Acceptance Testing For Users

GTC East
Albany, NY
26 Sep 2007

Kathleen Costello, PMP, SWPM
KC Consulting
Acknowledgements

- **Quality Software Project Management**
  by:
  - Robert T. Futrell
  - Donald F. Shafer
  - Linda I. Shafer

- **Software Project Management Program**
  - University of Texas At Austin,
    College of Engineering Center for Lifelong Engineering Education,
    Software Quality Institute
My Background

- Native New Yorker - born and raised in the Mohawk and Schoharie Valley area
- Bachelor of Science Degree in Management Information Systems from University of South Florida
- Certified Project Management Professional (PMP) through the Project Management Institute (PMI)
- Certified Software Project Manager (SWPM) through the University of Texas College of Engineering, Software Quality Institute
- SWPM Instructor and Mentor at University of Texas
  Hook ‘em Horns!
- Almost 20 years experience as an employee and consultant in Texas State Government projects
Today’s Agenda

- Introduction to Software Quality
- Requirements and Test Criteria
- Test Strategy and Planning
- Test Execution
- Acceptance and Beyond
Importance of Software Quality

Why do we need to manage projects toward good software quality?
Some Reasons for Project Trouble

- Poor Planning – Unrealistic Costs/Schedule
- Ill Defined Charter Or Contract
- Unstable Problem Definition
- Poor Communications
- Poor Requirements Understanding
- Inexperienced Project Management
- Unrealistic Expectations
- Ineffective/Non-existent Change Control
- Political Pressure
- Shortage Of Skilled People
- . . .
The Real Quality Cost Issues

- Customer Or Problem Reports
- Lawsuits
- QA & Test Department Costs
- Excessive Turnover
- Poor Teamwork
- Lack Of Planning
- Lack Of Business Understanding
- Poor Problem Handling
- Stress Due To Constant Overtime
- Lost Opportunities
- Lack Of Good Practices & Standards
- Ineffective Project Performance
- Poor/Ineffective Customer Relations
A project strives to deliver a product of a given scope, within a given cost and schedule, with a certain degree of quality.

“We want it Fast, Good and Cheap!”

Quality is *built in* into a software product through the implementation of software engineering methodologies and software process improvements.
Fundamental Teachings on Quality

● The GURUs of Quality
  ◆ Walter Shewhart
  ◆ Joseph Juran
  ◆ Phillip Crosby

● Quality Programs
  ◆ TQM
  ◆ Business Process Reengineering
  ◆ Software Quality Assurance
  ◆ Capability Maturity Model
  ◆ Six Sigma
Walter Shewhart

- “Father of statistical quality control” - Created statistical methods for quality control in 1930’s while at Bell Labs
- Today we call the whole set SPC (Statistical Process Control)
- Invented the Plan-Do-Check-Act cycle of process improvement
- Divided process variability into:
  - common causes (normal)
  - special causes (abnormal)

Walter Shewhart Biography at: http://www.asq.org/about/history/shewhart.html
Joseph Juran’s Contributions

- Known for bringing the Pareto Principle into use, coining the terms “trivial many” and “vital few”

- Also known as the “80/20” Rule
  - the vital few weak elements of a product account for cost of the errors
  - formalized into Pareto charts (ordered histograms) separate the vital few from the trivial many.
Joseph Juran’s Contributions

● On Quality:
  ◆ Said quality was “fitness for use”, not just “conformance to specifications”

● Promoted:
  ◆ upper management leadership in quality improvement
  ◆ quality training for all
  ◆ structured annual improvement plans
  ◆ teams examine defects, form theories, test theories, install remedial action
  ◆ some defects are worker-controllable, some are management-controllable; the latter are often the harder ones to fix
Philip Crosby’s Contributions

- Leader of Total Quality Management (TQM)
- Quality Director at ITT
- Authored “Quality is Free” 1979
  - drew lots of attention to poor quality in U.S. products
Philip Crosby’s Four Absolutes

1. Quality means **conformance to requirements**
2. Quality comes from **prevention** - based on training, discipline, example, leadership
3. Quality performance standard is **Zero Defects** - don’t tolerate “acceptable quality levels” - “Do It Right the First Time” (DIRTFT)
4. Quality measurement is the **Price of Nonconformance**
   - manufacturing companies spend about 25% doing things wrong
   - service companies spend about 40% on wasteful actions

Quality Focus

Total Quality Management Concepts applied to Software Project Process Management

- Sort out the many Quality Mgmt labels
- Apply key ideas from:
  - Walter Shewhart
  - W. Edwards Deming
  - Joseph Juran
  - Phil Crosby
- Use Assessments of Quality
  - ISO9000
  - SEI CMM (CMMI)
  - MBNQA (Malcolm Baldrige)
Quality Labels - What do they mean?

- **TQM** - Total Quality Management
- **BPR** - Business Process Reengineering
- **FURPSI** - Functionality, Usability, Reliability, Performance, Supportability, Integrateability
- **CQI** - Continuous Quality Improvement
- **CPI** - Continuous Process Improvement
- **TCS** - Total Customer Satisfaction
- **QMS** - Quality Management System
- **Hoshin** - Japanese term for strategic quality plan (literally, policy deployment)
More Quality Labels—
What do they mean?

- VoC - Voice of the Customer
- QFD - Quality Function Deployment
- SPC - Statistical Process Control
- SEI - Software Engineering Institute
- MBNQA - Malcolm Baldrige National Quality Award
- ISO - International Organization for Standardization
- CAT, ET, QIT, PIT, PPT - Corrective Action Team, Effectiveness Team, Quality Improvement Team, Process Improvement Team, Process Perfection Team
- Six Sigma - DMAIC = Define, Measure, Analyze, Improve and Control
What is TQM?

Total Quality Management

- Organization-wide continuous process improvement
  - to meet and exceed customer needs (delighted customers)
  - at lower costs, with higher revenue
  - with employee empowerment
  - focused on processes and systems
  - using management by fact
  - enabled by continual training and learning
  - assessing progress to goals that evolve with a plan
What is TQM?

Total Quality Management

♦ MUCH more than a “program of the month,” an integration of the work of many gurus (Deming, Juran, Crosby, ...)

♦ Total Quality Mindset
Software Quality Framework for TQM

Building quality products is nurtured by a solid framework that supports quality efforts for all employees.

- **Policy** clearly expresses organization commitment.
- **Objectives** are current measurable improvements identified by the organization (corporation, group, ...).
- **System** is the collection of processes and procedures used to achieve the objectives.
- **Organization** is a team of experts who coach, teach, evaluate, and check the quality activities of the teams in the whole organization.
What is BPR? Business Process Re-engineering

- The fundamental rethinking and radical redesign of business processes to achieve dramatic improvements in performance (cost, quality, capital, service, speed)
  - **Fundamental** - first determine what must be done, then how; ignore the current approach
  - **Radical** - disregard current procedures; get to the root of things
  - **Processes** - entire collection of activities that transform inputs into outputs of value to the customer
  - **Dramatic** - aimed at major, not tuning, improvements

What is Six Sigma?

- Developed by Motorola
- Sigma often used as a scale to measure goodness
- Process improvement program aimed at reducing defects
- Levels of abilities within organizations
  - Yellow Belts - trained
  - Green Belts – implementors that retain regular job responsibilities
  - Experts or Black Belts – Experts that focus on implementation full time
  - Master Black Belts – Organizational Coaches

Source:
http://www.motorola.com/motorolauniversity.jsp
Conceptualize the business... then the software

- Customers optimize the benefits of software by first developing a business vision that includes:
  - People and organization
  - Process and procedures
  - Environment and climate
  - New and emerging technologies
  - Changing attitudes and politics
  - Economy and funding

- What is the strategic vision for the business?
- How can the strategic vision be realized through change and supported by software?
Software Quality Assurance – Throughout the Development Life Cycle

- **Validation**
  - Are we doing the **right** things?
    - Is what’s being created that which is **required**?
    - Will it support our vision?

- **Verification**
  - Are we doing the **things** right?
    - Does the implementation provide **accurate results**?
    - Does the implementation do **what** was specified?
    - Does the implementation work as **documented**?
Understanding The Software Development Life Cycle

- Concept Exploration
  - Statement of need

- System Exploration
  - System Interface Specification

- Requirements
  - Software Requirements Specification

- Design
  - Software Design Description

- Implement
  - Software Validation/Verification Plan

- Installation
  - Software Validation/Verification Report

- Operations & Support
  - User Documentation

- Maintenance
  - Maintenance Documentation

- Retirement

Business/Project Strategy and Planning
- Feasibility Study
- Business Case
- Goals & Objectives
- Project Charter
- Initial Project Plan
- Business Rqmts

Test Planning
- Software Test Plan
- Acceptance Criteria

Test Development and Execution
- Software Test Cases
- Software Test Scripts
- Software Test Results
- Problem Reports
- Change Requests
- Acceptance

Verification and Validation
- Software Validation/Verification Plan
- Software Validation/Verification Report

The Software Quality Institute
© 2007 The University of Texas at Austin
Software Quality - A Tug of War

Software Quality \( (t) \) = 
# defects injected \( (t) \) - # of defects removed \( (t) \)

what should your quality target at delivery be?
The Role of Users in Improving Quality Outcomes

- What you want vs. what you get – don’t be surprised!
- Participation throughout the software development process
- Users and developers share responsibility for
  - Requirements elicitation and management
  - Prototyping feedback
  - Static testing (reviews)
  - Dynamic testing (system and acceptance testing)
Managing Requirements are Key

- Understand your needs and avoid being overly prescriptive in the early stages
- Make sure your requirements are requirements – do your part to control scope
- Develop scenarios, storyboards, work flows to help developers understand the business context/need
- Assist developers in writing Use Cases, Requirements Specifications
- Work with developers and management to prioritize requirements
- Develop test criteria along with requirements
- Trace requirements through to acceptance testing
Prototyping

- Conceptual representations of the software
  - Paper-based
  - User-interface based
- Prototypes are tools - not a complete system
- Interactive process between users and developers
- Improves understanding of requirements – reduces misunderstandings
- Provides critical view of the software to users – look and feel
- Increases user satisfaction
Static Testing with Reviews

- Does not require execution on a computer
- Validation of software development products before the code is written
- Allows the detection and removal of errors and identifies omissions early in the development process
- Provides an opportunity for learning and a basis for making improvements
- Users should be involved in some reviews
  - Most importantly – Requirements and Test Plan reviews
  - Other reviews on Planning documents and user documentation.
Dynamic Testing

- Requires execution on a computer
- Developers perform dynamic tests early in the stages of code development
- Users perform dynamic testing in the latter stages of the code development
  - System Test (sometimes)
  - Acceptance Test
- Goal is to uncover and fix as many errors as possible within the bounds of diminishing returns
- Cannot ‘Test in Quality’ but can confirm the ‘Built-in Quality’
Acceptance along the way

- **Acceptance testing**
  - Determines whether the software is satisfactory and meets the business needs
  - Is typically the last step in software development
  - Major misunderstandings/mistakes discovered during acceptance testing are the most difficult and expensive to correct

- Early input and involvement of users in validating software development artifacts and deliverables reduces the likelihood of ‘surprises’ during Acceptance Testing
Requirements

The critical success factors for a software product!