ProTech

protechtraining.com

"Charting the Course ...

... to Your Success!"

Software Estimating

Course Summary

Description

Unreliable estimates are a major reason many software projects are late, over-budget, and poor quality. Historically, estimating has been so weak in IT that some people simply assume it is impossible to estimate IT activities accurately, which in turn can become a self-fulfilling prophecy. In fact, though, a number of methods and approaches enable more accurate estimates, which can produce huge direct and indirect benefits. This interactive seminar describes key causes of estimating pitfalls and effective estimating concepts and techniques to overcome the difficulties. Methods address estimating not only coding but also other project components in both agile and more traditional projects. And, rather than just being a static up-front exercise, the course shows dynamic techniques that effective estimators use throughout the project to control progress as well as to refine and improve their estimates and estimating skills. 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:

- Estimation and control roles, issues, and impacts on project success.
- Appropriate uses and limitations of rapid top-down and parametric estimates.
- Work breakdown structure bottom-up estimating techniques and issues.
- Point counting and related size estimation techniques.
- Estimating schedules and resources.
- Separately estimating testing, importance and techniques.
- Controlling activities and refining estimates throughout the project.

Topics

- Estimating And Project Success
- Estimation Concepts, Top-Down
- Bottom-Up Estimating
- Agile Project Estimating
- Point-Counting Estimates
- Test-Based Task Estimates
- Other Testing Tasks To Estimate
- Estimating The Schedule
- Re-Estimating During The Project

Audience

This course has been designed for project managers, analysts, designers, programmers, QA/testers auditors, and others who need to estimate software and software projects.

Prerequisites

There are no prerequisites for this course.

Duration

Two days

ProTech

protechtraining.com

"Charting the Course ...

... to Your Success!"

Software Estimating

Course Outline

I. Estimating And Project Success
- A. Project management, development lifecycles
- B. Why estimates destine most projects to fail
- C. Impediments to influencing estimates
- D. Game-playing, countering Parkinson's Law
- E. Real causes of scope creep
- F. Self-fulfilling self-defeating prophecies
- G. Defining scope that doesn't creep so much
- H. Problem Pyramid™ disciplined definition
- I. Requirements negotiation model
- J. Deliverable value-based estimating
- K. Product backlog similarities and differences

II. Estimation Concepts, Top-Down
- A. "Knowing" estimating is impossible
- B. How effective estimators differ
- C. All estimates relate reference to target
- D. Similarity, scalability, history
- E. Top-down estimating, advantages
- F. Main reasons top-down often is inaccurate
- G. Top position relevance for top-down
- H. Rule-of-thumb techniques, traps
- I. Wide-band Delphi
- J. Code sizing-based project estimates
- K. Parametric estimating algorithms
- L. Factors affecting parametric accuracy

III. Bottom-Up Estimating
- A. Need to identify project activities regardless
- B. Major reason estimates are inaccurate
- C. Work-breakdown structure technique
- D. Level-by-level increase in precision
- E. Near- and far-term detail differences
- F. Identifying effort by resource and skill level
- G. Implicit vs. explicit duration
- H. Work packet roll-up
- I. Addressing contingencies and oversights
- J. Relating to top-down estimates, fudge factor
- K. Adjusting for skill level
- L. Dealing with risk and uncertainty
- M. Theory of Constraints, buffers
- N. Life cycle and other ways to identify tasks

IV. Agile Project Estimating
- A. Agile project concept differences
- B. Determining sprints and backlogs
- C. Defining user stories
- D. Conversations to elaborate user stories
- E. Story point sizing
- F. Planning poker
- G. T-shirt sizing
- H. Timeboxing vs. estimating
- I. Refactoring

V. Point-Counting Estimates
- A. Lines of code sizing issues
- B. Function point counting concepts
- C. Reliability
- D. Identifying types of functions
- E. Categorizing complexity
- F. Calculating unadjusted function points
- G. Determining adjustment factor
- H. Calibrating to effort
- I. Productivity variables
- J. Individual differences
- K. Applicability to agile development
- L. Use case point counting
- M. Defining consistent use case detail level
- N. Accounting for supplemental specifications

VI. Test-Based Task Estimates
- A. Importance of separately estimating testing
- B. Issues estimating testing based on code size
- C. What is a test case
- D. Use case scenarios
- E. White box structural control flow tests
- F. How many tests do we need, quality levels
- G. Historical defect statistics, precision
- H. Risk-based testing
- I. Estimating execution effort, duration
- J. Sizing test documentation
- K. Creating test data
- L. Analyzing and reporting results
- M. Defect isolation
- N. Defect advocacy
- O. Test set-up and tear-down time
- P. Factoring in defect, fix, and bad fix rates
- Q. Determining test cycles, regression tests
- R. Exploratory and ad hoc testing
- S. Test automation factors and issues
- T. Unique factors for each type of special test
- U. Relevance to testing in agile development

Software Estimating

Course Outline (cont'd)

VII. Other Testing Tasks To Estimate
- A. Establishing the test environment, lab
- B. Data creation, loading, extraction, updating
- C. Receiving and installing software
- D. Acquiring automated testing tools
- E. Getting training
- F. Anticipating and addressing delays
- G. Administrative demands on TM, Test Team
- H. Recruiting and hiring staff
- I. Coordinating with developers and users
- J. Manual vs. automated productivity
- K. Applying inspections, higher-yield methods

VIII. Estimating The Schedule
- A. Separating task definition from scheduling
- B. Productive time scheduling practicalities
- C. PERT and weighted averages risk reduction
- D. Gantt charts and resource leveling
- E. Concurrencies and dependencies
- F. Dependency networks
- G. Dependency network diagramming
- H. Critical Path and Critical Chain
- I. Value of slack
- J. Risk reserves, contingencies, and Plan B
- K. Constraints, imposed dates
- L. Identifying and addressing resource conflicts
- M. Fast tracking, schedule compression
- N. Resource-constrained project issues

IX. Re-Estimating During The Project
- A. Recording actuals against estimates
- B. Identifying systematic estimating errors
- C. Earned value measure of completion
- D. Relation of defects to project control
- E. Defect categorization, trends
- F. Projecting defects remaining
- G. Release criteria
- H. Monitoring and projecting arrival rate
- I. Identifying and attacking bug colonies
- J. Automated PM tools suitability, warnings
- K. Metrics for communication
- L. Metrics for control and improvement
- M. Reporting to management, key to influence