Integrating Quality Assurance into the Software Development Life Cycle

Leslie Tierstein, STR LLC
Hilary Benoit, W R Systems

Overview (1)
- Why bother with QA?
- QA and the SEI CMM/CMMI
- Defining the Software Development Process
- Setting up the QA Function
- Selecting the Pilot Project
Overview (2)

- Tools, Procedures and Activities
- Lessons Learned
- The next step - from pilot project to all projects
- Summary

What is Quality?

- Quality - “The totality of features and characteristics of a product or service that bears on its ability to satisfy given features.” (American Society for Quality, 1978)
What is Quality Assurance?

- Quality Assurance - “consists of all the planned and systematic activities implemented within the quality system that can be demonstrated to provide confidence that a product or service will fulfill requirements for quality.”

Why Bother with QA?

- Need to produce quality software products in a repeatable and consistent manner
- Checks and Balances
- Customer Assurance
- Carnegie Mellon’s Software Engineering Institute’s Capability Maturity Model (SEI CMM) - requires Software Quality Assurance (SQA)
SEI CMM and CMMI

- Model to gauge the maturity of the software development process
- Superceded by CMM Integration (CMMI), incorporating ISO-9000 principles
- Software Process framework
  - Five maturity levels
  - Key Process Areas (KPAs)

SEI CMM Maturity Levels

- Level 1 - Ad hoc (chaotic)
- Level 2 - Repeatable (disciplined)
- Level 3 - Defined (standard; consistent)
- Level 4 - Managed (predictable)
- Level 5 - Optimizing (continuously improving)
SEI CMMI Maturity Levels

- Level 1 - Ad hoc
- Level 2 - Managed
- Level 3 - Defined
- Level 4 - Quantitatively Managed
- Level 5 - Optimizing

CMM/CMMI KPAs

<table>
<thead>
<tr>
<th>CMM Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial – Adhoc (&quot;chaotic&quot;)</td>
<td>None</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>CMMI Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>Initial – Adhoc (&quot;chaotic&quot;)</td>
<td>None</td>
</tr>
</tbody>
</table>

- 80 - 90% of all software development organizations
# CMM/CMMI KPAs

<table>
<thead>
<tr>
<th>CMM Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
</table>
| 2 Repeatable - Disciplined | • Software Configuration Management  
 • Software Quality Assurance  
 • Software Subcontractor Management  
 • Software Project Tracking and Oversight  
 • Software Project Planning  
 • Requirements Management |

<table>
<thead>
<tr>
<th>CMMI Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
</table>
| 2 Managed - Planned Performed Managed Controlled | • Configuration Management  
 • Process and Product Quality Assurance  
 • Supplier Agreement Management  
 • Project Monitoring and Control  
 • Project Planning  
 • Requirements Management |

<table>
<thead>
<tr>
<th>CMM Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
</table>
| 3 Defined - Standard Consistent | • Peer Reviews  
 • Inter-group Coordination  
 • Software Product Engineering  
 • Integrated Software Management  
 • Training Program  
 • Organization Process Definition  
 • Organization Process Focus |

<table>
<thead>
<tr>
<th>CMMI Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
</table>
| 3 Defined - Consistent across the organization | • Verification  
 • Integrated Project Management  
 • Requirements Development  
 • Technical Solution  
 • Product Integration  
 • Organizational Training  
 • Process Definition and Process Focus |
CMM/CMMI KPAs

<table>
<thead>
<tr>
<th>CMM Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Managed</td>
<td>• Software Quality Management</td>
</tr>
<tr>
<td></td>
<td>• Quantitative Process Management</td>
</tr>
<tr>
<td>Predictable</td>
<td></td>
</tr>
</tbody>
</table>

CMMI Maturity Level | Key Process Areas                      |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>4 Quantitatively</td>
<td>• Quantitative Project Management</td>
</tr>
<tr>
<td>Managed</td>
<td>• Organizational Process Performance</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

- Measures to quantify quality, process, and improvements

CMM/CMMI KPAs

<table>
<thead>
<tr>
<th>CMM Maturity Level</th>
<th>Key Process Areas</th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>• Process Change Management</td>
</tr>
<tr>
<td>Continuously</td>
<td>• Technology Change Management</td>
</tr>
<tr>
<td>improving</td>
<td>• Defect Prevention</td>
</tr>
<tr>
<td></td>
<td></td>
</tr>
</tbody>
</table>

CMMI Maturity Level | Key Process Areas                      |
<table>
<thead>
<tr>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>5 Optimizing</td>
<td>• Causal Analysis and Resolution</td>
</tr>
<tr>
<td>Continuously</td>
<td>• Organizational Innovation and Deployment</td>
</tr>
<tr>
<td>improving</td>
<td></td>
</tr>
</tbody>
</table>

- Proactive measures to improve quality
- 4-5 organizations nationwide
Define and Document the Development Process

- Software development process is the foundation to the QA process
- Should be:
  - well-defined
  - simple
  - clear phases
  - entry and exit criteria

Software Development Process/Methodology

- Strategy
- Analysis
- Design
- Build and Test
- Deploy
- Maintain
Define and Set up the QA Function (1)

- Purpose and Goals
  - Control cost, schedule, quality
  - “Time box” of development
- Activities - vary with life cycle phase
  - QA <> Testing
- How to staff?
  - Programmers or non-programmers
- Skills required

Define and Set up the QA Function (2)

- Resources
  - Corporate
  - Per project
- Independent Organization
- Management Support
Select the Pilot Project

- Oracle full life cycle development project
- Oracle Designer/Developer
- Client-Server - Windows and HP-UNIX
- Government contract - customer requirement to achieve SEI CMM Level 2
- Opportunity to integrate software quality assurance into the full life cycle

Integrate QA into Life Cycle Phases

- Phase entry and exit criteria - inputs and outputs
- Quality Checkpoints
- Audits and reviews of products and processes
- Timely management notification of problems - Risk Management
QA and the Strategy Phase (1)

- Develop the QA Plan and Procedures
  - MIL-STD-498
  - ISO-9000
- Create QA records
- Determine Metrics
- Review and Analyze Requirements
- Establish the Deliverable Review Process
QA and the Strategy Phase (2)

- Project Standards and Procedures
  - Shared components and their management
  - Externally developed coding standards
  - Internally developed standards and procedures

Tools and Techniques

- QA Records - Word templates
- QA Activities Tracking System (QATS)
- Deliverable Review Route Sheets
- Quality Control Reports
- Requirements Traceability Matrix (RTM)
- Checklists and Forms
Keep QA records

Document all QA reviews and audits
Audit trail of activity
Metrics
Sample form

Quality Assurance Tracking System (QATS)

Database
Reports
Metrics
QA activities tracking
Deliverable Review Process

1. Perform Technical Review
2. If Approved, Perform Required QA Inspection
3. If Approved, Establish Configuration Management Baseline
4. Client Review
5. If Approved, Enter Next Phase or Perform Corrective Action
6. If Approved, Enter Next Phase or Perform Corrective Action

Deliverable Review Route Sheet (Sample)

Project Deliverable Route Sheet

<table>
<thead>
<tr>
<th>Project:</th>
<th>Deliverable:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task #:</td>
<td>Return To:</td>
</tr>
</tbody>
</table>

- Configuration Management - Document Check-Out
  - Configuration Manager:
  - Deliverable name for check-out, including version:
  - Date of Check-Out:
  - Original Document Name(s):
  - New Document Name(s):

- Complete:

- Technical Review
  - Date Submitted:
  - Reviewed By:
  - Date Completed:
  - Follow-up Review Required?

- Complete:

- Initial Review
  - Date Submitted:
  - Reviewed By:
  - Date Completed:
  - Follow-up Review Required?

- Complete:

- Follow-up Review (if needed)
  - Date Submitted:
  - Reviewed By:
  - Date Completed:
  - Follow-up Review Required?

- Complete:

- QA Review

- Complete:

Complete:
QA and the Analysis Phase

- Begin Technical & QA Reviews and Audits
  - Requirements Document
  - Function Hierarchy/Process Flow Diagram
  - Requirements Traceability Matrix (RTM)
  - Logical Database Design
    - Entity Relationship Diagram(s) (ERD)
    - Data Dictionary
  - Create Read Update Delete (CRUD) Matrix

Requirements (1)

- Reviewed for clarity, completeness, redundancy, and testability
- Specific enough to be testable
  - specify what needed to be done,
  - not how to do it
- Uniquely identified - for later traceability
Requirements (2)

Functional Requirements List

<table>
<thead>
<tr>
<th>Requirement Identifier</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>FLS 01 XXX 002</td>
<td>Verify and activate DODAAC data.</td>
</tr>
<tr>
<td>FLS 01 XXX 003</td>
<td>Maintain and print DODAAC data.</td>
</tr>
<tr>
<td>FLS 01 XXX 005</td>
<td>Create Navy Unit Identification Code (UIC) reports. (Deferred at CCB of mm/dd/yy)</td>
</tr>
<tr>
<td>FLS 01 XXX 006</td>
<td>Automatically update Master Address file based on DODAAC inputs.</td>
</tr>
<tr>
<td>FLS 01 XXX 007</td>
<td>Maintain, print, and view Master Address file. (Deferred at CCB of mm/dd/yy)</td>
</tr>
<tr>
<td>FLS 01 XXX 012</td>
<td>Provide the capability to import and export transactions via DAAS. These transactions include: MILSTRIP, MILSTRAP, MILSBILLS, DODAAC, DLSC, DLSS, SSR, WSF, and KSS.</td>
</tr>
<tr>
<td>FLS 01 XXX 013</td>
<td>Unload mailing and shipping addresses to TANDATA and FEDEX.</td>
</tr>
</tbody>
</table>

Requirements (3)

- Functionality
- Usability
- Performance
Requirements (4)

- Verified by QA as implemented in finished application and that every feature of the application corresponds to a requirement
- Possible defects
  - Missing functionality
  - Functionality with no requirement ("creeping featurism")

Quality Control Reports
QA and the Design Phase (1)

- Technical and QA Reviews and Audits of all Deliverables
  - Physical Database Design
  - Module Network (Menu) Hierarchy
  - Module Specifications
  - Prototypes
    - User Interface
    - Scenarios/walkthroughs

QA and the Design Phase (2)

- Updated RTM
  - Configuration Control Board (CCB)
  - Requirements Management - MoSCoW List
- PDRs and CDRs attendance (Quality Checkpoints)
- Verification of Corrective Action
  - Action items
  - Problem reports
Other QA Techniques

- Code Walkthroughs
  - group activity
- Peer Reviews
  - one-on-one
  - inspection
- Centers of Excellence (COE)
  - training forum
  - information exchange

Peer Review Form

Peer Review - Report

<table>
<thead>
<tr>
<th>Date/Time:</th>
<th>Date:</th>
<th>Start Time:</th>
<th>End Time:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Work Product:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Peer Review Leader:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Author:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Reviewers:</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Notes taken by:</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Note all action items resulting from this peer review session. Continue on a separate sheet, if necessary.
QA and the Build and Test Phase (1)

- Configuration Management (CM)
  - Version/build control through software
  - Control software and documentation
- Peer Reviews and Code Walkthroughs
- Prototypes - customer demos
  - Review form and follow-up
- Unit Test - formal
- Problem Tracking

Prototype Review Form
Unit Test Checklist

Initial Forms Module Checklist (Prior to Demo)

Module Name:       Tester: <Tester name>
Date of Test:  mm/dd/yy    Developer:

<table>
<thead>
<tr>
<th>Item</th>
<th>Tested</th>
<th>Tester Comments</th>
<th>Priority</th>
<th>Fixed</th>
<th>Verified</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Pass</td>
<td>Fail</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Layout / Window</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Data</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Fields Color</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Size</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Tab position</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Labels and Titles</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Date</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hints</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Required Fields</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Scrollbars</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Help</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Abbreviations</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Phone Numbers</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Test Form (Sample)

- Formal
- “with a form”
- with a review and approval process
Problem Tracking (1)

problem tracking software interface

Problem Tracking (2)

Problem Tracking Summary Report

<table>
<thead>
<tr>
<th>#</th>
<th>Module</th>
<th>Date Reported</th>
<th>Assigned To</th>
<th>Problem Status</th>
<th>Doc#</th>
<th>Description of Problem</th>
</tr>
</thead>
<tbody>
<tr>
<td>244</td>
<td>ENR_424</td>
<td>04/14/2012</td>
<td>Mark Camside</td>
<td>Opened</td>
<td>No</td>
<td>424 needs to be fixed.</td>
</tr>
<tr>
<td>512</td>
<td>ENR-CDT</td>
<td>05/16/2012</td>
<td>Mark Camside</td>
<td>Opened</td>
<td>No</td>
<td>1. What does it mean?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>2. What is the budget?</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>3. What is the issue?</td>
</tr>
<tr>
<td>326</td>
<td>ENR_424</td>
<td>05/16/2012</td>
<td>Mark Camside</td>
<td>Opened</td>
<td>No</td>
<td>3. What is the issue?</td>
</tr>
</tbody>
</table>
QA and the Build and Test Phase (2)

- Integration Test - partially automated
  - Business scenarios via QA/Director
  - Load Runner for load testing
- System Test
  - Customer/client involvement
  - Acceptance test
- Integrated Project Teams (IPT)
- Independent Validation and Verification (IV&V)

A Test Data Entry Screen in QA/Director
Entering Bugs into QA/Director

QA and the Deployment Phase

- Phased implementation (no “Big Bang”)
  - By function/subsystem
  - By organization/user group
- QA reviews and test procedures continued
- Expedite test and delivery of modified code - fast turnaround required
- User training - review and test of training materials
QA and the Maintenance Phase

- Continue with established QA and CM procedures
  - Action Items/User meetings
  - MoSCoW evaluation and followup
  - Problem Reports - user accessible

Collect Project Metrics

- Areas of greatest problems/defects
- Number/results of QA audits and reviews
- Test coverage and test results
- Problem Reports/Defects found - e.g., per module, per subsystems, classification and type, time taken to resolve
- Development Time - Estimated vs. Actual
- - SEI CMMI Level 4 Quantitatively Managed
Process Improvement

- Take existing process
- Analyze step-by-step
- Modify to improve
  - e.g., testing/QA/CM process
  - unit testing - formalized
- Training - e.g., COEs, Test Writing, Testing
- - SEI CMMI Level 5 Optimizing

Lessons Learned

- Acceptance of QA
- What worked
- What didn’t work
Acceptance of QA

- QA function - perceived as “value added”
- Not confrontational/critical
- Provide guidance, oversight, training
- Assistance in process improvement
- Well-designed QA Plan and procedures
- Concrete activities and reports
- QA Schedule
- Part of the team

What Worked

- Formal Review process of deliverables
- Strong Requirements Management and RTM
- Collaboration of QA with TM and CM
- Participation of QA in meetings
- QA sign-off in Testing
- Formal bug tracking
- Peer Reviews
What did NOT work

- Excessive paperwork for developers
- Anything causing lengthy turnaround on deliverables
- Expecting developers to read lengthy standards documents
- Assuming developers would enter all required RTM information
- Informal Unit Testing

Implementing QA on all Projects

- “Clone” the process
- Use successful “artifacts”
- Target training
- Use “Lessons learned”
- Expand the SQA group
Summary: QA activities to integrate into the SDLC (1)

- Scheduled audits & reviews of all project processes and deliverables
- Maintenance of QA records of reviews and audits
- Management notification of non-compliance with standards and procedures, or of notable problems
- Resolution of Problem Reports and Verification of corrective action

Manage the Requirements’ Traceability process

Summary: QA activities to integrate into the SDLC (2)

- Managing the Requirement Traceability process
- Peer Reviews, code walkthroughs
- Including QA personnel in project and customer meetings
- Providing training in standards, testing, or other QA-related topics
- Independent Testing
Summary of Steps

- Define and Document the Software Development Process
- Verify/Obtain support of Top Management
- Set up the QA function
- Select the Pilot Project
- Integrate QA activities into the development life cycle phases
- Use Lessons Learned to implement QA on other projects
- Expand QA group function, as required

Conclusion

- Successful deployment of pilot project
- Integration of software quality assurance into the life cycle
- SEI CMM - Level 3 compliant
Quality-Related Web Sites

- [www.asq.org](http://www.asq.org) - American Society for Quality (ASQ)
- [www.iqa.org](http://www.iqa.org) - Institute of Quality Assurance
- [www.iso.ch](http://www.iso.ch) - International Organization for Standardization (ISO)
- [www.nist.gov](http://www.nist.gov) - National Institute of Standards and Technology (NIST)
- [www.qaiusa.com](http://www.qaiusa.com) - Quality Assurance Institute (QAI)
- [www.sei.cmu.edu](http://www.sei.cmu.edu) - Carnegie Mellon University's Software Engineering Institute (SEI CMM)
- [www.quality.org](http://www.quality.org) - Quality Resources Online

About the Authors

- ltierstein@earthlink.net
- hbenoit@wrsystems.com