

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
- Superseded by CMM Integration (CMMI), incorporating ISO-9000 principles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

CMM Maturity Level	Key Process Areas
I Initial – Adhoc ("chaotic")	• None

CMMI Maturity Level	Key Process Areas
I Initial – Adhoc ("chaotic")	• None

- 80 - 90% of all software development organizations

CMM/CMMI KPAs

CMM Maturity Level	Key Process Areas
2 Repeatable - Disciplined	<ul style="list-style-type: none"> • Software Configuration Management • Software Quality Assurance • Software Subcontractor Management • Software Project Tracking and Oversight • Software Project Planning • Requirements Management

CMMI Maturity Level	Key Process Areas
2 Managed - Planned Performed Managed Controlled	<ul style="list-style-type: none"> • Configuration Management • Process and Product Quality Assurance • Supplier Agreement Management • Project Monitoring and Control • Project Planning • Requirements Management

CMM/CMMI KPAs

CMM Maturity Level	Key Process Areas
3 Defined - Standard Consistent	<ul style="list-style-type: none"> • Peer Reviews • Inter-group Coordination • Software Product Engineering • Integrated Software Management • Training Program • Organization Process Definition • Organization Process Focus

CMMI Maturity Level	Key Process Areas
3 Defined - Consistent across the organization	<ul style="list-style-type: none"> • Verification • Integrated Project Management • Requirements Development • Technical Solution • Product Integration • Organizational Training • Process Definition and Process Focus

CMM/CMMI KPAs

CMM Maturity Level	Key Process Areas
4 Managed Predictable	<ul style="list-style-type: none"> • Software Quality Management • Quantitative Process Management

CMMI Maturity Level	Key Process Areas
4 Quantitatively Managed	<ul style="list-style-type: none"> • Quantitative Project Management • Organizational Process Performance

- Measures to quantify quality, process, and improvements

CMM/CMMI KPAs

CMM Maturity Level	Key Process Areas
5 Optimizing Continuously improving	<ul style="list-style-type: none"> • Process Change Management • Technology Change Management • Defect Prevention

CMMI Maturity Level	Key Process Areas
5 Optimizing Continuously improving	<ul style="list-style-type: none"> • Causal Analysis and Resolution • Organizational Innovation and Deployment

- 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

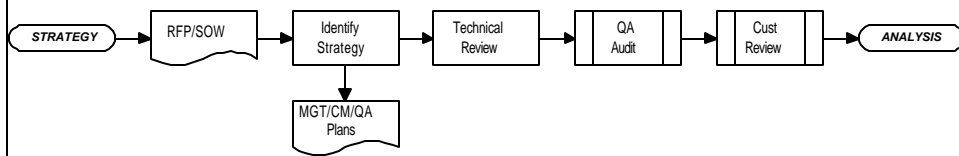
Select the Pilot Project

- 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

Strategy Phase



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

QA QUALITY REVIEW
QAR-02

PROJECT: INITIAL POLL OUTLET

TASK#: _____ DATE: _____

REMEMBERED BY: _____

Page 1 of 2

1. TASK DESCRIPTION

2. CHECKLIST

DESCRIPTION	CONFORM TO SITE REQUIRE	COMMENTS
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	
	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/>	

3. WORK PRODUCT

Is the end result polished, professional and for delivery?

Are all items trainable to the equipment?

Does the end result satisfy all of the requirements?

4. METHODOLOGY

Were procedures adopted and used?

Were the items established and achieved?

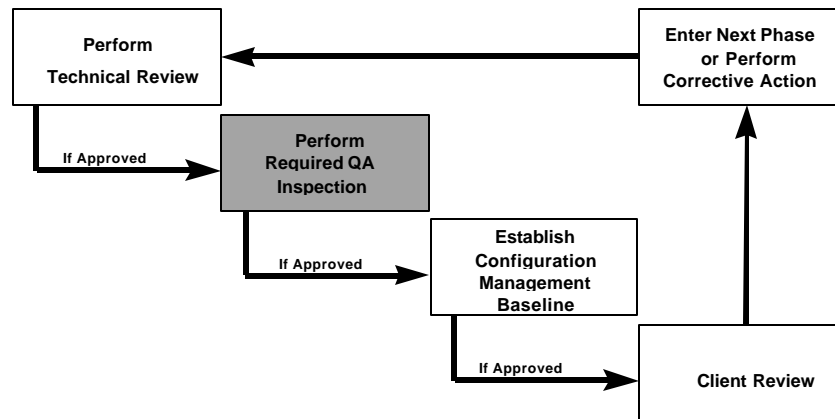
How do the methods meet the standards?

- 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



Deliverable Review Route Sheet (Sample)

Project Deliverable Route Sheet

Project: _____ Files of: _____
 Task #: _____ Deliverable: _____
 Date: _____ Return To: _____

Configuration Management - Document Check-Out Complete

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

Technical Review Complete

Initial Review

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

Follow-up Review (if needed)

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

QA Review 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

FLS Main

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

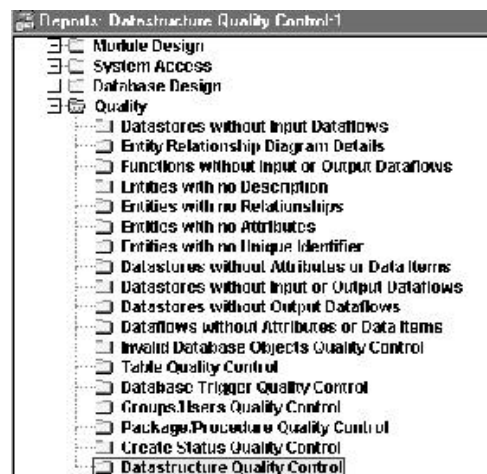
Requirements (3)

- 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
 - I User Interface
 - I 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

Date/Time:	Date:	Start Time:	End Time:
Work Product:			
Peer Review Leader:			
Author:			
Reviewers:			
Notes taken by:			
Action Items:	<i>Note all action items resulting from this peer review session. Continue on a separate sheet, if necessary.</i>		

Unit Test Checklist

Initial Forms Module Checklist (Prior to Demo)

Module Name:
Date of Test: mm/dd/yy

Tester: <Tester name>
Developer:

Item	Tested		Tester Comments	Priority	Fixed	Verified
	Pass	Fail				
Layout / Window						
Data						
Fields Color						
Size						
Tab position						
Labels and Titles						
Date						
Hints						
Required Fields						
Scrollbars						
Help						
Abbreviations						
Phone Numbers						

Test Form (Sample)

Test Form

Project:		SW Version:	
Module ID & Short Name:		Date:	
SPR #(s) associated with this test:			
SPR(s) attached: YES <input type="checkbox"/> No <input type="checkbox"/>		If there is no bug SPR associated with this module, please check below, as appropriate: This is an Enhancement <input type="checkbox"/> This is New Development <input type="checkbox"/>	
Associated L.C.B. #(s) - if applicable:			
Test Site: WBS <input type="checkbox"/> Customer <input type="checkbox"/>	Developer Tester(s):	TEST database: HP725 <input type="checkbox"/> T600 <input type="checkbox"/>	
Test Plan: Software Test Plan <input type="checkbox"/>		PRODUCTION database: T600 <input type="checkbox"/>	
Brief Description/Purpose of Test: <i>Direct appropriate the Report, if possible.</i>			
Other affected modules: (for unit integration/impact analysis) These include other screens or reports that this change will affect and that changes will need to be made.			
Other associated packages, functions, procedures and views: <i>Specify ALL that need to accompany this test.</i>			
Test Complete? Yes <input type="checkbox"/> No <input type="checkbox"/>		Test Result: Pass <input type="checkbox"/> Fail <input type="checkbox"/>	
Results/Comments:			
Item/Action to be Tested (complete on separate sheet, if necessary)		Result	
Developer Tester Signature(s) & Date:		Peer Review Signature & Date:	
Does this test documentation number resolution as a result of this test (Enhancement)? YES <input type="checkbox"/> NO <input type="checkbox"/>			
QA Signature & Date:			
CM Information:	Date moved to T600-TEST:	Date moved to T600- PROD/CM/CP:	

- Formal
- “with a form”
- with a review and approval process

Problem Tracking (1)



Problem Tracking (2)

Run by MCASTALDO
on June 4, 2002 4:19 PM

eGrants Problem Tracking Summary Report

#	Module	Date Reported	Assigned To	Problem Status	Doc?	Description of Problem
244	ENT_424	04/24/2002	Mark Castaldo	Opened	No	424 needs a ROW LOV.
312	ENT_GO_CERT	05/16/2002	Mark Castaldo	Opened	No	<ol style="list-style-type: none"> 1. If the GO obligates funds and then realizes the amount is mistyped, how does it get fixed? 2. When the NGA button comes up, the Cancel option still obligates the funds, instead of actually cancelling. 3. (From Katie Dooley:) If by error incorrect funds are obligated, how do you correct this error?
326	ENT_424	05/16/2002	Mark Castaldo	Opened	No	I can submit two applications continuing the same prior grant under the same NOFA. Check grants 03SR016696 (the newer one, not submitted) and 03SR016692 (the older one, created yesterday and submitted earlier today after fudging mightily on the performance dates). Both in the test database here. Interestingly enough, when the system populated data "from the original grant" on 03SR016696 it used values taken from the earlier continuation application 03SR016692, INCLUDING the stuff on the front page for total amounts and allocating outside support.

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

The screenshot shows a 'Test Form' window with a toolbar at the top containing icons for file operations and navigation. The form fields are as follows:

Test Name:	01APR-1_tc1	18
Status:	Design	Designer: guest
Subject:	Automate PR Generati	Creation Date: 3/8/96
Main:	<input checked="" type="checkbox"/>	TSL Test: <input type="checkbox"/>

Description :
Verify the creation and maintenance of a PR. Ensure a user can create a PR for a wholesale inventory item, that the Purchase Description for this item can be accessed, that this PR will be routed to Supply Technical, and that Technical can identify the item as wholesale inventory and take appropriate action. Ensure that Suggested Source of Supply is related to the PR.

Buttons for 'OK' and 'Cancel' are located on the right side of the form.

Entering Bugs into QA/Director

The screenshot shows a 'Bug Form' window with the following fields and values:

- Status:** Closed
- Assigned To:** Daniel
- Project:** SCCA
- Subject:** Cataloging, Provision
- Severity:** (empty)
- Priority:** (empty)
- Reproducible:**
- To Mail:**
- Summary:** Record failed to delete
- Description:** Step 5 says to delete the record and commit. On commit, the following message is displayed: Error: TRANSACTION FAILED while updating audit data in PATTERN...
- User defined:**
 - Category:** SW
 - Due Date:** (empty)
 - Subsystem:** Cat.Provision
 - Module:** CAT17130

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

About the Authors

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