SESSION E-QUAL

Quality Assurance for Database Applications

Whil Hentzen

Hentzenwerke Corporation

whil@hentzenwerke.com

www.hentzenwerke.com

Session Goals

- “Quality” Definition
- Quality Control vs. Quality Assurance
- Integrating Quality Assurance into the Development Process
- Specific QA Techniques

Who...

- Independent Xbase Developer since ‘83
- Hentzenwerke Corp - Current Projects
- Four Books, including DevGuide ‘97
- Editor of FoxTalk
- Wonderful Wife & 3 Kids (so far)

Session Requirements

- Programming Experience Not Required
- Corporate or Independent

The Manufacturing Paradigm

- Creating software has many similarities to manufacturing durables
  - Steps are the same
- Differences are due to perception
  - Not a physical object

Definition of Quality

- “Fitness for Use”

QC vs. QA

- Quality Control comes at the end of the process
- Quality Assurance is part of the process

What is the System Intended to do?

- This is the fundamental issue
- Can’t determine Fitness For Use without knowing what the “Use” is

Difficulties in Determining System Definition

- Changing requirements from users
- Unwillingness for developer to commit to functionality
- Specification goes against the nature of developer as artist

System Specifications

- Must be written down
- Must be updated
- Types of documentation
  - Technical specifications
  - Description of unit functionality
  - Use cases

Technical Specifications

- Database Structures
  - Databases
  - Tables
  - Rules/Validations
- Class Designs
  - Hierarchy
  - Use

Unit Functionality

- Screen Appearance
- Purpose
- Access
- Step-by-Step Usage
- Rules

Use Cases

- Defining a set of Use Cases
- Done during initial specification
- Refined through specification development

Use Case Bullet Points

- How Many?
- First level of description
- Define major functions
- Example:
  - 1. Quote Generation
  - 2. Quote Status Tracking
  - 3. Tickler File

Bullet Point Breakdown

- Functions that make up a Bullet Point
- Quote Generation Example:
  - Create a Quote
  - Duplicate a Quote
  - Merge Multiple Quotes
  - Quote Output

Variations on Bullet Point Breakdown = Use Cases

- Specific function or purpose that user wants to accomplish
- Create Quote Example:
  - Create a quote with one standard line item
  - Create a quote with multiple standard line items
  - Create a quote with one made-to-order line item

Documenting a Use Case

- Series of steps to follow
- Steps include entire process, not just the computer part
- Can be validated through a checklist

Code Reviews

- Types of code reviews
  - Code Inspection
  - One-on-one code review
  - Group code reviews
- Purposes
  - Not to find bugs
  - Mechanism for sharing techniques
  - Encourage proper practices
  - Encourage maintainability

Code Inspection

- A “Quality Control” function
- Done on One-On-One basis
- Who gets inspected?
- What gets inspected?
- Who does inspection?

One-on-One Code Review

- Purpose: Mentoring
- Who gets reviewed?
- What gets reviewed?
- Who does reviewing?

Group Code Review

- Purpose:
  - “Defending Your Life”
  - Encourage interaction
- Who gets reviewed?
- What gets reviewed?
- Who does reviewing?

Implementing a Review

- Obstacles to doing reviews
  - No format
  - Not a deliverable for a customer
  - Busy, busy, busy

Review Checklist

- Building a set of guidelines from scratch
- Make guidelines available to entire group
- Make guidelines easy to follow - 300 pages of guidelines is too much
- These are guidelines, not rules

Review Guidelines - I

- Does the system compile without errors?
- Has any source code been stubbed out?
- Does the project contain unused items?

Review Guidelines - II

- Does each routine have a proper header?
- Are parameters described in the header?
- Are parameters checked upon entry?
- Do comments exist? Any source code commented out? Is code indented and formatted with white space?
- Are logic structures complete? (IF/ELSE)
- Do variables follow naming conventions?
- Are there long/complex nestings?
- Are there comments for modified code?

Review Guidelines - III

- String comparisons handled with case?
- Are file locations hard-coded?
- Are similar functions used interchangably (DTOS and DTOC)?
- Does arithmetic on dates handle Y2K?
- Are divisors tested for zero?
- Are variables declared and initialized?
- Do “magic numbers” exist?
- Are there blocks of repetitious code?

Review Guidelines - IV

- Can custom code be handled by libraries?
- Is code in the proper place in the call stack?
- Are variables scoped and released?
- Are common cases tested first?
- Do routines have one exit?
- Is code contained within loops necessary?
- Are external device calls trapped?
- Are files checked for existence?
- Is environment returned to original state?

Review Deliverable

- Doesn’t get done because no one is waiting for it
- Deliverable must be to QA department
- “Code that is not Reviewed is Code that is not Finished”

Not Enough Time

- Part of a review can be a clerical function
- Do 10% of the application
- Review during breaks
- Review during downtime
- Measurable goal of developer
- Discipline

Test Data

- Four required items
  - Multiple data sets
  - Original data
  - Proof cases
  - Multiple snapshots

The Testing Process

- 1. Push all the buttons
- 2. Test unit functionality
- 3. Test the rules
  - Unit rules
  - Use cases
- 4. Break it!

Test Plans

- “Plan Your Work and Work Your Plan”
- Documenting what was tested
- Regression Testing

Testing Personnel

- Who to test?
  - The worst person to test
  - The second worst person to test
- This leaves dedicated staff

Internal Testing - Pros

- Testing gets done
- Testing is done properly and consistently
- Training to test not needed
- Dedicated individuals develop skill at testing and continually get better
- Special tools and techniques are available

Internal Testing - Cons

- Higher expense that may not be able to be billed to customer
- Scheduling keep work load even
- Availability of testing personnel
- Additional overhead of managing testing personnel
- Learning curve for application can't be re-used

External Testing - Pros

- Lower perceived outside cost
- Burden to find personnel placed elsewhere
- Learning curve of testing personnel is a useful investment
- Only one transfer of knowledge necessary - less expensive

External Testing - Cons

- Getting customer to agree to test (That’s the realm of the developer)
- Testing not done on time
- Accuracy of testing results
- Skill level of testing personnel is low

Application Feedback

- When to start documenting bugs
  - Bugs can fall through the cracks internally as well as externally
  - Defects caught internally can provide valuable feedback as well (the manufacturing QA paradigm)

Categorizing Defects

- Analysis
- Design
- Coding
- Environmental
- Installation
- Training
- Data
- Can’t reproduce

Tracking Defects

- Require a form to be filled out
- Three types of feedback
  - Defect
  - Question
  - Enhancement Request (ER)
- Issues list

Acceptance

- Definition of Acceptance
- Delivery
- Additional examination time
- Final payment

More Info

- Samples on www.hentzenwerke.com
- Books - MSPress, DevGuide