Course ID
SWENG1
Course Duration
2 days

Course Title
Principles of Software Engineering

Related Courses

- Software Engineering: An Advanced Tutorial (SWENG2, 3 days)
- Project Management Workshop (PROJMGT2D, 2 days)
- Project and Team Management Workshop (PROJMGT4D, 4 days)

Aimed At

Individuals involved in software development who need a broad understanding of the entire software development lifecycle, guiding principles, and techniques.

Group Size

5-25

Prerequisites

While there are no specific prerequisites, exposure to software development and/or project management will be very helpful.

Course in a Nutshell

Software engineering is far more than programming. It is a broad discipline that encompasses many important concepts and techniques.

This, the first of our two courses on software engineering, provides an introduction to the objectives, principles, and methodologies of software engineering. Upon the completion of this course, you will have an understanding of the entire software lifecycle from conception to development to maintenance.

Customize It!

- *Are you a project manager, analyst, programmer, testing/debugging, or integration specialist and* need a better understanding of how your work fits into the software lifecycle? We can emphasize the phases relevant to your particular needs while also trying to give you a broad understanding of the full lifecycle.
- *Are you interested in particular techniques of software development?* If so, we can emphasize those parts of this course that focus on the areas and tools pertinent to your project or product.
- *Are you involved in the marketing, sales, or post-sales support of software* and need a better understanding of how your job fits into the overall software lifecycle? We can customize the course to focus on those issues.

Learn How To

Describe and apply the key concepts and techniques for:

- Software engineering principles and objectives
- The end-to-end software lifecycle
- Software inception.
- Requirements elicitation and definition.
- Design and analysis.
- Prototyping and user interfaces.
- Coding and testing.
- Deployment and maintenance.

Course Outline

- Introduction
  - Why Study Software Engineering?
  - History
- Life Cycle Models
  - Software Project Life Cycle Models
  - Software Project Life Cycle Processes
  - Model Descriptions: Waterfall, Component-based (Reuse), Evolutionary, Spiral, Object Oriented
- Objectives and Principles
- Types of Interfaces
- Project Initiation
  - Description of the Business Needs/Problem to Be Solved, Objective(s), and Scope
  - Feasibility Studies
  - Estimating
- Requirements Definition
  - Stakeholders, Buy-in
  - Requirements Elicitation, Interviewing, and Analysis
  - Object Oriented Analysis
  - Functional and Non-functional Requirements
  - User Interfaces
  - Prototyping
- Design
  - Architectural Design
  - Object Oriented Analysis
  - Design Patterns
- Development and Testing
  - Coding Standards
  - Types of Testing

- Deployment and Maintenance
- Wrap-up
  - Course Recap
  - Preview of Advanced Topics
  - Q/A and Evaluation

How You Will Learn

- A seasoned software engineer/instructor will present this course in interactive lecture format
- Along with lecture, we use exercises, puzzles, case studies, and interesting group activities to enrich the instruction and drive home the essential points.
- If you already know something about software engineering, we build on that.
- If your background is less technical, we use examples and analogies to simplify the complex subject matter.
- You will receive a printed Participant Handbook, which will help you remember and retain what you learned in class and apply it on your job. This will also include useful web links to equip you with additional resources for future reference and study.

Revised *February 28, 2007*