Defining a Quality Quality Plan

www.construx.com

© 2003 Construx Software Builders, Inc. All Rights Reserved.

Construx
Delivering Software Project Success

Who is Earl?

- 15 years in IT
- 3 of those with DoD as a Quality Assurance Representative
  - Certified for Mechanical, Electronics, and SW
- Several years at Boeing
- Several years in wireless (stint as QA and SEPG manager!)
- Currently instructor/consultant at Construx Software
What is Quality?

- Quality is seven attributes (portability, reliability, efficiency, usability, testability, understandability, modifiability) *Glass*
- Quality is conformance to requirements *Crosby*
- Quality is fitness for use *Deming*
- Quality is value to some person *Weinberg*
- Quality is whatever the customer decides quality is *Ginac*
- Quality is an attitude or state of mind *Juran*

What is a Quality Plan

- A Quality Plan is different than a Test Plan
  - Defines the Quality Goals
  - Realistic about where defects come from
  - Selects appropriate detection and prevention methods
  - Has means not to “go dark”
Upstream/Downstream effect

Phase That a Defect Is Created

- Requirements
- Architecture
- Detailed design
- Construction

Phase That a Defect Is Corrected

Cost to Correct

50-200X

50-200X

Requirements Architecture Detailed design Construction Release

Effort/ Cost/ Schedule

Quality improvement motivated primarily by economics

Quality improvement motivated by quality, per se

Percentage of Defects Removed Before Release

~95% 100%

Brute Force Quality

consulting ● training ● project outsourcing ● products ● construx.com

consulting ● training ● project outsourcing ● products ● construx.com
Cost of Quality Details

- Prevention
  - Quality planning
  - Formal process audits
  - Training
- Detection
  - In-process and inter-process review
  - Test equipment
  - Equipment calibration and maintenance
  - Testing

- Failure Costs
  - Rework
  - Repair
  - Scrap
  - Failure mode analysis
  - Complaint resolution
  - Product return and replacement
  - Help line support
  - Warranty work

Economical Quality

- Effort/Cost/Schedule
- Percentage of Defects Removed Before Release

- To the left of 95%, move down and right
- To the right of 95%, move down to the line

≈ 95% 100%
Defining Quality Goals

- A goal is SMART
  - Specific
  - Measurable/Testable
  - Attainable
  - Relevant
  - Time-bound
- The primary project goals must be described in the chartering process

Danger of Lack of Goals

- Repair-service behavior
  - Without any clear idea of what the benchmarks are, we go in search of things that are broken and our goal becomes fixing them
- Know-how behavior
  - We often don’t solve the problems that need to be solved but the ones we know how to solve
- No justification for any quality action
  - “I’m doing it for quality” anarchy
  - Wasting a lot of time with no alignment
ISO/IEC 9126-1 - Quality Model

- **Functionality**
  - Suitability, Accuracy, Interoperability, Security
- **Reliability**
  - Maturity, Fault tolerance, Recoverability
- **Usability**
  - Understandability, Learnability, Operability
- **Efficiency**
  - Time behavior, Resource utilization
- **Maintainability**
  - Analyzability, Changeability, Stability, Testability
- **Portability**
  - Adaptability, Installability, Conformance, Replaceability

---

Can’t Maximize All of Them

<table>
<thead>
<tr>
<th>Functionality</th>
<th>Reliability</th>
<th>Usability</th>
<th>Efficiency</th>
<th>Maintainability</th>
<th>Portability</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- ● = Typically Conflicting Qualities
- ○ = Typically Supporting Qualities
Business Decisions

- The business should set a scale for every Quality Attribute

Tough Questions

- There is no right/wrong ordering of the attributes or the scales
- Each project must decide these at some point
  - As with any requirement, the question is not “if”, but “when”, “who”, and “how”
  - If not decided up front, will be decided over and over again throughout the project
Bugs Everywhere

- Inserting faults is the norm
  - High reliance on human communication
  - High complexity
  - Non-stable environment
- As the amount of effort increases, the number of faults will increase

How Faults Happen

- Fault Introduction
- Existing Faults
- Project Activity
- Fault Removal
- Carry Forward Faults
Quality Philosophy

All effective quality approaches are based on two realities

1. That mistakes will be made throughout the project
   Project success depends on positioning the project team to detect these mistakes early so that they can correct them quickly and easily
2. That the way things are built greatly impacts how well they can be built
   Project success depends on using effective and efficient methods by the project team

QA vs. QC

- Quality Assurance (QA) is fault prevention through process design and auditing
  - Creating processes, procedures, tools, jigs, etc. to prevent faults from occurring
  - Examples: Templates, checklists, guides
- Quality Control (QC) is fault/failure detection through static and/or dynamic testing of artifacts
  - Examining the artifact against pre-determined criteria to measure conformance
  - Examples: Code testing, peer reviews, beta trials
Correct More Faults Earlier!

Phase That a Defect Is Created

Find Here

Not Here

Cost to Correct

Phase That a Defect Is Corrected

Basic Early Strategy

Fault Introduction

QA Process, Tools, Templates

Existing Faults

Project Activity

Fault Removal

QC Event

Carry Forward Faults
Reduce Fault Impact

Phase That a Defect Is Created

Phase That a Defect Is Corrected

Cost to Correct

Requirements
Architecture
Detailed design
Construction

1X
1X
10X?
10X?

Lowering Fault Impact

Taking small baby steps with each release so the fault can’t grow too big

Designing for fault resistance using hardened design strategies or redundant systems

Making complete set of “blueprints” of the software so faults can be found quickly
Detection Methods

Quality Control

- Reviews
  - Personal, peer, pair, management, QA, independent

- Testing
  - Structural, functional, integration, stress/performance, regression, field, acceptance

- Simulations
  - Prototypes, models

- Field Trials
  - Prototypes, beta testing

- Mathematical
  - Proofs of correctness

Detection Effectiveness

- Design Check
- Design Review
- Design Inspection
- Code Inspection
- Prototype
- Code Check
- Unit Test
- Functional Test
- Integration Test
- Field Trial
- Cumulative

- Highest
- Modal
- Lowest
Multiple Techniques

- To get to acceptable defect removal rates requires a combination of techniques
- Unit testing, component testing, and system testing often remove less than 60% of defects
  - Neither effective or efficient!
- Skipping reviews and/or inspections will result in high tail-end costs

Prevention Tactics

Quality Assurance

- Culture
- Professional development
- Practice toolbox selection
- Checklists & Templates
- Audits
- Quality gates
- Team structure
- Continuous process improvement
Culture Quiz

- Who are recognized as the “best” performers?
  - Why?
- What is the first thing likely to get cut when the deadline is fast approaching?
  - Are there reasonable things to cut?
- How does management get rewarded?
- Can you name more than two quality initiatives in the last four years?

Culture Change

- Not easy
  - Like hitting your head against a huge stone flywheel
  - Needs a critical mass of people
- Logic loses
  - It has more to do with emotions and rewards
  - Demonstrated success is critical
- Truth-telling is critically important
  - Need to break through the painted ceiling
- Work with motivation
  - “Quality” is not a magic word
Aspects of Prevention

Continuous Process Improvement

Feedback

Project Work

Conclusion
Quality Assurance Practices in the Software Lifecycle

- System Objectives
- Requirements
- Design
- Coding
- Quality Goals
- Acceptance Test Execution
- Integration and Component Test Execution
- System Test Execution
- Retrospectives and data collection

Quality Focus/Circles/Analysis/Continuous Improvement

Quality Control Practices in the Software Lifecycle

- System Objectives
- Requirements
- Design
- Coding
- Fault Detection & Removal
- Integration and Component Test Planning
- Unit Test Planning
- Unit Test Execution
- System Test Execution
- Acceptance Test Planning
- Acceptance Test Execution
Cost of Quality

QA/QC Late in the Project

QA/QC Early in the Project

Contact Information

www.construx.com
(425) 636-0100

Construx
Delivering Software Project Success

- Software Projects
- Project Consulting
- Seminars
- On-site Training

sales@construx.com

Free Stuff! → CxOne Basic

www.construx.com