Software Quality Group of New England

Develop Extraordinarily Powerful Test Scripts

December 14, 2011
Joseph Zec
Sr Manager, Corp SQA
Boston Scientific Corp

1

Agenda

- The mechanics of an auditable test script
- Measures of the thoroughness of a set of test cases
  - Requirements coverage
  - Test type coverage
    - Unusual and unexpected conditions
  - Structural coverage
- Q & A

2

The mechanics of test scripts

- Powerful test cases are wasted if their test scripts are not auditable
- Specific information needs to be captured both prior to and during test execution
- Use of a test script template will facilitate the consistent recording of test data

3

Let’s start with the basics

- What is being tested?
  - Purpose of test / Requirement IDs
  - Version of software under test
  - Configuration of test environment
- Pre-testing conditions
  - Scripts that need to be run in sequence
  - Set-up instructions
  - Any required test data
  - Any dependencies

4

The script itself

- Step number
- Action
- Expected result
  - Needs to be detailed
  - Needs to be unambiguous

An essential element of a software test case is the expected result. It is the key detail that permits objective evaluation of the actual test result.

5

Data to capture during testing

- Date of testing
- Name / signature of tester
- Actual result
  - Same rules apply as for the expected result
- An answer to the question, “Does the actual result match the expected result?”
  - Should be answered “yes/no”, not “pass/fail”
- Issue IDs
  - Only link from test script to issue tracker

6

Test script example

Step #	Test Step Description	Expected Result	Actual Results/Comments	Actual Results = Expected Results? (Yes/No)	Issue Tracking #
	Login to Adjudication System				
1.0	Open an Internet browser window and paste the following URL into the Address field: https://www.bostonscientific-cectest.com. Press ***Enter***.	The Adjudication Log-in page should be displayed			
1.1	Enter the Lead Adjudication Manager (LAM) Login ID and Password for your assigned user account. Login ID: < **LAM Login ID**> Password: <**LAM Password**> Press **Login.**	The Event Work Queue page should be displayed			

7

Why is thoroughness in testing important?

- It is the purpose of software testing to uncover defects
- Defects in software most often exist where the application is subjected to unusual and unexpected conditions
- Therefore, test cases must thoroughly probe unusual and unexpected conditions to be successful

8

Some Best Practice on this subject

- *A good test case has a high probability of exposing an error*
- *A successful test is one that finds an error*
- *Examining only the usual case is insufficient*
- *Software testing that finds no errors should not be interpreted to mean that errors do not exist in the software product; it may mean the testing was superficial*

9

How do you know you have a thorough set of test cases?

- To start with, make sure that every requirement has at least one test associated with it (requirements coverage)
  - Usually demonstrated via a trace matrix
  - Is this sufficient?
- "But I don't have any requirements for my project"
  - Abandon all hope, ye who enter here

10

Example of simple requirements coverage

- Requirement
  - The application shall allow an authorized user to disable a user
- Test case
  - Log onto the system as a user administrator
    - ❖ Verify the “Disable User” button appears
  - Select User1 and click “Disable User”
    - ❖ Verify User1 is now disabled

11

Okay, but what about “unusual and unexpected” test cases?

- Requirement
  - The application shall allow an authorized user to disable a user
- Test cases

12

Okay, but what about “unusual and unexpected” test cases?

- Requirement
  - The application shall allow an authorized user to disable a user
- Test cases
  - Make sure unauthorized users can’t disable a user

13

Okay, but what about “unusual and unexpected” test cases?

- Requirement
  - The application shall allow an authorized user to disable a user
- Test cases
  - Make sure unauthorized users can’t disable a user
  - Try to disable an already disabled user

14

Okay, but what about “unusual and unexpected” test cases?

- Requirement
  - The application shall allow an authorized user to disable a user
- Test cases
  - Make sure unauthorized users can’t disable a user
  - Try to disable an already disabled user
  - Try to click the button without selecting a user first

15

How to develop thorough unusual and unexpected cases

- Software test cases can be categorized by type
- Thoroughness can be achieved by including as many different test types as are applicable
- This is known as “test type coverage”
- “Disabling User1” is known as a “normal case” test type, for instance

16

So what are these test types anyway?

- Common test case types
  - Functional (or normal)
  - Output forcing
  - Robustness
    - Error testing
    - Negative testing
    - Performance testing
    - Volume testing
    - Stress testing
    - Boundary testing
  - Combination of inputs
    - Operational testing

17

Structural coverage is another method for assuring thorough test cases

- Structural coverage is only applicable when you have access to source code that has been instrumented by a coverage analyzer tool
- This is an extremely powerful method but is technically challenging and labor and time intensive
- There are many ways of measuring structural coverage

18

Structural coverage examples

- Statement coverage
- Decision (branch) coverage
- Condition coverage
- Multi-condition coverage
- Loop coverage
- Path coverage
- Data flow coverage

19

Summary

- A trace matrix is the foundation of thorough testing
- Expand basic test cases by covering as many different applicable test types as possible
- Add structural coverage analysis of test cases
- The result? – A thorough set of test cases with the power to uncover any bug!

20

Conclusion

Q & A

21

Thank you!

Joseph Zec
Corporate Software Quality Assurance
Boston Scientific Corp
Joseph.Zec@bsci.com

22