

7 Low Overhead Software Process Improvements

Course Summary

Description

Software process improvement doesn't have to be synonymous with expensive formalized approaches, such as the Software Engineering Institute's Capability Maturity Model (SEI CMM). This interactive seminar workshop describes it along with seven alternative approaches that can provide significant software productivity and quality improvements without extensive bureaucracy or organization-wide cultural change. Exercises enhance learning by allowing participants to practice applying practical techniques to realistic examples.

Objectives

At the end of this course, students will be able to:

- Recognize real processes and distinguish them from presumed processes.
- Avoid common traps that lead to making only illusory improvements.
- Analyze and measure processes to identify meaningful improvements.
- Use 7 specific methods to quickly and economically improve software processes.

Topics

- "Real" vs. "Presumed" Processes
- High-Overhead Approaches
- If You Don't Know What You're Doing, You Don't Know What You're Doing
- Powerful Improvement Practices

Audience

This course has been designed for managers, analysts, designers, programmers, testers, auditors, and users who are concerned about the efficiency and effectiveness of software development and support.

Prerequisites

There are no prerequisites required for this course.

Duration

One day

7 Low Overhead Software Process Improvements

Course Outline

I. "Real" vs. "Presumed" Processes

- A. Your process, issues, improvement efforts
- B. What a process is and why we care
- C. Relation between process and results
- D. The only way to change your results
- E. Why most process improvements fail
- F. Distinguishing "real" from "presumed"
- G. When the real process is not recognized
- H. Defined and documented processes
- I. Silos, stovepipes, and smokestacks
- J. Measuring a process to its full end result
- K. Real vs. presumed testing process
- L. Overcoming unwise conventional wisdom

II. High-Overhead Approaches

- A. Process capability
- B. Capability Maturity Models
- C. Benefits and advantages
- D. Stepwise vs. continuous models
- E. Proliferations, e.g., Testing Maturity
- F. Activity vs. results
- G. Empirical analysis of actual improvement
- H. Piece of paper mentality
- I. Process imposition vs. process improvement
- J. Role of management styles and practices
- K. Six Sigma as applied to software
- L. Strengths and weaknesses
- M. Identifying appropriate measures
- N. Religious approaches vs. real improvement
- O. Management gaps

III. If You Don't Know What You're Doing, You Don't Know What You're Doing

- A. Evolutionary vs. revolutionary improvement
- B. Key perspective to identify the real process
- C. How to measure a process
- D. Multivariate process mapping
- E. Analyzing handoffs and bottlenecks
- F. Evaluating value added
- G. Streamlining and eliminating error sources
- H. Measurement and metrics
- I. Avoiding main causes of resistance
- J. Implementing effective metrics baselines
- K. Fallacies of on-time and in-budget
- L. Non-operational "soft" components
- M. The three envelopes
- N. Emperor's new clothes risk
- O. Measuring and improving people processes
- P. "They won't let us use this training"

IV. Powerful Improvement Practices

- A. Making good practices work
- B. Stop turning inadequacies into virtues
- C. REAL, business requirements
- D. System/software requirements
- E. Use cases
- F. Problem Pyramid™ tool
- G. Effective reuse
- H. Reviews and inspections
- I. Meaningful early involvement
- J. Proactive Testing™
- K. Letting testing drive development
- L. Scaling risk