ProTech *"Charting the Course ...*

protechtraining.com

... to Your Success!"

Proactive Project Quality Management

Course Summary

Description

Many organizations mistakenly think managing project quality (PQM) means creating and enforcing "traffic cop" compliance with well-intentioned but often onerous procedural busywork. Many also equate QA with tail-end quality control (QC) testing, catching errors right before they go out the door when they are most expensive and risky to fix. Effective organizations realize PQM/QA can and should do far more, contributing positively to assure that the project process in fact better produces high quality in the first place, as well as catching and preventing errors early, which is the secret to delivering quicker and cheaper. It's not easy though—up to two-thirds of classic QA groups fail. Drawing upon concepts that influenced the totally-revised IEEE Std. 730 for Software Quality Assurance, this interactive workshop explains the six functions PQM/QA should perform that provide far greater value, analyzes why such groups so frequently have failed especially in IS, and presents practical approaches for successfully using PQM/QA effectively to produce high quality systems. Concepts emphasize software quality but apply to all types of projects. Exercises reinforce learning.

Objectives

After taking this course, students will be able to:

- Concepts, roles, and relation of PQM and QA for producing and assuring quality projects.
- Reasons for QA failures and factors critical to success of QA, especially in IS development
- The six functions that Proactive Project Quality Assurance™ should perform.
- Proven methods for managing quality and preventing errors throughout the life cycle.
- A structured Proactive Testing* model of which static and dynamic testing activities should be performed when and by whom within the life cycle to maximize testing efficiency and effectiveness.
- Measuring project quality and effectiveness of QA/Testing and PQM.

Topics

- Quality and Quality Assurance
- Project Processes
- Quality Models
- Life Cycle Practices and Quality
- Quality Assurance Concepts
- Requirements Role in Quality
- Discovering Real Requirements
- Proactive Testing Overview
- Active Static Testing
- Dynamic Testing
- Managing Project Quality

Audience

This course has been designed for project, systems, and business managers, as well as analysts, developers, engineers, quality specialists, auditors, and others responsible for the quality of projects and their results.

Prerequisites

There are no prerequisites for this course.

Duration

Three days

ProTech protechtraining.com *"Charting the Course ... ... to Your Success!"*

Proactive Project Quality Management

Course Outline

I. Quality and Quality Assurance
- A. Exercise: What is quality, quality assurance
- B. Quality in the project manager's triangle
- C. What gurus, we, others mean by quality
- D. Quality is free, cost of poor quality
- E. Need for positive common quality definition
- F. Quality factors and quality dimensions
- G. Engineered Deliverable Quality
- H. Quality assurance vs. quality control
- I. SQA in IEEE Stds. 12207 and 730
- J. Proactive SQA changes in IEEE Std. 730
- K. Not just 'traffic cop' compliance

II. Project Processes
- A. Exercise: Your project process
- B. REAL vs. Presumed processes, silos
- C. Exercise: Your REAL project process
- D. Defect injection, detection, ejection metrics
- E. Economics of quality problems in life cycle
- F. Making the business case for QA
- G. Life cycle concepts, waterfall vs. iterative
- H. Process capability, variation, improvement
- I. Project, process, product measures
- J. Direct and indirect process evaluation
- K. SEI Process Capability Maturity Model

III. Quality Models
- A. Capability maturity models
- B. CMM and CMMI
- C. Discipline-specific models
- D. SW-TMM, Test Process Improvement (TPI)
- E. Six Sigma, DMAIC
- F. Lean software practices
- G. Agile development and quality
- H. Exercise: Using quality models

IV. Life Cycle Practices and Quality
- A. Life cycle concepts, waterfall vs. iterative
- B. Requirements development
- C. Requirements management
- D. Software design
- E. Software coding quality
- F. Traditional reactive software testing
- G. Reviews and inspections
- H. Defect tracking and categorization
- I. Defect analysis and reporting
- J. Configuration management and control
- K. Release management
- L. Exercise: Your life cycle(s) and quality

V. Quality Assurance Concepts
- A. Why SQA groups so often fail
- B. Common definitions of SQA, issues
- C. QA vs. QC/Testing distinctions
- D. "Traffic cop" mentality value and resistance
- E. Verification vs. validation, usefulness
- F. Proactive SQA™ for effectiveness
- G. 6 functions of effective project QA
- H. QA Plans, quality reviews of deliverables
- I. Engineering standards, conventions
- J. Quality controls at all key points
- K. Recordkeeping and auditing
- L. Metrics and analysis for improvement
- M. Promoting awareness and recognition
- N. Relation of QA to QM
- O. Quality Manager role, skill set

VI. Requirements Role in Quality
- A. How requirements produce value
- B. Sources and economics of system errors
- C. Survey on improving requirements quality
- D. Business analysis role inconsistent attention
- E. Project management model misses key BA
- F. Business analyst vs. project manager roles
- G. System development model BA conflicts
- H. BABOKÒ model Knowledge Areas, issues
- I. Business vs. product/system requirements
- J. Common erroneous requirements beliefs
- K. A different, better model

Proactive Project Quality Management

Course Outline (cont'd)

L. Integrating business and analysis
M. Requirements process overview
N. Exercise: Review BA's requirements

VII. Discovering Real Requirements
A. Discovering, not gathering, requirements
B. Quantifying value
C. Requirements are the should be process
D. Horizontal processes and vertical silos
E. The "we don't have time" fallacy
F. Problem Pyramid™ tool to get on track
G. Exercise: Your Problem Pyramid™
H. Exercise: Applying Evaluation Guidelines
I. Aligning strategy, management, operations
J. Technology requirements vs. design
K. Management/supervisor vs. worker views
L. Who should define requirements
M. Exercise: Review BA's recommendation
N. Exercise: Identifying the REAL problem

VIII. Proactive Testing Overview
A. Testing for correctness vs. testing for errors
B. Developer views of testing
C. Reactive testing—out of time, but not tests
D. Proactive Testing Life Cycle model
E. CAT-Scan Approach to find more errors
F. Dynamic, passive and active static testing
G. V-model and objectives of each test level
H. Developer vs. independent test group testing
I. Strategy—create fewer errors, catch more
J. Four keys to effective testing
K. Need for testing sampling
L. Written vs. not written benefits and issues
M. Test activities that save the developer's time
N. The "we don't have time" fallacy

IX. Active Static Testing
A. Often overlooked walkthrough limitations
B. Formal technical reviews, procedures
C. Why reviews so economically find defects
D. Why review of requirements fails
E. Exercise: Reviewing Requirements
F. Unrecognized weaknesses of "Regular Way"
G. Evaluating requirements form, testability
H. REAL, business vs. system requirements
I. Finding overlooked, incorrect requirements
J. Reviewing design suitability and content
K. Four powerful design review CAT-Scans

X. Dynamic Testing
A. IEEE Std. 829 test plans and designs
B. Reactive and proactive risk analysis
C. Test design techniques, checklists, guidelines
D. IEEE Std. 1008 plan, design, implement
E. Technical/development tests vs. UAT
F. Test execution
G. Test automation types, issues

XI. Managing Project Quality
A. Defect reports that prompt suitable action
B. Common measures of test status, issues
C. Exercise: Test status report audiences
D. Projecting when software is good enough
E. Exercise: Measuring testing effectiveness
F. Where project quality management fits in
G. Need for continuing extra-project status
H. Exercise: Post-Implementation Review