

# Practical Estimation: Getting the Most Value From Your Estimation Practices

## **Duration**

2 days

## **Instructor**

Geoff Hewson

## **Class Limit**

20 students

## **Prerequisite**

None

## **Price**

On-site

Please contact SPC for pricing (contact information on page 2)

Public Training

\$1095 (2 days)

\*Discount available for early registration

## **Materials Provided**

- Student manual containing the course slides
- Student handouts with class exercises and class studies

The ability to create reliable estimates is essential to the success of any software project. It's the basis upon which successful software organizations make sound individual and project commitments. Yet, software organizations find this important activity challenging. In a recent Cutter research survey, 40% of companies surveyed reported poor software schedule and budget estimation, while only 14% reported good performance.

What is it that the 14% do right and the other 40% do wrong? Effective software estimation is much more than simply creating accurate technical estimates of proposed work. It allows you to make good commitments for the individual, project, or business by working successfully with management and customers. It also empowers you to realistically align these commitments with business targets. With an effective estimate, you are able to focus the whole organization's attention on the opportunities and risks in a project.

*Practical Estimation: Getting the Most Value from your Estimation Practices* is a highly interactive two-day seminar focusing on how to create reliable estimates for whole projects, individual product features, and work tasks using a variety of estimation techniques.

You will learn how to use these techniques to refine your estimates as you proceed through the planning activities of the software lifecycle. Participants will complete the seminar having the skills and knowledge to arrive at more realistic and effective estimates that individuals, teams and business partners can rely upon to make achievable work commitments.

Discussion topics include:

- How estimation fits in to the process of making commitments between the software team and the business
- How estimation strategies are shaped by why the estimate is needed
- The characteristics of good estimates and good estimation
- A framework for successful estimation
- Guidelines for when and how to use different estimation techniques
- How to improve the quality of your estimates
- Effective strategies for presenting and negotiating your estimates
- How to build up your own capability for repeatable estimation

## **Intended Audience**

This seminar is ideal for senior management, project directors, project managers, program managers, software developers, engineers, project leaders, product managers, sales and marketing executives, and senior engineers preparing to take on a leadership role.

## **Instructor**

As President and Chief Knowledge Officer of the SPC, Geoff Hewson champions the company's mission of enabling software organizations throughout the world to realize the full business potential of their software development processes, technologies and teams. Geoff also oversees the service delivery division of SPC and drives overall thought leadership behind SPC's service offerings and research.

During his ten years with SPC, Geoff has directed numerous projects, where his interests have covered most aspects of software development. Geoff's 20+ years of IT experience has also included senior positions with Microsoft, Ford Motor and the Canadian Federal Government. Geoff holds a Ph.D. in Physical Chemistry from the University of Sheffield, UK.

TRAINING

# Practical Estimation: Getting the Most Value From Your Estimation

## Estimation in Context

- A brief history of estimation
- The role of estimation
- Differentiate between Estimate, Target, and Commitment
- Motivations for estimation depend on your role in the project
- Estimation strategy depends on who you are and why you need the estimate
- What is a good estimation and why is it desirable
- Characteristics of good estimates
- Characteristics of good estimation

## Strategies for estimating

- The analytical flow of creating an estimate
- Introduction to the various estimation approaches
- Illustrate how estimates are used through the software lifecycle
- Historical data calibration as a key factor in creating good estimates
- Structure and Counting leads to different Estimation Approaches
  - Top-down vs. Bottom-up estimating
  - Estimation by analogy
  - Requirements-based estimating
  - Design/Architecture-based estimating
  - Estimating from Work Breakdown Structure
- Techniques you use depend on where you are in the software lifecycle and what you are trying to do
- Compare and Calculate
  - Effort-based & Size-based
  - Importance of historical data and calibration
- Introduce purpose of Negotiate & Commit

## Estimating Project Tasks with Work Breakdown Structures

- Reasons for estimating the WBS
- Introduce and define WBS
- Structure & level decomposition depends on the development strategy
- Examples of common software project WBS structures
  - Project life cycle
  - Software life cycle
  - Requirements decomposition
  - Functional team specializations
  - Feature team specializations
- Explain the bottom-level work packages
- How to estimate the tasks (Delphi, PROBE, historical comparison)
- Building up the estimate for the whole project

## Early Project Estimates

- The reason for early project estimates
- Early project estimating approaches
  - Analogy – comparison with past projects
  - Expert Judgment & Wide-Band Delphi
- Historical data – what to use and what to collect
- Next steps – setting targets, determining feasibility

## Estimating from Requirements and Designs

- Reasons for estimating from product details
- The Law of Large Numbers
- Two general strategies: estimating from requirements or from architecture/design
- Estimating from requirements
  - Benefits of requirements-based estimates
  - Approaches for structuring requirements-based estimates:
    - Estimating the individual requirements “elements”
    - The Standard Component method
    - Building the Bottom-up project estimate
    - Estimated Tasks vs. Allocated Tasks
    - Historical data to collect & use
- Estimating from architecture/design
  - Benefits of Architecture-based estimates
  - Approaches for structuring architecture/design-based estimates:
    - Estimating the design components
    - Proxy-based estimation
    - Estimating component size
    - Building the bottom-up project estimate
    - Historical data to collect and use
- Next steps – what-if analysis and trade-offs vs. business targets

## Presenting and Negotiating Your Estimates

- What are we trying to achieve by negotiating the estimate?
- What happens with poor negotiation skills
- Approaches for presenting your estimate
- Principled Negotiation – Getting To Yes
  - Separate the people from the problem
  - Focus on interests vs. positions
  - Resolving the gap – options & strategies
- Good negotiating strategy example

## Building Your Own Estimation Capability

- Factors for repeatable estimation success
- Building a personal estimation process
- Building an estimation process for your organization
- The NASA SEL estimation procedure
- Building a historical database
- Goal-Question-Metric approach for gathering metrics
- Implications of project tracking and close-out
- Strategies for maintaining and distributing historical data for use



TRAINING