pre-Requisites

- Understanding of Software Project execution process / Software Engineering concepts
- Prior understanding of the ISO 9001-2000 & CMMI<sup>®</sup> model is preferred

## Software Metrics

### Overview

The course provides a conceptual and practical framework for designing and implementing a software metrics program. Participants will be able to understand the usage of metrics in identifying and solving problems, leading to a more effective management of software projects and products. The course also covers concepts of Metrics in SW CMM® and CMMI® implementation.

### Topics to be covered

- Concept of Measurement
  - Basic measures
  - Derived metrics
  - Sample measurements
  - Prerequisite to measurements
- Managing projects using metrics
  - Managing schedules, cost, effort
- Managing processes using Metrics
  - Process capabilities
  - Process performance
  - Process improvement goals
- Managing product quality and performance using metrics
  - Defect metrics
  - Performance metrics like reliability, maintainability etc.
- GQM (goal question metric) based metrics program management
  - Concepts
  - Application
- Metrics and process models
  - Metrics and CMM®
  - Metrics and CMMI®
  - Metrics and Six Sigma
- Using statistical techniques in metrics analysis
  - Applying seven QC tools in metrics analysis
- Setting up metrics program
- Case study
  - Objective
  - Metrics collection mechanism
  - Metrics storage and presentation
  - Metrics tools
- Case studies

### Target Audience

Software developers, software process engineers, software quality practitioners and software quality managers

### Duration

1 day

### Pre-Requisites

None

## Software Estimation

### Overview

The objective of the course is to provide inputs on function points and COCOMO, which are perhaps the most popular methods of sizing and estimating software systems today. Function point analysis developed is now universally accepted as an adequate and effective measure of functionality of a business, commercial, MIS software system. Thus, representing a vital first step towards generating realistic estimates. The course also covers COCOMO model which is an estimation model used to estimate effort and calendar time.

### Topics to be covered

- Introduction and overview
  - Factors that affect productivity
  - The size effect
- Function point analysis
  - Unadjusted function points
  - General system characteristics
  - Case study
  - Quick function point analysis
  - FPA in enhancement projects
- COSMIC FFP
  - Overview : measurement context (purpose, scope and viewpoint)
  - Mapping rules and method
  - Measuring rules and methods
  - Data movement functions (entry, exit, read, write)
  - Reporting
  - Case study/ example
  - Converting: function points <-> COSMIC cfu
- COCOMO (**CO**nstructive **CO**st **MO**del)
  - Basic COCOMO
  - Intermediate COCOMO
  - Detailed COCOMO
  - Delphi's wideband estimation technique
  - Using the various techniques / models
  - Estimation capability maturity

### Target Audience

Project managers and leaders , marketing personnel and metrics group and systems analysts

### Duration

2 days

### Pre-Requisites

None



## H. Software Testing

*The main objective of this track is to provide participants with the methodologies, techniques and tools necessary for software testing including: test planning and design; inspection and reviews and testing automation.*

## Software Testing

<b>H001</b>	Software Test Planning and Design	76
<b>H002</b>	Writing Testable Requirements	77
<b>H003</b>	Software Inspections and Reviews	78
<b>H004</b>	Managing Testing Process	79
<b>H005</b>	Principles Architectures and Strategies of Test Automation	80

## Software Test Planning and Design

### Overview

This course offers a systematic approach to effective software testing. The course covers the different types of testing performed at each phase of the software lifecycle and issues involved in these types of tests.

The course will also cover the test design phase of the testing lifecycle. It starts with in-depth coverage of test case design techniques both for black and glass box testing. This includes equivalence class partitioning, boundary value analysis, cause-effect diagram, decision tables and decision trees. The last part of the course will cover test design activities and writing Test design specification, including documenting test conditions, test cases, test scripts and procedures, as well as the expected results.

### Topics to be covered

- Principles of software testing
- Disciplined software testing practices
- Testing levels
- System test planning
- Test case design techniques
- Systems test design

### Target Audience

- Software testing programmers interested in performing effective unit testing
- Development or test managers who are seeking a better control over the different test activities and the quality of the software product
- Test Engineers and test analysts interested in performing effective planning, designing as well as monitoring all test activities

### Course Duration

2 days

### Prerequisite

None

## Writing Testable Requirements

### Overview

This course covers details on how to document different types of requirements in a form that is testable. The course adopts a model-driven requirement process with emphasis on assuring completeness, correctness, testability, and precision of requirements. The course also discusses the elements of requirement management process.

### Topics to be covered

- Requirement validation
  - Types of requirements
  - Requirement traceability
  - Functional Vs quality requirements
  - Documenting quality requirements
  - Assessing the quality of software requirements
  - Forms of requirements
- The common sense model for software development and testing
- The requirement management process
- Models for exploring and refining requirements
  - Data models
  - Process models
  - Use cases
  - State models
  - User interface models

### Target Audience

- Test and quality professionals
- Requirement Engineers
- Business and system analysts
- Everyone involved in writing, validating and testing requirements

### Course Duration

2 days

### Prerequisite

None

## Software Inspections and Reviews

### *Overview*

Software inspections have proved to be very effective in capturing more defects early enough to avoid the cost of rework. Many types of defects can only be found by inspection, yet many organizations are struggling to make the inspection process effective and achieve the real benefits of inspections. This course will examine the different elements that make an effective inspection process and help establish a process which is customized for your organization.

### *Topics to be covered*

- Introduction
- Establishing the process
- Gathering relevant information
- In-class team exercise
- Defining a standard inspection process
- Defining process variations
- Training
- Monitoring
- Elements of an inspection process
- Planning
- The overview meeting
- Preparation
- The inspection meeting
- Record keeping and recording
- Rework and follow-up
- Causal analysis

### *Target Audience*

All software persons involved in the software development and maintenance process including managers of all levels.

### *Course Duration*

2 days

### *Prerequisite*

None

## Managing Testing Process

### *Overview*

The course presents a “Best Practice” risk-based methodology for planning, sizing, managing and executing the test effort that is both defensible and reproducible. Case studies illustrate how to tailor the methods to get the best results in traditional plan driven, Rapid Application Development (RAD) and Agile /extreme efforts. These test methods are applicable whether for testing Internet/Intranet applications, traditional client server application, as well as mobile application environments, or closed systems and embedded firmware environments.

### *Topics to be covered*

- Risk based test management
  - Introduction
  - Fundamental methods and metrics for software testing
  - Test estimation and sizing
  - Identify and enumerate tests, fill out the inventory, prioritize the tests, identify the most important tests, estimate the effort, track the test effort and evaluate it
  - Analysis techniques for enumerating tests and test environments
  - Risk analysis
  - Estimating the test effort
  - Adding value beyond the test effort
- Identifying the most important tests
  - Introduction: analytical methods for performing risk based testing
  - Analytical methods for picking the most important test
  - Applying risk analysis and ranking to selected tests
- Executing the most important tests
  - Managing test execution and reporting test progress
  - Reporting and tracking bugs
  - Measuring the performance of the test effort and demonstrating the value of testing

### *Target Audience*

Developers and programmers who are responsible for testing and quality assurance

### *Course Duration*

2 days

### *Prerequisite*

None

## Principles Architectures and Strategies of Test Automation

### *Overview*

This course provides an introductory approach to test automation. The objective is to provide the trainees with a general knowledge of how to incorporate test automation into current test practices. It also provides knowledge on how to get started, develop successful test automation strategies for short and long term ROI, and manage the test automation expectations of testers, developers, managers and business clients.

### *Topics to be covered*

- Introduction
- Automation framework
- Test automation expectations
- Automation development
- Customize test automation
- Modularize test automation
- Workflow testing
- Class project(s)
- Review session

### *Target Audience*

This course is valuable to anyone considering test automation in the process of implementing test automation, or who is looking to improve the test automation they have already implemented. The audience is any stakeholder that would benefit from the investment of automation. This would include but is not limited to business analysts, test managers as well as practitioners, developers, project managers, CIOs, CTOs and business application owners.

### *Course Duration*

2 days

### *Prerequisite*

None



# I. SEI Training Track

*This track covers courses from SEI providing participants with courses related to:*

- *CMMI based courses covering main SEI courses related to CMMI*
- *TSP/PSP based courses covering main SEI courses related to Team Software Process and Personal Software Process.*

## SEI Training Track

<b>I1</b>	CMMI Based Courses	84
<b>I101</b>	Introduction to CMMI	85
<b>I102</b>	CMMI Executive Workshop	86
<b>I103</b>	Leading and Managing CMMI Based Process Improvement	87
<b>I104</b>	CMMI Based Implementation Essentials	88
<b>I105</b>	Intermediate Concepts of CMMI	89
<b>I106</b>	Mastering Process Improvement	90
<b>I107</b>	Defining Software Process	91
<b>I108</b>	High Maturity with Statistics Outlines	92
<b>I2</b>	TSP / PSP Official Training Courses	93
<b>I201</b>	Introduction to PSP : Personal Software Process	94
<b>I202</b>	Leading a Development Team	95
<b>I203</b>	PSP For Engineers I : Planning PSP For Engineers II : Quality	96
<b>I204</b>	TSP Executive Strategy	97

## CMMI Based Courses

Code	Course Name	Duration (Days)
I101	Introduction to CMMI	3
I102	CMMI Executive Workshop	2
I103	Leading and Managing CMMI Based Process Improvement	2
I104	CMMI Based Implementation Essentials	2
I105	Intermediate Concepts of CMMI	5
I106	Mastering Process Improvement	5
I107	Defining Software Process	2
I108	High Maturity with Statistics Outlines	2

## Introduction to CMMI

### *Overview*

This course introduces systems and software engineering managers and practitioners, appraisal team members and engineering process group (e.g., SEPG, EPG) members to Capability Maturity Model® Integration (CMMI®) fundamental concepts. CMMI models are tools that help organizations improve their ability to develop and maintain quality products and services. CMMI models are an integration of best practices from proven process improvement models, including the CMM® for Software, EIA 731 and the Integrated Product Management CMM. This course focuses on the CMMI models with a staged representation.

### *Topics to be covered*

- Introduction to CMMI
- Model Based Process Improvement
- Overview of CMMI Models and the staged representation
- Maturity level 2 ( managed )
- Maturity level 3 ( defined )
- Integrated product and process development
- Maturity level 4 ( quantitatively managed )
- Maturity level 5 ( optimizing )
- Summary

### *Target Audience*

- Product developers and process implementers
- Anyone interested in learning more about CMMI

### *Course Duration*

3 days

### *Prerequisite*

None

## CMMI Executive Workshop

### *Overview*

The main target of this course is to describe the background and advantages of Capability Maturity Model Integration (CMMI®), understand and correctly use selected CMMI®-specific vocabulary, explain the concepts of process area capability and organizational process maturity. It also describes the elements of an effective process for conducting process improvement, key elements for successfully managing the human and organizational components of a process improvement effort, as well as identify ways in which they can demonstrate their support for their organization's process improvement effort.

### *Topics to be covered*

The CMMI® executive workshop provides organization executives with a three pronged approach for successful process improvement (PI). Together, these three essential elements address the technical, procedural and organizational aspects of change. Failure to address any of these elements will result either in outright failure or in expenditure of far more time and resources than are necessary to achieve success. If executives ensure that the elements are addressed deliberately and explicitly, they can strongly influence the outcome of their organizations' PI efforts and lead their organizations to greatly improved quality, productivity and competitiveness in a global economy.

### *Target Audience*

- Product developers and process implementers
- Anyone interested in learning more about CMMI

### *Course Duration*

2 days

### *Prerequisite*

None



## Leading & Managing CMMI Based Process Improvement

### *Overview*

This course aims at recognizing and assigning change management responsibilities to individuals involved with their organizations' process improvement effort based upon well-defined roles. Also it will establish and maintain effective sponsorship for process improvement at all management levels within their organizations and communicate effectively with organization members regarding various aspects of their organizations' PI efforts.

### *Topics to be covered*

Effective CMMI<sup>®</sup>-based process improvement (PI) requires that an effective infrastructure be put in place to manage and administer its execution. This infrastructure requires the involvement of individuals who assume, or are assigned, specialized roles and responsibilities. The training provided by leading and managing CMMI<sup>®</sup>-Based Process Improvement provides participants with the knowledge they need to put an improvement infrastructure in place and use it effectively. The course integrates the use of lecture and exercise in a way that allows participants apply the course content immediately to their own circumstances and situations.

### *Target Audience*

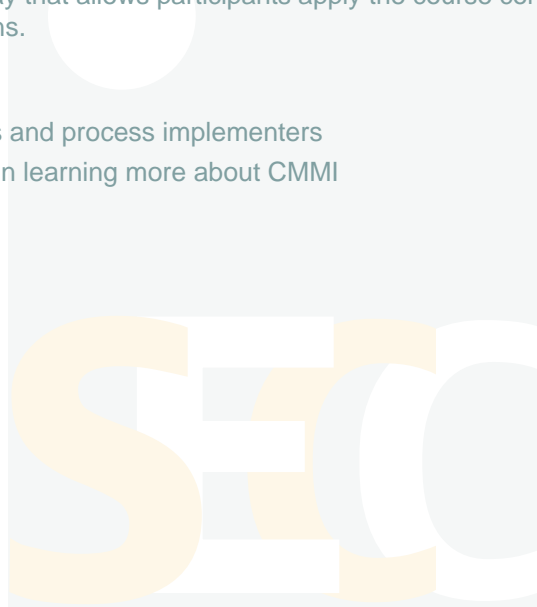
- Product developers and process implementers
- Anyone interested in learning more about CMMI

### *Course Duration*

2 days

### *Prerequisite*

None



## CMMI Based Implementation Essentials

### *Overview*

The aim of this course is to identify and characterize organizational factors that may affect organizations' CMMI-based process improvement (PI) efforts. It also plans for reducing the impact of productivity losses associated with introducing new or revised business practices, and builds widespread support for their organizations' PI efforts while mitigating the effects of individual resistance to change.

### *Topics to be covered*

Experience in many organizations around the world shows that organizational barriers pose the greatest threat to successful implementation of CMM<sup>®</sup> - and CMMI<sup>®</sup>-based process improvement initiatives. The CMMI<sup>®</sup> Implementation essentials training course provides participants with proven approaches for dealing explicitly and effectively with these delicate, difficult and crucial issues. The course integrates the use of lecture and exercise in a way that allows participants to apply the course content immediately to their own circumstances and situations.

### *Target Audience*

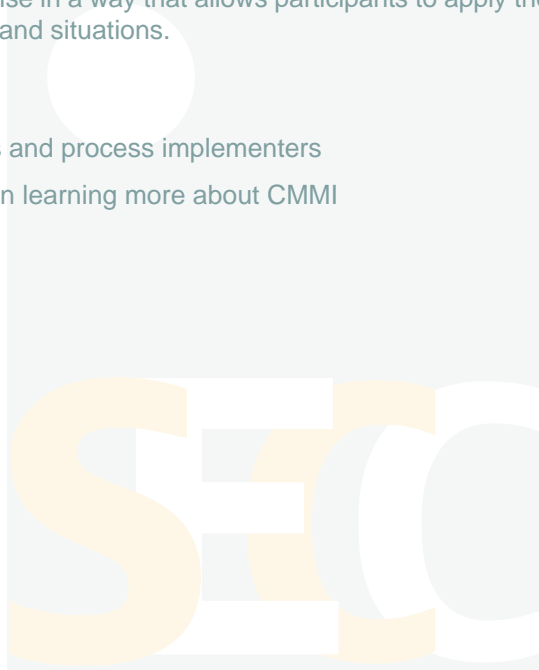
- Product developers and process implementers
- Anyone interested in learning more about CMMI

### *Course Duration*

2 days

### *Prerequisite*

None



## Intermediate Concepts of CMMI®

### Overview

This course introduces candidate SCAMPI Lead Appraisers, systems and software Engineers, Engineering process group (e.g., EPG, SEPG) members, candidate CMMI® instructors and others to detailed CMMI concepts, including the relationships among CMMI model components. CMMI models are tools that organizations can use to help improve their ability to develop and maintain quality products and services. CMMI models are an integration of best practices from proven process improvement models, including the CMM for Software, EIA 731 and the Integrated Product Management CMM. Successful completion of this course requires a passing score on a test that covers the concepts taught in this course. This course is a prerequisite for SCAMPI Lead Appraiser Training and CMMI Instructor Training.

### Topics to be covered

- Establish links from their past model use and experiences to CMMI® models
- Understand relationships among model components, including both staged and continuous model representations
- Understand how to interpret and apply CMMI® models effectively
- Share, learn and exchange ideas with other course participants

### Target Audience

- Candidate lead appraisers for the SCAMPI appraisal method
- Systems and software engineers, SEPG, EPG process personnel who need more in depth knowledge of CMMI® models
- Candidate instructors interested in teaching the Introduction to CMMI®

### Course Duration

5 days

### Prerequisite

- Complete an SEI authorized offering of an Introduction to CMMI® course (staged or continuous)
- Complete a minimum of 90 days of self- study, following the completion of the Introduction to CMMI® course, and obtaining experience with using the model before applying for the Intermediate class (e.g., serving on an Engineering process group, mapping CMMI® to the organizations processes, as well as serving as a member of an appraisal team, etc.)
- Carefully study the full content of one representation of the CMMI®SE/SW/IPPD/SS model, Version 1.1, or of the CMMI® book (Guidelines for Process Integration and Product Improvement), by Chrissis, et. al., published by Addison Wesley
- Carefully study chapters 1-5 of the other representation of the CMMI®SE/SW/IPPD/SS model, Version 1.1, or chapters 1-5 of the CMMI® book. You must understand both the continuous and staged concepts before coming to the class
- Complete and submit an assigned exercise before attending the course, according to instructions that will be supplied. This exercise will require 4-6 hours for most participants
- Capable of communicating well in English

## Mastering Process Improvement

### *Overview*

This course will help participants understand the process improvement life cycle, implement one effective process improvement infrastructure; understand a systems-level view of how process improvement is planned and implemented using a capability maturity model; know what knowledge and skills are important for process group members and how they can be applied; apply a toolkit of relevant process improvement and change management skills and techniques within their organization, and begin gaining confidence in their ability to perform their role in the process group.

### *Topics to be covered*

- Process improvement life cycle
- Process group and process improvement infrastructure
- Process change methodology (PCM)
- Skills and knowledge associated with the PCM

### *Target Audience*

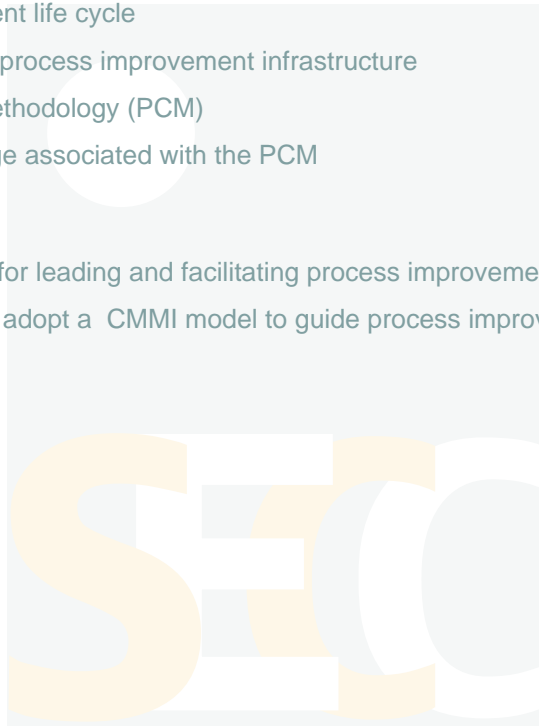
- Those responsible for leading and facilitating process improvement in their organization
- Those preparing to adopt a CMMI model to guide process improvement in their organization

### *Course Duration*

5 days

### *Prerequisite*

None



## Defining Software Process

### *Overview*

This course introduces participants to processes, methods, and skills for working as a team to understand, model and document current processes; analyze process efficiencies, and specify needed process improvements. Defining processes is a key step to achieving maturity level 2 of the Software CMMI, as well as to becoming certified for ISO 9000 and 9001. This course emphasizes the basic skills and knowledge required to effectively document and define processes and design improvements.

### *Topics to be covered*

- Conduct a process definition project
- Gather information and model processes effectively
- Identify weaknesses in current processes and find/develop alternatives
- Specify improvements and present them to management

### *Target Audience*

- Process action team (PAT) and/or Engineering process group (e.g., EPG, SEPG) members
- Anyone involved in process definition in an organization

### *Course Duration*

2 days

### *Prerequisite*

None



## High Maturity with Statistics Outlines

### *Overview*

This course provides participants with the software engineering and management practices of high maturity organizations as rated against the CMM for Software and CMMI models. The concepts and practices underlying maturity level 4 and the application of statistical process control (SPC) are a powerful combination that enables organizations achieve excellent software engineering practice. This course is designed to cover the concepts and issues that lie behind classical statistical process control and the steps associated with effectively implementing and using SPC to manage and improve software processes.

### *Topics to be covered*

- Place CMM-based process improvement in context: TQM, best practices, and people issues
- Deploy repeatable, defined processes
- Understand and interpret maturity level 4 of the SW-CMM and CMMI models
- Use basic tools such as scatter diagrams, run charts, regression analysis, analysis of variance and test of hypotheses
- Analyze process behavior with control charts
- Understand and interpret maturity level 5 of the SW-CMM and CMMI models

### *Target Audience*

Lead appraisers, lead assessors, lead evaluators, Engineering process group (e.g., SEPG, EPG) members and other engineering professionals focused on appraising or improving engineering processes.

### *Course Duration*

2 days

### *Prerequisite*

Introduction to CMMI course.

## TSP / PSP Training Courses

Code	Course Name	Duration (Days)
1201	Introduction to Personal Software Process (PSP)	2
1202	Leading a Development Team	3
1203	Personal Software Process (PSP) for Engineers : Planning & Quality	10
1204	TSP Executive Strategy	2



## Introduction to Personal Software Process

### Overview

This course prepares students to participate in Team Software Process<sup>SM</sup> (TSP<sup>SM</sup>) teams. It covers the key concepts and principles of the Personal Software Process<sup>SM</sup> (PSP<sup>SM</sup>) at a high level so that students understand the terminology and the philosophy behind the method. Students are guided through a series of lectures and exercises that teach the PSP process and prepare them to plan and track their work as part of a TSP project team. Successful completion of the course will enable participants understand the rationale and foundational concepts behind the PSP and TSP; to develop ways to measure the size, time and defects in their work. This is to enable them use the PSP planning framework to plan and track tasks; and to use the PSP quality strategy to manage the defects in their work and support the team's quality goals.

### Topics to be covered

- PSP process framework
- Measurement in the PSP
- Defining a personal process
- PSP planning framework
- Estimating and scheduling
- In-process control and tracking
- Defect management
- Process analysis and improvement
- Team software process

### Target Audience

- System, hardware and test engineers, technical writers, trainers, and others who are not software engineers
- Engineers participating on a TSP project
- Members of integrated product development teams that are building software-intensive systems

### Course Duration

2 days

### Prerequisite

None

## Leading a Development Team

### *Overview*

This course covers the key concepts and principles of the TSP and the Personal Software Process<sup>SM</sup> (PSP<sup>SM</sup>). The purpose of the course is to provide managers and team leaders with the knowledge and skills needed to effectively lead and coach TSP development teams.

The TSP was designed to provide both a strategy and a set of operational procedures using disciplined software process methods at the individual and team levels. At the individual level, the PSP shows managers and engineers how and why to plan and track their projects, demonstrates the principles and benefits of effective quality management, and involves the engineers in process measurement, management, and improvement. At the team level, the TSP builds accurate, achievable plans for software projects teams, provides a formal team-building process, and the mechanism for tracking progress against project plans.

### *Topics to be covered*

- An overview of the team leader's responsibilities and ways to lead and motivate the team
- An overview of the TSP and how it works to build self-directed high-performance teams
- Team management in a TSP team, the role of the team leader
- The TSP launch process, how the team builds its work plan and negotiates its commitment with management
- Understanding process data at the team level, coaching the team based on process data
- How the team tracks status against its plan and maintains its commitment to management
- Maintaining quality during software development
- How the team measures and manages development quality, and produces quality products
- How to report team status to management
- How to handle various leadership challenges

### *Target Audience*

First-level software managers who directly manage software development, such as software project managers, software team leaders, supervisors.

### *Course Duration*

3 days

### *Prerequisite*

None

## PSP For Engineers I: Planning, PSP For Engineers II: Quality

### *Overview*

This course provides the principles, concepts, and benefits of the PSP, a process-based approach for developing software. Students who complete this course will be able to apply the PSP methods to their own personal work process and participate on a Team Software Process (TSP) team. Students learn how to measure and analyze their personal software processes, use process data to improve their personal performance and apply PSP methods to other structured tasks.

### *Topics to be covered*

- PSP process-based approach for developing software
- PSP process-based approach for developing software Measure and analyze their personal software processes
- PSP process-based approach for developing software Use process data to improve their personal performance
- PSP process-based approach for developing software Apply PSP methods to other structured tasks
- PSP process-based approach for developing software Target Audience  
Software Engineers, software Engineering instructors, software quality management instructors and third-party vendors of software Engineering or quality

### *Course Duration*

- Planning : 5 days
- Quality : 5 days

### *Prerequisite*

Participants must be proficient in at least one programming language (excellent programmers with inquisitive minds). Due to the quick pace of the courses, students will not be successful if they use this course to learn a new language. Students must bring their personal portable.

## TSP Executive Strategy

### *Overview*

This course covers the key concepts and principles of the Team Software Process (TSP) and Personal Software Process (PSP) from a management perspective. The purpose of the course is to provide the foundation that managers need to start introducing and applying the TSP in their organizations. Organizations that develop software, recognize that controlling their software process significantly affects their ability to be successful in business. However, organizations still struggle when trying to apply disciplined methods in the software process. Historically, this struggle has been due to a lack of operational procedures that teams and individuals can use to develop software in a disciplined fashion. The TSP was designed to provide both a strategy and a set of operational procedures using disciplined software process methods at the individual and team levels. At the individual level, the PSP shows managers and engineers how and why to plan and track their projects, demonstrates the principles and benefits of effective quality management and involves the engineers in process measurement, management and improvement. At the team level, the TSP builds accurate, achievable plans for software projects teams, provides a formal team-building process, and the mechanism for tracking progress against project plans.

### *Topics to be covered*

- Strategic perspective of the software business, its problems and possible solutions
- Overview of the PSP and how it provides a means for introducing sound engineering practices into software Engineering work
- Overview of the TSP and how it works to build self-directed high-performance teams
- Rational management (managing with facts and data) and why this type of management is required to maintain successful TSP teams
- Quality during software development, how TSP teams manage development quality and produce quality products
- Introducing TSP into an organization

### *Target Audience*

Executives and middle managers who are responsible for software development or maintenance , managers who are responsible for software support areas such as software quality assurance, and software engineering process groups, systems or hardware engineering, documentation, or finance

### *Course Duration*

2 days

### *Prerequisite*

None

## J.Rational Training

*The main focus of this track is to provide participants with the tools and techniques of Rational system covering the full suite of Rational, allowing developers to use these tools and techniques efficiently in the overall process of software development.*

## Rational Training

<b>J001</b>	Essentials of Rational Requisite Pro	100
<b>J002</b>	Essentials of Rational ClearQuest	101
<b>J003</b>	Essentials of Rational ClearCase	102
<b>J004</b>	Essentials of Rational ClearCase UCM for Windows	103
<b>J005</b>	Essentials of Configuration Management with Rational ClearCase® UCM	104
<b>J006</b>	Essentials of Rational Project Console	105
<b>J007</b>	Essentials of Rational SoDA for Microsoft Word	106
<b>J008</b>	Essentials of Test Management with Rational Testingmanager	107

## Essentials of Rational RequisitePro

### Overview

This course provides instruction on the use of Rational® RequisitePro®, Rational's tool for capturing and managing requirements for a project. Students will gain an understanding of RequisitePro project structure by using the tool to apply software development best practices as recommended in the Rational Unified Process®. Upon completion of the course, participants should be able to:

- Define and customize RequisitePro project structure and components
- Create and build a project
- Create, import, and revise requirements, attributes, and requirement documents
- Define hierarchies of requirements
- Trace requirements to one another
- Manage changing requirements through views, queries, and metrics
- Participate in group discussions

### Topics to be covered

- Introduce requirements management and RequisitePro product
- Establish a RequisitePro project
- Gather and enter requirements
- RequisitePro documents
- Views and traceability
- Offline authoring, metric, and group discussions

### Target Audience

Project team members responsible for capturing, managing, and interacting with the project requirements. The primary audience is analysts.

### Duration

1 day

### Prerequisite

Students must have a solid understanding of:

- Requirements management
- The software development lifecycle

## Essentials of Rational ClearQuest

### *Overview*

This course introduces new users to using Rational ClearQuest to track defects and change requests. It covers the basic concepts involved in defect and change request tracking and how Rational ClearQuest facilitates this process. Upon completion of the course, participants should be able to:

- Plan and implement defect and change tracking processes
- Use Rational ClearQuest to track defects and change requests
- Understand how to analyze and report defect and change tracking results with Rational ClearQuest
- Know how to use Rational ClearQuest Web and e-mail

### *Topics to be Covered*

- Module 1: Concepts of Change Management and the architecture of Rational ClearQuest
- Module 2: Change Management with Rational ClearQuest
- Module 3: Using Rational ClearQuest to analyze and report project status
- Module 4: Using Rational ClearQuest through the Web and e-mail

### *Target Audience*

Project team members concerned with tracking defects and change requests.

### *Duration*

1 day

### *Prerequisite*

Students should be familiar with defect and change tracking processes.

## Essentials of Rational ClearCase

### Overview

This course teaches developers how to perform common day-to-day source code change tasks using Rational ClearCase. It focuses on teaching the concepts and skills developers need to successfully manage source code changes in their development environments. Upon completion of the course, participants should be able to:

- Explain ClearCase concepts
- Set up their development workspace
- Access and work with files under source control
- Merge work to and from the integration workspace
- Create and work with snapshot views
- Configure workspaces for parallel development
- Perform advanced merge operations
- Apply ClearCase metadata including labels, attributes, and hyperlinks
- Generate ClearCase reports
- Manage software builds

### Topics Covered

- Getting started with ClearCase
- Typical developer workflow
- Workspace management
- The checkin/checkout model
- Versioning and branching software assets
- Snapshot and dynamic views
- Configuring workspaces for parallel development
- Integrating parallel development
- ClearCase metadata
- Managing builds with clear make and omake

### Target Audience

Project team members concerned with source code change tasks.

### Duration

2 days

### Prerequisite

Students should be familiar with software change management and versioning control concepts.

## Essentials of Rational ClearCase UCM for Windows

### Overview

This course introduces the concepts of Rational ClearCase Unified Change Management (UCM), Rational's software development best practice that integrates artifact and activity management. Students will learn to use Rational ClearCase UCM and Rational ClearQuest to perform common, day-to-day software development tasks such as joining a project, checking out files from source control and checking in files to source control, merging your work with the work of others, updating your workspace, and working offline. Upon completion of the course, participants should be able to:

- Demonstrate knowledge of ClearCase and UCM concepts
- Join a ClearCase UCM project and set up a developer workspace
- Find, set, and work on development activities
- Deliver and merge work
- Update your workspace
- Work offline with ClearCase UCM

### Topics Covered

- UCM concepts
- Joining a project
- Setting up a development workspace
- Finding and setting activities
- Working on activities
- Delivering activities
- Rebasing your workspace
- Advanced delivery scenarios
- Working offline

### Target Audience

Project team members concerned with source code change tasks.

### Duration

1 day

### Prerequisite

Students should be familiar with:

- Software change management and versioning control concepts
- Knowledge of ClearQuest is preferred

## Essentials of Configuration Management with Rational ClearCase® UCM

### Overview

Essentials of Configuration Management with Rational® ClearCase® UCM is a two-day, instructor-led course designed to teach configuration managers how to implement ClearCase UCM effectively in their development environment (Windows). This course introduces participants to the discipline of software configuration management (SCM) using a unified change management approach (UCM) and discusses the attributes of effective SCM. Participants learn what information is critical to include in a configuration management plan. They practice creating and implementing a usage model based on a simulated software development project. The usage model includes details about organizing project artifacts, streaming strategies, and project process and policies. Upon completion of the course, participants should be able to:

- Define software configuration management and list the attributes of an effective software configuration management system
- Explain the purpose and describe the contents of a configuration management plan
- Complete a ClearCase UCM usage model
- Implement a ClearCase UCM usage model by:
  - Setting up project artifacts
  - Creating UCM projects that implement a streaming strategy
  - Implementing project process and policies through ClearCase and ClearQuest

### Topics Covered

- Software configuration management overview
- Planning components, VOBs, and naming conventions
- Stream strategy and views
- Baselines and project policies
- Change request process model
- Setting up a UCM project
- Project maintenance and policy enforcement using triggers
- Monitoring Project Status using ClearQuest and ClearCase report builder/viewer
- Building and recommending baselines
- Build management with IBM Rational ClearCase
- Managing multiple projects in UCM

### Target Audience

Configuration managers.

### Duration

2 days

### Prerequisite

Students must have completed the Essentials of Rational ClearCase UCM course. Knowledge of ClearQuest is also required.

## Essentials of Rational Project Console

### *Overview*

**Essentials of Rational Project Console** emphasizes the practical uses of the Rational Project Console product, which provides a project console that enables Web reports of up-to-date project information and metrics graphics, extracted directly from the tools used to support software development activities. Upon completion of the course, participants should be able to:

- Describe the components of Project Console and how they relate to each other
- Create and customize the Project Console information structure
- Define and run dynamic Web-based reports directly from Rational tools and other sources
- Define and exercise elements of the metrics warehouse, which collects information directly from Rational tools and other sources
- Assess the usefulness of Project Console for different members of the development team

### *Topics to be Covered*

- Introduction to the Rational Project Console
- Customizing the console navigation tree
- Addition of static project information
- Creation of automatically updated Web reports
- Introduction to dashboard, including metrics display capabilities
- Introduction to the basics of the metrics warehouse and star schemas
- Description and design of target tables and source templates (specifications of source information and the corresponding warehouse storage tables)

### *Target Audience*

- Project managers
- Project team leaders
- Data administrators
- Process Engineers

### *Duration*

1 day

### *Prerequisite*

Students must have good knowledge of software project management and familiarity with Rational tools.

## Essentials of Rational SoDA for Microsoft Word

### *Overview*

This one-day course provides an introduction to using Rational® SoDA for Microsoft® Word®. It covers the generation of documents and reports, as well as the use of SoDA commands, templates, and source domains. Upon completion of the course, participants should be able to:

- Understand the Rational SoDA for Microsoft Word documentation process
- Understand the key concepts of SoDA for Microsoft Word
- Generate reports and regenerate documents from SoDA templates
- Use SoDA wizards and the template view
- Create and customize SoDA templates
- Issue SoDA commands
- Create tables in reports or documents
- Publish documents on the Web

### *Topics to be Covered*

- Module 1: The key concepts of the SoDA for Microsoft Word product
- Module 2: Creating and customizing SoDA templates
- Module 3: SoDA commands
- Module 4: Managing documentation with Rational SoDA for Microsoft Word

### *Target Audience*

Any project team member in the development lifecycle.

### *Duration*

1 day

### *Prerequisite*

Students must have very good knowledge of MS Word and familiarity with rational tools.

## Essentials of Test Management with Rational TestManager

### *Overview*

This course is designed to familiarize testing professionals with the basics of Rational® TestManager. It provides hands-on instruction for those who want to explore using Rational TestManager to manage test assets, build sophisticated test suites, generate reports, and log defects. Rational TestManager is used to centralize test results and test data to assist the analysis of the results. Upon completion of the course, participants should be able to:

- Create users and groups using Rational Administrator
- Build a test plan using Rational TestManager
- Define test inputs using Rational TestManager
- Define project iterations, configurations and computers using Rational TestManager
- Design and configure test cases using Rational TestManager
- Create and execute a manual test using Rational ManualTest
- Create and execute a suite using Rational TestManager
- Run planning reports using Rational TestManager
- Report defects using Rational ClearQuest from the Rational TestManager test log
- Create a project using Rational Administrator
- Create a Sybase SQL Anywhere Test Datastore

### *Topics to be Covered*

- Test management challenges
- Managing test assets
- Working with projects
- Inputs and activities of test planning and design
- Defining test inputs and registering test input sources
- Building a graphical test plan
- Defining, configuring, and designing test cases
- Creating and executing a manual test script and test suite
- Defining a test suite and test scenario
- Coordinating suite execution
- Distributing functional tests
- Creating and running planning reports and evaluation reports
- Change request lifecycle and submitting a defect

### *Target Audience*

QA professionals and software testers.

### *Duration*

1 day

### *Prerequisite*

Familiarity with QA processes, testing practices, and MS Windows platform.

## **K. Embedded Systems**

*This track is designed to provide participants and developers with methodologies, techniques and tools required to develop and design embedded system with emphasis on Linux environment, FPGA, ARM, RTOS, and multimedia based systems.*

## Embedded Systems

<b>K1</b>	<b>Embedded Linux Track</b>	
K101	Embedded Linux Basics	110
K102	Embedded Linux Kernel and Device Driver Development	111
K103	Embedded Linux Application Development and Debugging	112
<b>K2</b>	<b>FPGA Track</b>	
K201	Digital Design Using Verilog	113
K202	Verilog's VPI Standard	114
K203	FPGA Overview Based on Virtex™ FPGA	115
K204	HW/SW Co-Simulation for SoC Design and Verification	116
K205	HW/SW Co-Emulation for SoC Design and Verification	117
K206	Embedded System Co-Verification Methodologies	118
K207	HW/SW Co-Design for ARM-Based SoC	119
<b>K3</b>	<b>ARM Track</b>	
K301	ARM Architecture	120
K302	ARM Assembly	121
K303	ARM-Based Embedded System Development	122
K304	ARM-Based RTOS Porting	123
<b>K4</b>	<b>Embedded Multimedia Track</b>	
K401	Embedded GUI	124
K402	Embedded Audio Processing	125
K403	Embedded Video Processing	126
K404	Camera Support in Embedded Multimedia Systems	127
<b>K5</b>	<b>RTOS Track</b>	
K501	RTOS Basics	128
K502	Multitask Programming in RTOS	129
K503	µC/OS-II the RTOS	130
K504	Porting µC/OS-II to ARM Processor	131
<b>K6</b>	<b>Embedded Networking Track</b>	
K601	Basics of Embedded Networking	132
K602	Embedded Web Server	133

## Embedded Systems Tracks

Code	Course Name	Duration (Days)
<b>K1. Embedded Linux Track</b>		
K101	Embedded Linux Basics	1
K102	Embedded Linux Kernel and Device Driver Development	1
K103	Embedded Linux Application Development and Debugging	1
<b>K2. FPGA Track</b>		
K201	Digital Design Using Verilog	3
K202	Verilog's VPI Standard	2
K203	FPGA Overview Based on Virtex™ FPGA	1
K204	HW/SW Co-Simulation for SoC Design and Verification	1
K205	HW/SW Co-Emulation for SoC Design and Verification	1
K206	Embedded System Co-Verification Methodologies	1
K207	HW/SW Co-Design for ARM-Based SoC	1
<b>K3. ARM Track</b>		
K301	ARM Architecture	1
K302	ARM Assembly	1
K303	ARM-Based Embedded System Development	1
K304	ARM-Based RTOS Porting	1
<b>K4. Embedded Multimedia Track</b>		
K401	Embedded GUI	1
K402	Embedded Audio Processing	1
K403	Embedded Video Processing	1
K404	Camera Support in Embedded Multimedia Systems	1
<b>K5. RTOS Track</b>		
K501	RTOS Basics	1
K502	Multitask Programming in RTOS	1
K503	μC/OS-II the RTOS	1
K504	Porting μC/OS-II to ARM Processor	1
<b>K6. Embedded Support Track</b>		
K601	Basics of Embedded Networking	1
K602	Embedded Web Server	1

## Embedded Linux Basics

### *Overview*

Linux is widely used as an embedded operating system and its market share is continuously growing. Reasons as quality, reliability, and configurability make its use quite attractive. More important, its components and many of its development tools are available free of charge. This workshop describes basics of embedded Linux and what is needed to use Linux in the embedded domain.

### *Topics to be covered*

- Introduction to the embedded and real-time features of the Linux operating system
- Tools required to set up a Linux cross-development environment
- Introduction to Linux kernel architecture

### *Target Audience*

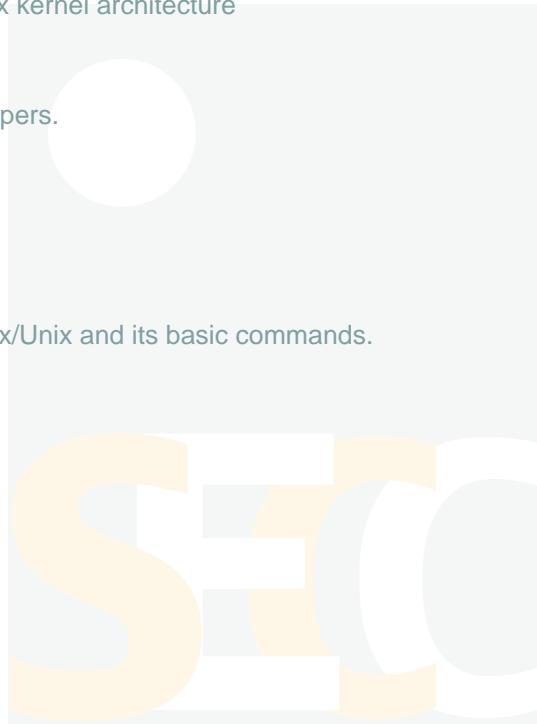
Embedded software developers.

### *Course Duration*

1 day

### *Prerequisite*

General Knowledge of Linux/Unix and its basic commands.



## Embedded Linux Kernel and Device Driver Development

### *Overview*

Linux is widely used as an embedded operating system and its market share is continuously growing. Reasons as quality, reliability, and configurability make its use quite attractive. More important, its components and many of its development tools are available free of charge. This workshop gives advanced concepts about kernel internals for embedded Linux. It also introduces the attendees to embedded Linux device driver development.

### *Topics to be covered*

- The embedded Linux file system
- Cross-compiling and booting the Linux kernel
- Modules support and device driver development in the Linux kernel

### *Target Audience*

Embedded software developers.

### *Course Duration*

1 day

### *Prerequisite*

- Embedded Linux basics
- Good C programming skills

## Embedded Linux Application Development and Debugging

### *Overview*

Linux is widely used as an embedded operating system and its market share is continuously growing. Reasons as quality, reliability, and configurability make its use quite attractive. More important, its components and many of its development tools are available free of charge. This workshop explains how to develop and debug an application so as to run on an embedded Linux system

### *Topics to be covered*

- Developing simple application using Linux libraries
- Using CW-Linux/GNU to compile, download, and debug the application
- Introduction to Linux process/thread model

### *Target Audience*

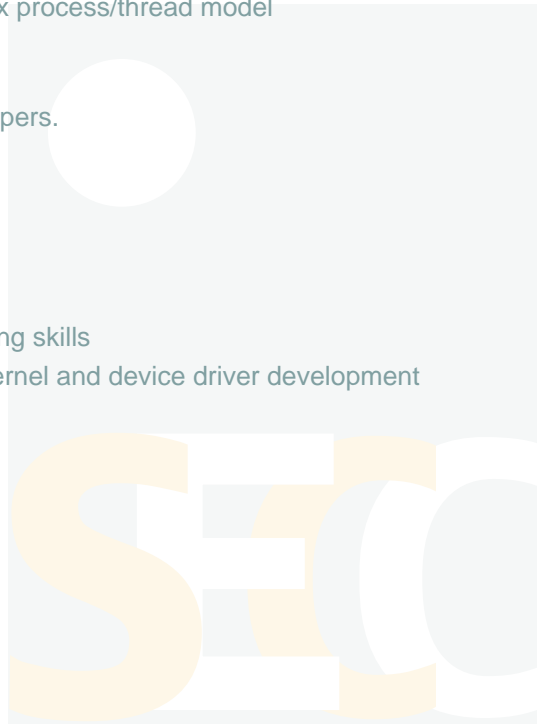
Embedded software developers.

### *Course Duration*

1 day

### *Prerequisite*

- Good C programming skills
- Embedded Linux kernel and device driver development



## Digital Design Using Verilog

### *Overview*

Hardware description languages are high-level programming languages used for modeling microelectronic circuits. They have the advantage of being portable and independent of technology allowing design modification and reuse. They also serve as a platform for several design tools such as design entry, design verification, timing analysis and verification, and synthesis. This workshop introduces Verilog as one of the most powerful hardware description languages.

### *Topics to be covered*

- Introduction to hardware description languages and design flow
- Introduction to Verilog
- Basic concepts of Verilog
- Modeling combinational and sequential logic for synthesis using Verilog
- Functional verification of Verilog models

### *Target Audience*

Embedded software developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Knowledge of fundamental programming language constructs

## Verilog's VPI Standard

### *Overview*

Verilog is one of the most powerful hardware description languages used for modeling. One of the main advantages of Verilog is its extensibility via the Verilog procedural interface (VPI) which is a collection of routines that allow foreign languages as C to access information contained in Verilog designs. This workshop explains the VPI standard and describes its various routines.

### *Topics to be covered*

- Introduction to VPI
- Using VPI to interface to Verilog simulators
- Dealing with objects and handles in a VPI application
- Interfacing Verilog simulations to C models using VPI routines

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of C language
- Good knowledge of Verilog and simulation

## FPGA Overview Based on Virtex™ FPGA

### *Overview*

The design and verification of integrated circuits is a very complex, expensive, and time-consuming task. Field programmable gate arrays (FPGAs) provide a hardware platform that facilitates and speeds up the verification of the hardware implementation for applications where time-to-market and project cost are important. This workshop demonstrates the FPGA features utilizing an evaluation board that employs a Virtex™ FPGA.

### *Topics to be covered*

- Introduction to FPGA
- FPGA programming elements
- FPGA logic cells
- Detailed view on Virtex™ FPGA
- Design flow for Virtex™ FPGA-based application
- Debugging and verification of Virtex™ FPGA-based application

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of Verilog and simulation

## HW/SW Co-Simulation for SoC Design and Verification

### *Overview*

The conventional design and verification methodologies of a system on chip (SoC) are very complex and time consuming. Software designers have to wait until hardware Engineers finish their job to start software design and testing. New design techniques are being developed to facilitate both hardware and software design of the SoC implementation to proceed concurrently saving both time and effort.

One of these techniques is hardware/software co-simulation that is covered in this workshop.

### *Topics to be covered*

- Introduction to SoC design environments
- Inter process communication (IPC) as one of most powerful co-simulation techniques

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of C language programming and compilation
- Good knowledge of Verilog-HDL and simulation
- Good knowledge of Verilog Procedural Interface (VPI)

## HW/SW Co-Emulation for SoC Design and Verification

### *Overview*

The conventional design and verification methodologies of a system on chip (SoC) are very complex and time consuming. Software designers have to wait until hardware Engineers finish their job to start software design and testing. New design techniques are being developed to facilitate both hardware and software design of the SoC implementation to proceed concurrently saving both time and effort. One of these techniques is hardware/software co-emulation where the hardware part will be modeled on a real hardware platform (FPGA) to speed up the verification process.

### *Topics to be covered*

- Introduction to SoC design environments
- Cycle-level emulation model
- Transaction-level emulation model

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of C language
- Good knowledge of Verilog and simulation

## Embedded System Co-Verification Methodologies

### *Overview*

Hardware/software co-verification is about making sure that embedded system software works well with the target hardware before chips and boards are made available. For applications where time-to-market and project cost are important, co-verification saves time and reduces the risk of costly hardware design errors. This workshop illustrates some of the co-verification methodologies used for that purpose.

### *Topics to be covered*

- Definition and benefits for embedded system co-verification
- Basics of various verification methodologies
  - Test bench-based verification
  - Equivalence checking using VPI routines
  - Using co-simulation and co-emulation in embedded system verification

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of C language
- Good knowledge of Verilog and simulation
- Good knowledge of Verilog procedural interface (VPI)

## HW/SW Co-Design for ARM-Based SoC

### *Overview*

ARM processor is one of the most widely-spread processors for embedded systems. This workshop explains how to build an ARM/AMBA-based system using bus functional models (BFMs) and ARM instruction set simulator (ISS) together with the design under test.

### *Topics to be covered*

- Introduction to SoC design environments
- Introduction to cross compiler and ARM ISS
- Co-simulation of both the ARM/AMBA-based system in Verilog and the application program in C
- Co-emulation of both the ARM/AMBA-based system mapped on FPGA and the application program in C

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic knowledge of digital design and computer architecture
- Good knowledge of C language
- Good knowledge of Verilog and simulation
- Basic knowledge of ARM/AMBA systems
- Basic knowledge of Verilog procedural interface (VPI)

## ARM Architecture

### *Overview*

ARM processor is the most popular 32-bit RISC processor. Its market share is approximately 75 percent. Learning the internals of ARM is essential to write efficient application code. It also helps the embedded system developer to select the ARM version most suitable to the application. This workshop provides many insights into ARM internals.

### *Topics to be covered*

- ARM processor versions and structure
- ARM memory system
- AMBA bus
- Understanding types and features of commercial ARM-based SoCs

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of C language
- Basic Knowledge of computer organization
- Basic Knowledge of compiler, linkers, and assemblers

## ARM Assembly

### *Overview*

ARM processor is the most popular 32-bit RISC processor. Its market share is approximately 75 percent. This workshop teaches the ARM assembly language. It explains the instruction set and efficient coding techniques. It also discusses how to write C code optimized for ARM-based systems.

### *Topics to be covered*

- ARM processor instruction set
- Software optimization
- Advanced programming techniques

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of C language
- Basic knowledge of assembly languages
- Basic Knowledge of ARM architecture
- Basic Knowledge of compiler, linkers, and assemblers

## ARM-Based Embedded System Development

### *Overview*

ARM processor is the most popular 32-bit RISC processor. Its market share is approximately 75 percent. Most ARM-based embedded systems share similar properties. This workshop explains the steps needed to make simple ARM-based embedded application.

### *Topics to be covered*

- Embedded system structure using the S3C2410 SoC as an example
- Introduction to different downloading and debugging techniques
- How to design system memory structure and scatter loading
- How to write boot code
- How to write device drivers

### *Target Audience*

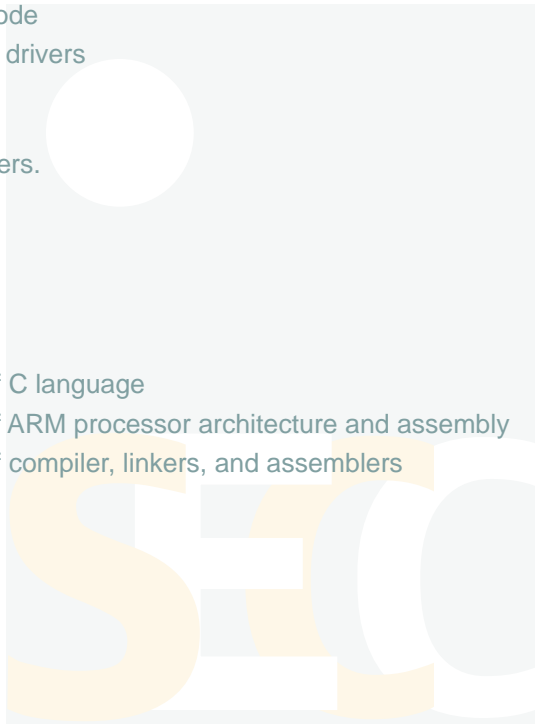
Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of C language
- Basic Knowledge of ARM processor architecture and assembly
- Basic Knowledge of compiler, linkers, and assemblers



## ARM-Based RTOS Porting

### *Overview*

An operating system is composed of hardware dependent and independent parts. Porting is about editing the hardware dependent part to be suitable for the hardware platform in concern. This workshop explains porting real-time operating systems on ARM-based platforms.

### *Topics to be covered*

- Introduction to porting
- Porting an RTOS on ARM-based hardware giving the VELOS RTOS as an example

### *Target Audience*

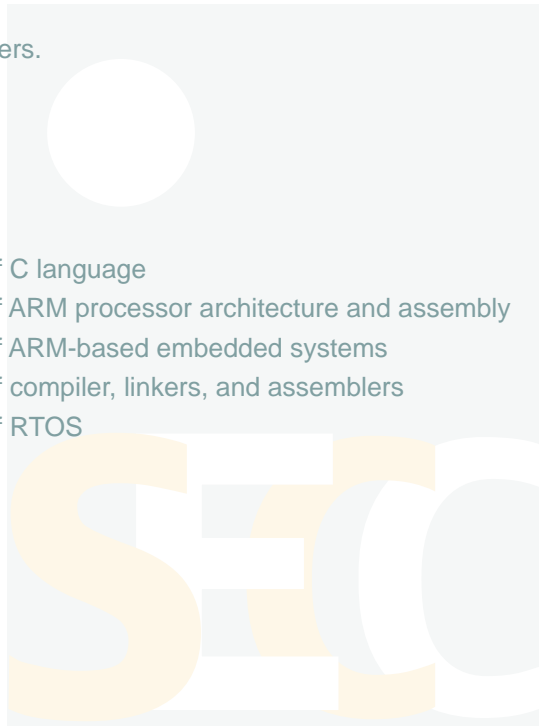
Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of C language
- Basic Knowledge of ARM processor architecture and assembly
- Basic Knowledge of ARM-based embedded systems
- Basic Knowledge of compiler, linkers, and assemblers
- Basic Knowledge of RTOS



## Embedded GUI

### *Overview*

Most consumer electronics nowadays contain small screens and are accessible through graphical user interfaces (GUIs). These small embedded systems have many restrictions on memory, power, and processing resources. In view of all this, writing an efficient embedded GUI is hence so important. This workshop explains concepts of embedded GUIs and how they differ from desktop ones. It utilizes the PEG (portable embedded GUI) library to discuss how to develop an embedded GUI application.

### *Topics to be covered*

- Introduction to embedded environment
- Introduction to embedded GUI concepts
- Introduction to PEG
- Writing a GUI using PEG

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Good Knowledge of C/C++ language
- Basic Knowledge of compiler, linkers, and assemblers

## Embedded Audio Processing

### *Overview*

Many consumer electronics provide multimedia applications, where image, audio, video and text data are extensively processed. Processing involves sampling, compressing, storing, and decoding. This workshop explains embedded audio processing basics and provides the requirements for developing an embedded audio application. A MP3 player is developed as an example.

### *Topics to be covered*

- Audio processing basics
- Requirements for an embedded audio application
- Audio device drivers
- Audio compression and audio codecs
- Case Study: MP3 player

### *Target Audience*

Embedded systems developers.

### *Course Duration*

1 day

### *Prerequisite*

- C language programming and compilation
- Basic knowledge of embedded systems and thread programming
- A basic knowledge of audio signal processing

## Embedded Video Processing

### *Overview*

Many consumer electronics provide multimedia applications, where image, audio, video and text data are extensively processed. Processing involves sampling, compressing, storing, and decoding. This workshop explains embedded audio processing basics and provides the requirements for developing an embedded audio application. An AVI player is developed as an example.

### *Topics to be covered*

- Video processing basics
- Requirements for an embedded video application
- Writing to an LCD
- Video compression and video codecs
- Case Study: AVI player

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- C language programming and compilation
- Basic knowledge of embedded systems and thread programming
- A basic knowledge of embedded audio processing

## Camera Support in Embedded Multimedia Systems

### *Overview*

Many consumer electronics provide multimedia applications, where image, audio, video and text data are extensively processed. Processing involves sampling, compressing, storing, and decoding. This workshop explains basics of camera support in embedded applications and provides the requirements for such support. A camera application is developed as an example.

### *Topics to be covered*

- Camera basics
- Requirements for camera support in an embedded application
- Camera controller and bus
- Camera functionality
- Case Study: camera application

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- C language programming and compilation
- Basic knowledge of embedded systems and thread programming
- A basic knowledge of the interrupt mechanism
- A basic knowledge of the different color spaces

## RTOS Basics

### *Overview*

An RTOS is a very key component in embedded systems development. Although, RTOSes may be composed of components similar to those of general purpose operating systems (GPOSes), they differ in their implementation so as to meet embedded and real-time requirements. The workshop explains RTOS essentials as scheduling, processes/threads, semaphores, and messaging. It discusses RTOS services as exceptions, interrupts handling, memory management, and I/O subsystem.

### *Topics to be covered*

- Introduction to RTOS
- RTOS Essentials
- Tasks
- Semaphores
- Messages
- RTOS Services
- Exceptions and interrupts handling
- Memory Management
- I/O subsystems
- Other services

### *Target Audience*

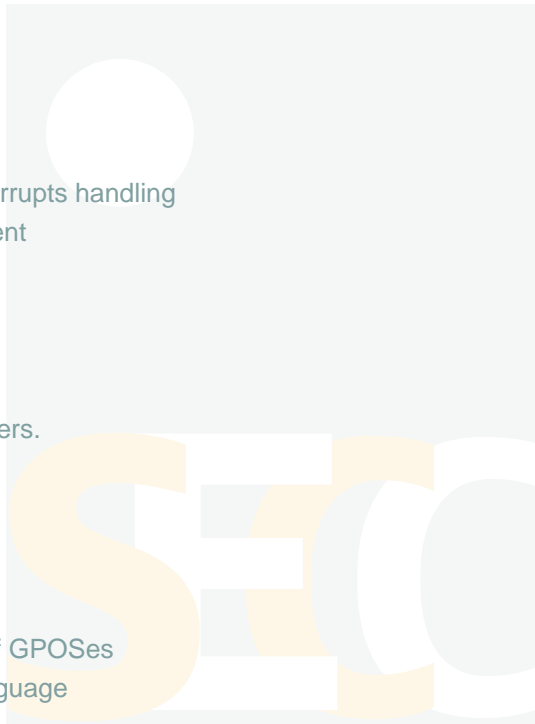
Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of GPOSes
- C programming language



## Multitask Programming in RTOS

### Overview

Writing everything in a single large program is not appropriate in many applications and in particular real-time (RT) ones. For RT applications, events run concurrently and in most cases independent of each other. A more appropriate approach is to handle each event in a separate task letting all of them running concurrently as in the real world. This workshop deals deeply with this issue through the explanation of multi-tasking, scheduling, thread model, synchronization, and messaging.

### Topics to be covered

- Multitasking
- Scheduling
- Threads life cycle
- Synchronization
  - Mutexes
  - Condition variables
  - Semaphores
- Messaging
  - Message queues
  - Mail boxes

### Target Audience

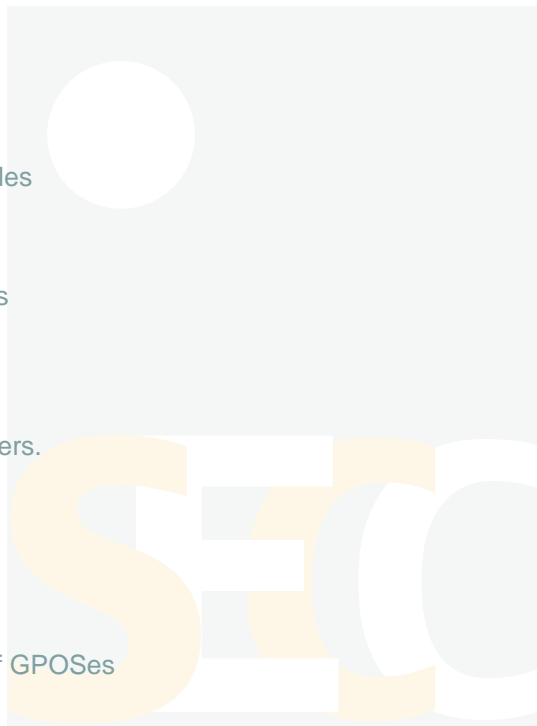
Embedded system developers.

### Course Duration

1 day

### Prerequisite

- Basic Knowledge of GPOSeS
- RTOS basics
- C programming language



## μC/OS-II the RTOS

### *Overview*

μC/OS-II is the one of the most popping-up RTOSes in the embedded development area. Its strength comes from fulfilling the requirements of many real-time applications in a very simple and efficient way. The workshop introduces the μC/OS-II, its kernel structure, and its real-time services.

### *Topics to be covered*

- Introduction to μC/OS-II
- μC/OS-II kernel structure
- Interrupts handling
- μC/OS-II objects & services
  - Task management
  - Time management
  - Semaphores management
  - Mutexes management
  - Message queues management
  - Message mailboxes management

### *Target Audience*

Embedded system developers.

### *Course Duration*

1 day

### *Prerequisite*

- Basic Knowledge of GPOSes
- RTOS basics
- C programming language

## Porting $\mu$ C/OS-II to ARM Processor

### Overview

One of the main advantages of an RTOS is its portability among different platforms. This is experienced in this workshop. This workshop explains porting concepts using as an example the porting of the  $\mu$ C/OS-II RTOS from x86 architecture to ARM-based architecture.

### Topics to be covered

- Introduction to porting
- Porting  $\mu$ C/OS-II
  - Development tools
  - Directories and files structure
  - Testing
- Case Study: Porting  $\mu$ C/OS-II to the S3C2410 SoC
  - Processor specific code
  - Application specific code
  - Console driver development
  - MMU port

### Target Audience

Embedded systems developers.

### Course Duration

1 day

### Prerequisite

- Basic knowledge of compilers, linkers, assemblers, and debuggers
- C programming language
- ARM architecture and assembly
- Basic knowledge of  $\mu$ C/OS-II

## Basics of Embedded Networking

### *Overview*

Embedded network systems provide a set of technologies that link the so widely-available consumer electronics to large scale networks. This link is essential for too many applications. This workshop explains basic concepts of embedded networking. It also utilizes socket programming to develop an example of embedded network applications.

### *Topics to be covered*

- What is embedded networking?
- Potential applications
- Socket programming
- Case study using the VELOS RTOS and the S3C2410 SoC

### *Target Audience*

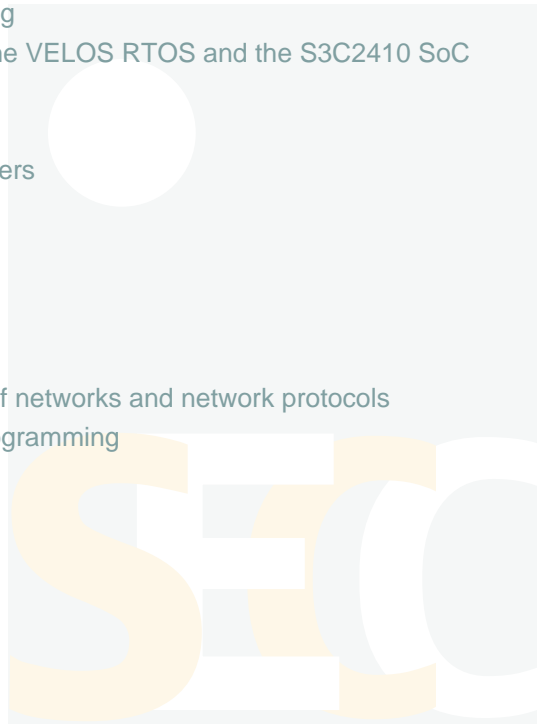
Embedded system developers

### *Course Duration*

1 day

### *Prerequisite*

- General concepts of networks and network protocols
- Knowledge of C programming



## Embedded Web Server

### Overview

Embedded web servers allow monitoring and controlling embedded applications using any standard browser. This workshop introduces their basic concepts and explains many of their implementation issues.

### Topics to be covered

- Introducing embedded internetworking
- Basic sockets networking
- Embedded web server
  - What is a web server?
  - Why embed a web server?
  - What is special about embedded web servers?
  - Key design goals of embedded web servers
- Introducing HTTP and HTML
- General web server process
- Case study: How to run Velos embedded web server?
- Labs
  - Demonstrate the development environment
  - Explore the HTTP transaction
  - Practice on how to add and link HTML pages
  - Develop a simple surveillance system

### Target Audience

Embedded systems developers

### Course Duration

1 day

### Prerequisite

- General concepts of networks:
  - OSI reference model
  - TCP/IP protocol suite
  - Network protocols
- Knowledge of C programming



## **L. Soft Skills**

*This track is designed to provide participants with information and skills allowing them to work effectively and improve their performance. Emphasis is given to presentation, communication, negotiation and report writing skills. Marketing, sales and training skills are also included.*

## Soft Skills

<b>L001</b>	Presentation Skills	138
<b>L002</b>	Communication Skills	139
<b>L003</b>	Business Writing Essentials	140
<b>L004</b>	Technical Writing Skills	141
<b>L005</b>	Negotiation Skills	142
<b>L006</b>	Marketing Essentials	143
<b>L007</b>	Essentials of Sales	144
<b>L008</b>	Sales Management Course	145
<b>L009</b>	Train the Trainer	146

## Effective Presentation Skills

### *Overview*

Effective presentation skills can enhance the likelihood of a successful outcome. In many business roles today, outstanding presentation skills have become a core competency for achieving key goals and influencing the decisions and actions of others.

This Course provides Participants with the skills required to present effectively to both groups and individuals and most importantly to develop their confidence which makes presenting so much easier and therefore more successful. The course covers comprehensively all the areas necessary to deliver presentations with real impact.

### *Topics to be covered*

- Introduction to high impact presentation
- Delivery skills
- First presentations
- Process of listening and understanding
- Planning your presentation
- Building your presentation
- Handling questions
- Presentation rehearsal

### *Target Audience*

Anyone starting to meet clients face-to-face, or in need of training in presentations, with much role-playing.

### *Course Duration*

3 days

### *Prerequisite*

None

## Communication Skills

### *Overview*

Communication is the art of successfully sharing meaningful information with people by means of an interchange of experience. Leaders wish to motivate the team members they work with and to provide them with information that will allow them to work effectively and improve performance. Communication from the leader to team members will initiate appropriate actions. This however, requires the team member to not only receive the information from the leader but also to understand and accept it.

The techniques learnt on this course will help the delegates to communicate effectively in a number of different environments. They will become more aware of how important good communication is and how situations can be affected by poor communication.

### *Topics to be covered*

- What is communication?
- Understanding the process and the barriers
- Communication styles
- Flexibility in communication
- Attentive listening
- Questioning skills
- Summarizing

### *Target Audience*

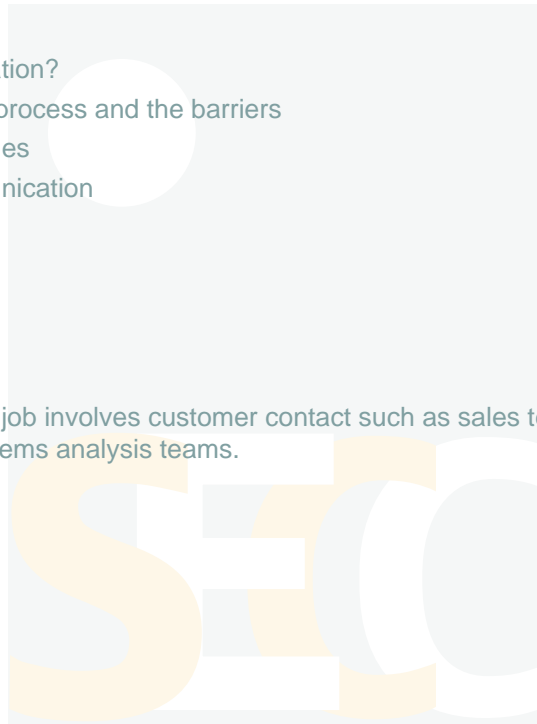
Anyone at any level whose job involves customer contact such as sales teams, marketing teams, customer services and systems analysis teams.

### *Course Duration*

2 days

### *Prerequisite*

None



## Business Writing Essentials

### *Overview*

The Business Writing Skills course teaches trainees how to write letters, e-mails, memos, and reports that people understand and take seriously. The course is designed to help trainees feel they can write a document tomorrow morning confident that it will be understood and generate the desired response.

### *Topics to be covered*

- Pre-write
- Prepare the information
- Organize the writing
- Introduce the content
- Write a clear document
- Write clear and complete explanations
- Write a conclusion with impact
- Write clear and effective sentences, paragraphs, and words
- Prepare a polished and correct final draft

### *Target Audience*

All office workers such as sales teams, marketing, and technical teams.

### *Course Duration*

2 days

### *Prerequisite*

Basic command of the English language

## Technical Writing Skills

### *Overview*

The Technical Writing course teaches business writers how to prepare letter reports and technical reports about subjects that require technical explanations, diagrams, charts, and jargon understood by technical readers. The course teaches the technical writer how to present technical information to technical readers so they understand the concepts and can apply them in their work. The course is not intended to explain technical subjects to non-technical readers.

### *Topics to be covered*

- The client and the audience
- Objectives and specifications for the technical report
- Methods of selecting information
- Formats for the major types of technical documents
- Presenting technical explanations clearly
- Organizing
- Using guideposts for clarity
- Synthesizing and organizing
- Writing clearly
- Using visual devices
- Complete sample report
- Editing
- Writing concisely
- Format and publishing
- Proofreading

### *Target Audience*

All technical writers or any one need to learn more about technical writing to improve his writing skills.

### *Course Duration*

2 days

### *Prerequisite*

Basic command of the English language

## Negotiation Skills

### *Overview*

This course is designed for personnel occupying Middle and Senior level positions in their organizations who are bound to confront a reasonable amount of negotiations. This hands-on course is designed for managers who feel that they have basic skills in negotiations or who realize the power of implementing the process of negotiation through their jobs. In many situations conflict and deadlocks are commonplace.

### *Topics to be covered*

- Understand and develop your communication skills (Guideline for Oral discussion, types and strategies for Listening, communication patterns, persuading people,...)
- Clearly understand the Negotiation process and manage its different BATNA phases and to obtain and implement detailed knowledge of negotiating and bargaining techniques
- Know the different types of negotiators (Novice, Competent and Complete) and how to deal with them
- Understand principled negotiations and positional bargaining, also know when to use "soft" or "hard" negotiations based on the 4 principles of negotiation)
- Learn how to apply the methods of principled win-win negotiation
- Clearly understand how to handle difficult negotiators (what if they are more powerful, what if they won't play, what if they use dirty tricks,...)
- Understand conflict and how to handle it (functional & dysfunctional conflict, conflict simulation techniques, how to overcome conflict)
- Know how to prepare for the negotiation (analyze audience, how to best arrange ideas, format for "yes" and "no" messages, 10 questions people ask about getting to "yes", Seating arrangement for negotiation, preparing your talk for negotiation, negotiation checklist preparation,...)

### *Target Audience*

Anyone starting to meet clients face-to-face, or in need of training in presentations, with much role-playing.

### *Course Duration*

3 days

### *Prerequisite*

Basic command of the English language

## Marketing Essentials

### *Overview*

This is an introductory course for marketing that focuses on the marketing of services and goods.

Through this course, participants will explore important marketing theories and concepts, and apply them to practical business situations.

### *Topics to be covered*

- Develop well-planned marketing strategies using the basics of marketing
- Demonstrate an understanding of the principles underlying the development of marketing strategies
- Understand the role of the marketer in contributing to the direction of an organization which offers services or products in order to satisfy the needs of customers
- Develop marketing research
- Evaluate customer needs
- Develop market segmentation
- Develop positioning plan
- Establish an effective marketing mix

### *Target Audience*

- New graduates or those new to the marketing field with under 3 years of marketing experience
- Individuals who have recently moved into positions with such responsibilities as sales and sales management, advertising, promotion, brand building, pricing, strategic planning, services, and product development

### *Course Duration*

2 days

### *Prerequisite*

None

## Essentials of Sales

### *Overview*

The Essentials of Sales course will enhance selling skills based on structured sales knowledge and techniques. Participants are engaged in a highly interactive series of exercises, role plays and case studies. Delegates will be provided with the opportunity for individual practice, coaching and development. In addition they will take away a series of checklists for preparing, conducting and analysing sales situations back in the field.

### *Topics to be covered*

- Understand the decision making process and the structure of selling
- Prepare for the sale using proven techniques and models
- Prospect for new business and manage their territory more effectively
- Communicate more clearly with the potential buyer
- Convert features to benefits and overcome doubts
- Handle the barriers to the sale, example doubts and objections, with more confidence

### *Target Audience*

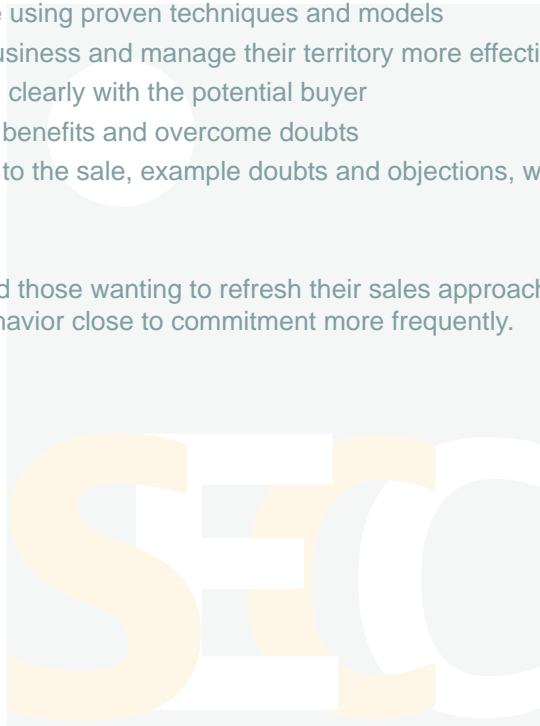
Both those new to sales and those wanting to refresh their sales approach and revise the fundamental principles of good sales behavior close to commitment more frequently.

### *Course Duration*

2 days

### *Prerequisite*

None



## Sales Management Course

### *Overview*

The Sales Management course is prepared to help the executive make an immediate impact. It will enable him or her to quickly assess the current position of the sales team, to rapidly enhance existing management skills, to quickly understand the essential responsibilities and tasks of a Sales Manager and to begin to practise those skills.

### *Topics to be covered*

- Lead, motivate and develop the sales team
- Train individual sales staff by using joint visits more productively
- Interview and select sales candidates more likely to succeed
- Run more effective meetings
- Manage their team by setting clearer objectives
- Identify the role adopted by individuals in a team and how teams develop
- Provide opportunities for individual practice, coaching and development

### *Target Audience*

Experienced sales managers, sales directors and all executives with 'front line responsibility' for sales, the sales force or the marketing manager who needs to acquire a better understanding of sales management.

### *Course Duration*

2 days

### *Prerequisite*

Essentials of Sales course or equivalent knowledge plus experience in sales management position

## Train the Trainer

### *Overview*

Train the trainer workshop provides the key knowledge and skills needed for new trainer. Good trainers are able to design and deliver effective knowledge, skills and attitudes development programmes in a positive and motivational way. Participants will learn methods to enable them to make an effective contribution to the training and development of others. Participants will receive specific feedback from the instructor and his peers and will leave with feedback sheets, and suggestions for improvement.

### *Topics to be covered*

- Learning process
- Learning management
- Learning styles and their implication
- Write training objectives
- Prepare and deliver the training
- Principles of learning
- Build fun and participation into your learning events
- Handling questions
- Training delivery rehearsal

### *Target Audience*

Trainer who have less than one year's experience in course delivery and who have no formal training in instructional design and development. This course also suitable for managers, facilitators, educators, course designers, HRD professionals and any one, who want to present, train and facilitate more effectively.

### *Course Duration*

3 days

### *Prerequisite*

No prerequisite, preferable having good communication skills



## M. Collective Track

*This track is a collection of different courses to help the fresh graduates and the developers, without enough background ; to have an intensive knowledge on these subjects, through providing them with the basics methodologies and tools necessary to be adapted in the overall software development within the software companies*



## M. Collective Track

Code	Course Name	Duration (Days)
<b>B</b>	<b>Software Development Methodologies Track</b>	
B001	Introduction to Software Development Methodologies	3
B002	Requirement Engineering	3
B003	Process Modeling: Concepts and Tools	3
B004	Data Modeling: Concepts and Tools	3
B005	Software Design (I)	3
B007	Software Quality Assurance and Testing	3
<b>C</b>	<b>Object Oriented Software Development</b>	
C001	Introduction to Object Orientation	2
C002	UML ( Unified Modeling Language )	3
C003	O.O. Analysis & Design	4
C004	O.O. Implementation	3
<b>L</b>	<b>Soft Skills Track</b>	
L001	Presentation Skills	3
L002	Communication Skills	2
L003	Business Writing Essentials	2
L004	Technical Writing Skills	3
L005	Negotiation Skills	3
L006	Marketing Essentials	2
L007	Essentials of Sales	2

*For more information about the contents of these courses , refer to the code number in the corresponding track.*