Project Management

Quality Assurance

Week 5 Winter quarter

02/04/02 SOS

Objectives

- Understand what Quality Assurance is
- Be able to identify and track defects
- Become familiar with Bugzilla as a defect tracking tool

Quality Assurance

- **quality assurance**
  - <testing> (QA) A planned and systematic pattern of all actions necessary to provide adequate confidence that the product optimally fulfils customers' expectations, i.e. that it is problem-free and well able to perform the task it was designed for.

Quality Assurance

- Steps
  - unit testing
  - reviews/walkthroughs
  - system testing
  - defect tracking

Quality

- What level of quality are you striving for:
- What is the likelihood and consequence of errors
- One way to categorize
  - life critical - like the space program
  - near life critical - lab information system
  - crucial - financial transactions
  - close enough - duplicate mailings

Quality

- Objective - easily measurable (requirements traceabilty)
- Subjective - *fulfills customer expectations*
- Relates back to having very clear goal, scope and criteria for success statements.

Quality Assurance Plan

- Must plan for QA
- Plan contains measurable criteria that are used to determine whether software is ready to release
- Most important - start at the beginning of the lifecycle and commit the plan to writing. *Content more important than format.*

Scheduling for testing

- Duration versus effort
- examples:
  - iCopyright roughly 8 testers to twenty something developers - initially duration of 1 week of development to 1 week of testing and bug fixes
  - Life critical (10 testers per developer)
  - Business applications (1 tester per 3 or 4 developers)

System Testing

- Purpose
  - Find examples where actual behavior deviates from expected behavior:

Types of testing

- Functional testing
  - does this work according to specifications
- Boundary testing
  - what happens at boundary points
- Performance testing
  - how long for a page to load, a report to generate

Types of Testing

- Ad Hoc testing
  - Testers try to find ways to make the system crash.
- Acceptance Testing
  - customer test to verify that specifications and expectations have been met.
- Alpha/Beta Testing
  - customer performs test and verifys that product meets specifications and expectations

Types of Testing

- Stress testing/Load testing/Performance testing
  - what happens when many simultaneous transactions are occurring, or many records are being added
  - peak versus average loads
  - capacity of system

Types of Testing

- Regression testing
  - <programming> Part of the test phase of software development where, as new modules are integrated into the system and the added functionality is tested, previously tested functionality is re-tested to assure that no new module has corrupted the system.

Life Cycle of a bug

- Reporting the behavior
- Confirming the behavior
- Classifying the bug
- Assigning the bug
- Accepting the bug or re-assigning
- Resolving the bug
- Verifying the resolution
- Closing the bug

Creating a Test Plan

- Assume a search engine
- Create metrics - how many successful transactions will give a reasonable level of confidence in quality
  - How will you know that that many transaction have occurred?

- Boundary testing
  - Need test cases that produce zero results, 1 results, 2 results, exactly one page of results, one page plus 1, one page minus one, several pages
- Load Testing
  - how many simultaneous users; what type of transactions (read/write)

- Conditions
  - operating systems,platform, browser and browser version, modem speed
  - how many combinations!
- Capacity
  - how records are in database when searching
- Speed
  - how long to load a page

Creating a test Plan

- Assume an e-commerce site
- Determine metrics - how many successful transactions versus total transactions
- What are the boundary conditions
- What type of error messages could you receive
- What type of credit cards
- How many transactions with money deposited in bank

Reporting a bug

- Summary - a title - quick overview
- Steps to reproduce
- Expected behavior
- Actual behavior
- Other info: OS, Platform, browser version, software version, component, url
- Reproducible?

Classifying a bug

- **Using a Bug Council**
- **triage** n : sorting and allocating aid on the basis of need for or likely benefit from medical treatment or food
- Assigning priority, severity, developer
- Assigning fix to a scheduled release
- Enhancement Request versus a Defect

Resolving a Bug

- **FIXED**
  - A fix for this bug has been checked into the tree and tested by the perso marking it `FIXED`.
- **INVALID**
  - The problem described is not a bug.
- **WONTFIX**
  - The problem described is a bug which will never be fixed, or a problem report which is a "feature", not a bug.

Resolving a Bug

- **DUPLICATE**
  - The problem is a duplicate of an existing bug. Marking a bug duplicate requires the bug number of the duplicating bug and will add a comment with the bug number into the description field of the bug it is a duplicate of.
- **WORKSFORME**
  - All attempts at reproducing this bug in the current build were futile. If more information appears later, please re-open the bug, for now, file it.

Bad Bug / Good Bug

- BAD: "My browser crashed. I think I was on foo.com. I think that this is a really bad problem and you should fix it or else nobody will use your browser. By the way, my sister thinks your icons really suck. Oh, and my mom's home page doesn't right, either, it's all messed up. Thx 4 fixing theze bugz
-
- GOOD: "I crashed each time when I went to foo.com, using Mozilla on a Win NT 4.0 (Service Pack 5) system. The build ID is 20000609. I also rebooted into Linux, and reproduced this problem using the 20000608 Linux build.Examples of a good bug

Configuration Management

- Source Control
  - CVS - open source
  - VSS - MS Visual Source Safe

Release Management

- Release Candidate/ Gold Candidate
- Release Management
  - write up release notes
  - includes existing functionality and known bugs
  - supports what operating systems, browsers, platforms

Project Closeout

- Post Mortem
  - What went well
  - What didn't go well
  - Lessons learned