Nitty Gritty of QA Project Management

11-Feb-09

Carol Perletz

Agenda

Introduction

Introduction

- My background
  - 1984 – hired as a part time tester
  - 1987 – became a “QA Engineer”
  - 1992 – attended the IBM Quality College; began evangelizing SQA
  - 1993 – spearheaded the development of the company’s 1st SDLC
  - 2000 – corporate quality core team member
  - 1989 – 2009
    - Managed test teams
    - Developed/modified SDLC processes
    - Established metrics programs, reviews, audits, quality plans, quality reports
- My experience
  - Companies expect QA engineers to cover testing
  - QA often does not perform quality assurance activities
  - People who call themselves quality professionals may not have the skills or background to perform quality functions
  - QA Engineer is equal to test engineer in the U.S.
  - QA Engineers are not generally found outside the U.S.

Definitions

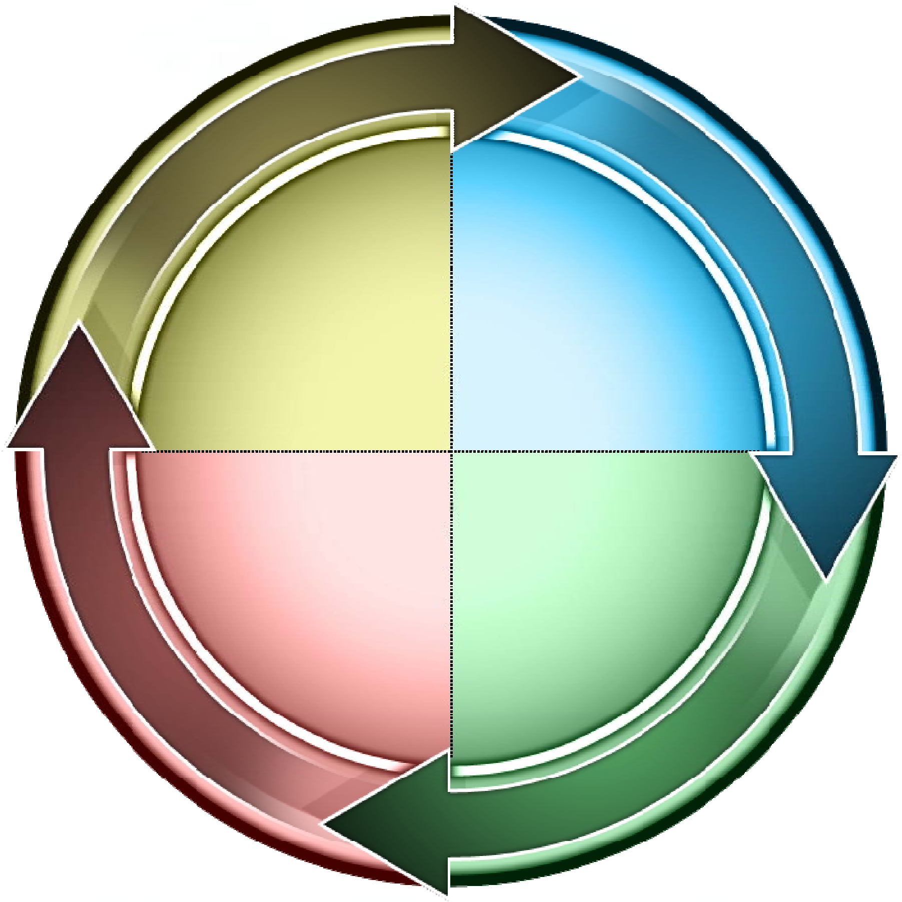
Definitions

- Quality – doing the right thing in the right way (Dave Miller)
- Software quality – software that has a low level of defects when deployed, is reliable, satisfies the majority of users, and is maintainable (Steve Rakitin)
- Software quality assurance – assuring that software meets the 4 criteria above (my definition)
- IEEE 6.10.12-1990 defined QA as assuring that software conforms to its technical requirements
- Testing – the execution of software to find its faults (Watts Humphrey)
- Project management – achieving the needed software quality level while simultaneously respecting the project lead-time and cost constraints (Bharath Srinivasan and Devi Sujathaa)

How do these definitions fit in with today's software projects?

- QA is recent terminology and is used more often when referring to shrink-wrapped or pure software products in the U.S.
- Testing and test engineering became an independent discipline in companies producing operating systems in the 1960s and 1970s
- Companies headquartered in Europe and many companies in the U.S. who produce low level software and hardware have System Test groups, not QA groups
- Companies who hire QA Engineers in the U.S. most often want skilled test engineers
- Most companies I have interviewed with, or been employed in, do not understand ASQ quality engineering

Software Development Life Cycle Processes



Software Development Life Cycle Processes

- Waterfall – a cascade of phases, each dependent upon the completion of the previous phase
  - Requirements analysis
  - Requirements definition
  - Design
  - Implementation
  - Testing
  - Release/maintenance/support
- Modified waterfall
  - Feedback loop back from implementation to design
  - Feedback loop back from testing to design/implementation
- Concurrent development/test
  - Requirements definition
  - Design of code and tests
  - As portions of code are completed, they are tested
  - Formal regression/validation testing of the entire product
  - Release/maintenance/support

Software Development Life Cycle Processes (con't)

- Rapid prototyping
  - Requirements definition
  - Rapid design
  - Prototype building cycles
  - Customer evaluation of prototypes in each cycle
  - Implementation
  - Testing
  - Release/maintenance/support
- Spiral
  - Requirements definition
  - Initial prototype
  - Planning
  - Risk analysis
  - Additional prototype cycles
  - Implementation
  - Testing
  - Release/maintenance/support

Software Development Life Cycle Processes (con't)

- The Unified Process
  - Use case driven – use cases drive the requirements, design, implementation, and test
  - Architecture-centric starting with the platform and then incorporating the use cases
  - Incremental – a staging and scheduling strategy in which parts of the system/product are developed and integrated as they are completed
  - Iterative – mini projects, repeating analysis, design, code & test
- Agile methodology
  - Requirements derived with the customer
  - Incremental development and delivery of working software every 2-3 weeks
  - Solid engineering practices and infrastructure
  - Extreme programming
  - Embraces change

Software Development Life Cycle Processes (con't)

- Scrum
  - Vision/planning – ROI goals, releases, milestones
  - Customer commits to one Sprint at a time; defining business value
  - Product backlog is a prioritized list of requirements
  - Sprint development based on a Sprint Backlog of explicit tasks
  - Sprints last for 30 days after which a product increment must be delivered
  - Daily Scrum meetings to remove obstacles to progress
  - Sprint Review meeting where functionality is demonstrated
  - Product Owner decides after which increment the functionality is released

Software Development Processes

- CMMI – the Capability Maturity Model with guidelines for improving the software process
  - Level 1 – initial
  - Level 2 – repeatable
    - Requirements management
    - Project planning
    - Project tracking and oversight
    - Subcontract management
    - Quality assurance
    - Configuration management
  - Level 3 – defined
    - Organization is process focused
    - Organizational process definition
    - Training program
    - Integrated software management
    - Product engineering
    - Intergroup coordination
    - Peer review
    - System testing

Software Development Processes (con't)

- Level 4 – managed
  - Quantitative process management
  - Software quality management
- Level 5 – optimizing
  - Defect prevention
  - Technology change management
  - Process change management

- The Team Software Process (TSP) – a process framework for building and guiding engineering teams that develop software
  - Cornerstone – early defect removal
  - Based on the practices in the CMM
  - An instance of a Level 5 CMM process for a team
  - A prerequisite for each team member is the Personal Software Process (PSP)

Software Development Processes (con’t)

- Project launch
  - Team building
  - Requirements
  - Detailed plans
  - Quality plan
- Requirements
- High level design
- Implementation
- Integration and system test

- Extreme programming -12 practices
  - Unit tests are written prior to the functional coding
  - Pair programming
  - Refactoring continuously to simplify the design
  - Implement functionality only when you need it
  - Produce functionality tests to ensure that requirements and performance goals are being met

Software Development Processes (con't)

  - On-site customer resolves ambiguities, sets priorities, and helps with user testing
  - Continuous integration and testing
  - Continuous planning – prioritizing and addressing user stories
  - Overall architecture is defined to drive the small iterations and owned by the entire team
  - Coding standards are adopted across the team
  - Code is completed

- Test-driven Development
  - Automated tests are written that describe a capability
  - A minimal API is created so that the tests compile (they fail)
  - The API and underlying logic are implemented so tests pass
  - Refactor the code so that it is clean and as simple as possible
  - When all parts are integrated, follow the traditional system testing phase

The SQA Role

SQA In SDLC Processes

- Waterfall & Modified Waterfall (V or X test model)
  1. Review requirements for testability
  2. Review functional and design specs for completeness and testability
  3. Write and have reviewed test plans and test cases
  4. Perform all testing, bug reporting, metric collection when implementation is complete
  5. In a modified waterfall model:
     1. Write new tests if there is a redesign feedback loop or new functionality added
     2. Run new tests if a redesign feedback loop or new functionality added
  6. Release

SQA In SDLC Processes (con't)

- Concurrent Development/test
  1. Steps 1 & 2 as in the waterfall model
  2. Iterations of builds passed from Development to QA
     - Test organization writes test plans, test cases for functionality in a build
     - Test organization receives incremental builds/drops
     - Test organization tests the functionality in each build
  3. Development produces an integrated build containing all functionality
  4. Formal testing
     - Rerun all previously run tests
     - Performance testing
     - System testing
  5. Release

SQA In SDLC Processes (con't)

- Rapid Prototyping
  - Same steps as waterfall or modified waterfall process
  - Or, same steps as the concurrent development/test process
- Spiral - combines rapid prototyping and waterfall methodologies, adding the risk analysis component
  1. First circle – planning, risk analysis, prototype, customer evaluates prototype
  2. Second circle – more refined prototype, requirements documented and validated, customer assesses new prototype, risk analysis
  3. Third circle – further prototyping, design, integration and testing, risk analysis
  4. Step 4 of the waterfall model
  5. Release

SQA In SDLC Processes (con’t)

- The Unified Process
  1. Inception phase – establish business case through use case diagrams, define success critera, risk assessment, resource estimates, and a project plan
  2. Elaboration phase
     - Analyze domain, establish architecture, eliminate highest risks, enhance project plan
     - Mitigate risks – find ways of handling technical issues and business concerns with the prototypes
  3. Construction – develop the rest of the components, integrate, thoroughly test
  4. Transition – beta testing, release the product to the customers

SQA In SDLC Processes (con't)

- Agile – plan only what you need, doc what you need, test what you need
    1. Product planning – provide general targets for the project; basic requirements and basic project plan
    2. Increment planning (for each increment of 2 – 3 weeks)
        - Test planning only for the increment
        - Tight communication between testers and developers
        - Expect significant changes regularly (customer works directly with the team)
    3. Increment development
        - Developers responsible for testing functionality (unit tests)
        - Customers responsible for usability testing
        - Test engineers become expert consultants helping developers with functional test strategy and customers with acceptance test strategy
        - Test engineers plan for system testing including: security, safety, performance, reliability, installability, and maintainability
    4. Increment is delivered to the customer as working software
    5. Final increment delivery is the full product release (integration is as you go)

SQA In SDLC Processes (con’t)

- SCRUM
  - The process is an Agile process
  - Increments are called Sprints and are 30 days long
  - Daily SCRUMs are for communicating problems and getting team members unstuck
  - Co-located team members manage a backlog
  - Programmers and testers are brought together to share development

SQA in Development Processes

- CMMI
  - Level 2 – repeatable: Quality Assurance Group plans and implements the project’s QA activities to ensure the software process steps and standards are followed
    1. SQA Plan is written
    2. Review the development plan, standards and procedures
    3. Review software engineering activities for compliance
    4. Audit software for compliance
    5. Report results of activities
    6. Document and handle deviations in software activities and products
  - Level 3 – defined: Software Testing Group performs formal testing
    1. Testing criteria developed
    2. Effective methods are chosen to test the software
    3. Test readiness criteria established
    4. Regression testing
    5. Test plans, test procedures, & test cases undergo peer review
    6. Test plans, test procedures, & test cases managed and controlled
    7. Integration testing
    8. System and acceptance testing

SQA in Development Processes (con't)

- TSP
  1. Team sets quality goals and standards
  2. Personal design and code reviews
  3. Team inspects all work products
  4. Weekly tracking of quality indicators
  5. Team identifies and resolves quality problems
  6. Cycle and phase postmortems
  7. Team sends the product to the System Test group
- Extreme Programming
  1. Customer chooses the most valuable story (highest priority)
  2. Programmers break the story into smaller tasks
  3. Programmer pairs turn each task into a set of unit test cases
  4. Programmer pairs code to make the test cases run, evolving the system design to be as simple as possible
  5. All test code and story code is integrated into the the full system
  6. Customer functional tests are integrated as they are supplied
  7. At the end of each iteration, the customer's functional tests are run along with all unit tests

SQA in Development Processes (con't)

- Test-driven Development: define the system's parts that must be present to test the high risk components
  1. Write a master test plan that identifies and prioritizes the project level risks
  2. Define the pieces of the system that need to be tested earlier
  3. Developers build the high risk pieces first and they are then tested
  4. Units and modules are tested immediately after they are integrated
  5. Full system testing

SQA Project Management

SQA Project Management

- What’s your sphere of quality assurance?
  - Sphere 1: the quality of the software development process
  - Sphere 2: the quality of the software
  - Sphere 3: the quality of both the development process and the software

SQA Project Management (con't)

Sphere 1: The Process

- Your role
  - Determine whether the product development process has been followed
  - Report your results to senior management
- Assumptions
  - You are independent from the teams developing and testing the software
  - The product development process has been documented and teams are trained
  - You have the authority from senior management to do your job

SQA Project Management (con’t)

Sphere 1: The Process (con’t)

- Your activities
  - Write a quality plan containing:
    - Objectives
    - Resource requirements
    - Schedule of activities
    - Standards being used
    - Evaluations to be performed
    - Audits and reviews to be performed
    - Procedures for documenting and tracking non-compliance
    - Documentation that will be produced
    - Method and frequency of reporting
  - Perform your tasks
  - Report your results

SQA Project Management (con’t)

Sphere 2: The Software

- Your role
  - Propose and get agreement on the definition of software quality
  - Plan and execute the activities needed to assure the software meets the quality definition
  - Report your conclusions
- Assumptions
  - You are a member of the program/project leadership team
  - Product requirements are documented
  - Functional specifications are documented
  - There is a change management process for dealing with new, modified, and removed requirements
  - There is a defect tracking system in use

SQA Project Management (con't)

Sphere 2: The Software (con't)

- Your activities
  - Write a master project test plan containing:
    - Overview with the purpose of the product release, audience addressed by the plan, scope of the plan, references
    - Test strategy
      - Development phase entrance & exit criteria
      - Test infrastructure
      - Test environments
      - Resources
      - Automation
    - Test methodology
      - Requirements traceability
      - Types of testing
    - Schedule (who is doing what, when, % effort)
    - Metrics program including release criteria
    - Limitations and risks
    - Deliverables

SQA Project Management (con’t)

Sphere 2: The Software (con’t)

- Perform tasks
- Produce deliverables
- Participate in program/project meetings, assisting with decisions & providing data
- Participate in bug triage meetings
- Participate in phase exit reviews
- Participate in the go/no go decision for release

SQA Project Management (con't)

Sphere 3: Process and Software

- Your role
  - Determine whether the product development process has been followed
  - Report to program/project manager your results
  - Propose and get an agreement from the project/program team on the definition of the product quality
  - Plan and execute the defined tasks
  - Report conclusions
- Assumptions
  - Product development process is documented
  - You have buy-in from program/project team and senior management for your quality role
  - Product requirements are documented
  - Functional specifications are documented

SQA Project Management (con't)

Sphere 3: Process and Software (con't)

  - A change management process is in place
  - A defect tracking system is in use
- Your activities
  - Write a quality plan including the scope of activities to be performed for assurance
  - Write a master test plan describing the testing and release criteria
  - Perform tasks
  - Produce deliverables
  - Participate in program/project and bug triage meetings
  - Participate in phase exit reviews
  - Participate in the go/no go decision for the release
  - Report process assurance results

Case Studies

Case Studies

- 1992 – EASEL Corporation went from CMM Level 1 to Level 2 (repeatable)
  - A painful software product release was 3 months late
  - As QA Manager, I spearheaded the development and approval of a SDLC process (waterfall model)
- 1994 – EASEL Corporation ran the first SCRUM Agile project
  - Acquired the San Diego company who developed Smalltalk
  - Jeff Sutherland, “a father of SCRUM” was brought in as VP of Engineering and led us in the first SCRUM project, a Smalltalk release
- 1998 – Lernout & Hauspie Speech Products (L & H) hired a VP of Quality
  - VP worked to improve IT, business operations, and customer support processes
  - I was tasked with improving the software quality
  - A team at corporate headquarters in Belgium wrote a SDLC (modified waterfall model)

Case Studies (con't)

- 2002 – Nokia's multimedia messaging group created test processes and templates to be used across development teams in 3 sites
- 2003 – I worked with a process engineer in our division at Nokia and with a program manager in Helsinki to develop a solution SDLC process
  - Three products were integrated into the solution
  - 2 products used a concurrent development/test model and the 3rd used an iterative development model
- 2006 – former Funk Software group acquired by Juniper Networks went from CMM Level 1 to Level 2, repeatable
  - There were no processes, no test plans, no written test cases, no standards for requirements documents or functional specifications
  - In January 2006, a new Juniper SDLC process was rolled out
  - A project team consisting of teams in CA, MA, and India successfully released a large integration project using this SDLC with a modified waterfall model

Case Studies (con't)

- 2007 – our Juniper division added processes and templates to the corporate SDLC
  - A division-wide team was established to better fit the corporate SDLC to software products
  - A modified waterfall model was used
- 2008 – Juniper created a new business unit to produce a product which was an integration of a software product in Cambridge with a software product in Canada to be delivered on the same hardware platform
  - The MA project team regularly used a concurrent development/test model
  - The Canadian team regularly used a strict waterfall model
  - I worked with the Canadian SQA Manager to develop some processes, templates, checklists, and role definitions to be shared
  - The Engineering Manager was located in Canada and owned all of the Development and SQA resources on the project
  - All project members were required to use a strict waterfall model

What are Your QA Project Management Challenges?

References

References

Watts S. Humphrey. Managing the Software Process. 1989. Addison-Wesley. Reading, MA

Steven R. Rakitin. Software Verification and Validation for Practitioners and Managers, Second Edition. 2001. ARTECH HOUSE, INC. Norwood, MA

Ivar Jacobson, Grady Booch, and James Rumbaugh. “The Unified Process”, IEEE Software. May/June 1999

Jennitta Andrea. “Piecing Together the Truth About TDD”, Better Software. January/February, 2008

Alan S. Koch. “The Role of Testers in the Agile Methods”, Software Quality Progress. VOL. 7, NO. 3/2005, ASQ

Ken Schwaber. “Scrum: It Depends on Common Sense”, Cutter Consortium. 2004

References (con't)

Kent Beck. "Embracing Change with Extreme Programming", Computer. IEEE October 1999

Robin Goldsmith and Dorothy Graham. "Test-Driven Development", Software Development. October 2002

Gertrud Bjornvig, James O. Coplien, and Neil Harrison. "A Story About User Stories and Test-Driven Development", Better Software. December 2007

Carnegie Mellon University Software Engineering Institute. <u>The Capability Maturity Model: Guidelines for Improving the Software Process</u>, 2000. Addison-Wesley. Reading, MA

James W. Over. "Quality Management in the Team Software Process (TSP)", Software Engineering Institute. 2001. Carnegie Mellon University