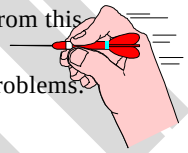
Surviving the Top Ten Challenges of Software Testing: A Closer Look at Understanding Software Testing

Randy Rice, CQA, CSTE
Rice Consulting Services, Inc.
405-692-7331
http://www.riceconsulting.com
rcs@telepath.com

Goal of This Workshop



- Learn how to get the most from this conference
- Examine common testing problems.
- Learn:
  - Basic testing terminology
  - The economics of testing
  - Effective testing methods and techniques
  - How to develop and customize a test strategy to your organization

Your Job

- Interact
- Reflect
- Apply the concepts to your situation

Common Testing Problems

- What are your most pressing testing problems?

The Top 10 Testing Problems

- Identified by surveys and interviews over the past three years
- Over 1,000 testers surveyed
- Basis for the book,*Surviving the Top Ten Challenges of Software Testing* by William E. Perry and Randall W. Rice

The Top 10 Testing Problems

10. Not enough training
9. "Us vs. Them" mentality
8. Lack of test tools

The Top 10 Testing Problems

7. Lack of management understanding/support of testing
6. Lack of customer and user involvement
5. Not enough time for testing

The Top 10 Testing Problems

4. Over-reliance on independent testers
3. Rapid change
2. Testers are in a "lose/lose" situation
1. Having to say "no"

Solutions for Training

- Obtain formal training in testing techniques.
- Seek Certification.
  - CSTE (Certified Software Test Engineer)
- Attend conferences.
- Read books and articles.

Solutions to the Teamwork Challenge

- The goal is to get to "Us and them."
- Each person on the team can have a role in testing:
  - Developers: unit and structural testing
  - Testers: independent testing
  - Users: business-oriented testing
  - Management: to support testing activities

Solutions for Acquiring and Using Test Tools

- Identify a "champion" for obtaining test tools.
- Base the case for test tools in costs vs. benefits.
- Have a basic testing process in place.
- Train people in tool usage.
- Measure the benefits.

Solutions to Educating Management in Testing Issues

- Cultural change is needed.
- Focus your message to management on:
  - reducing the cost of rework
  - meeting the project schedule
- The benefits of testing must relate to these two things to be persuasive.

Solutions to Identifying and Involving the Customer in Testing

- Involve the customer and users throughout the project by performing reviews and inspections.
- Include users on the system test team.
- Perform user acceptance testing.
- Understand the difference between the customer and users.

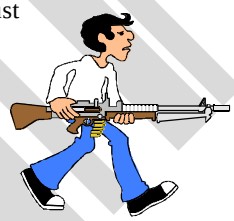
Solutions to the Time Crunch

- Base schedules and estimates on measurable testing activities.
  - Scripts to be executed
  - Cases to be tested
  - Requirements to be tested
- Have contingency plans for schedule slippage.
- Integrate automated testing tools to the project.

Solutions to Overcoming Throwing Stuff Over the Wall

- Developers must take ownership and responsibility for the quality of their work.
- Quality control is most effective when performed at the point of creation.
- Train developers to become excellent testers.
- Get management support for developer responsibility for quality.

Solutions for Hitting a Moving Target



- The testing process must accommodate change.
- Focus on testable requirements.
- Use automated testing tools.
- Manage the rate and degree of change.

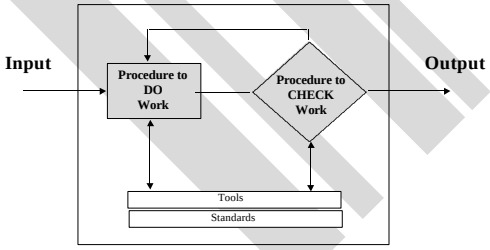
Solutions for Fighting a Lose-Lose Situation

- The perception of testing must change.
  - Testers are paid to find defects.
  - Each defect found is one more the customer or user will not find.
- Testers are not to blame for bottlenecks. It is management's responsibility to have an efficient process.

Solutions for Having to Say "No"

- Most responsibility is on management to:
  - have a quality software development process in place.
  - have contingency plan in place in case of problems.
  - understand that testing is only an evaluation activity.
  - accept the honest facts.
- Keep the test results objective.

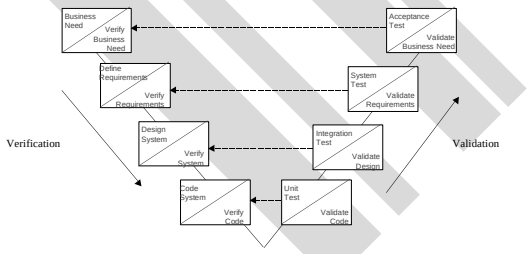
QAI Workbench Model



Test Terminology (Cont'd.)

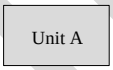
- Verification
  - All QC activities throughout the life cycle that ensure interim deliverables meet specific specifications.
- Validation
  - The "test phase" of the life cycle which ensures that the end product (e.g..., software or system) meets specifications or user needs.

Test Terminology (Cont'd.) When Testing Occurs



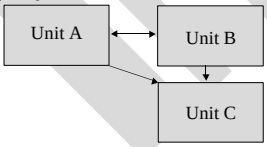
Test Terminology (Cont'd.)

- Unit Testing
  - Testing performed on a single, stand-alone module or unit of code.



Test Terminology (Cont'd.)

- Integration Testing
  - Testing performed on groups of related modules to ensure data and control are passed properly between modules.



Test Terminology (Cont'd.)

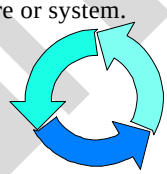
- System Testing
  - A predetermined combination of tests that, when executed successfully, satisfy management that the system meets specifications
  - Validates that the system was built right.

Test Terminology (Cont'd.)

- User Acceptance Testing
  - Testing to ensure that the system meets the need of the organization and the end user/customer
  - Validates that the right system was built.

Test Terminology (Cont'd.)

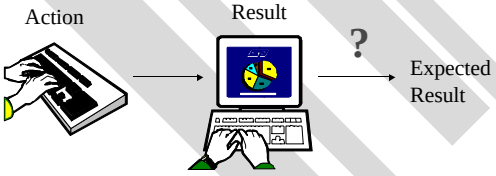
- Regression Testing
  - Testing after changes have been made to ensure that no unwanted changes were introduced to the software or system.



Test Terminology (Cont'd.)

- Functional Tests
  - Tests that validate business requirements
  - Tests what the system is supposed to do
- Black Box Tests
  - Functional testing
  - Based on external specifications without knowledge of how the system is constructed
  - Usually process and/or data driven

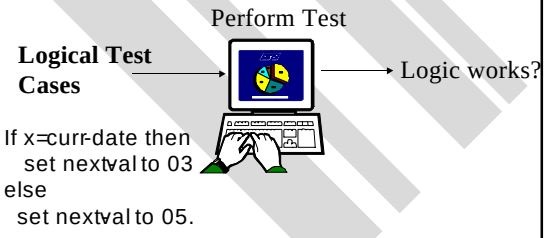
Test Terminology (Cont'd.) Functional Testing



Test Terminology (Cont'd.)

- Structural Tests
  - Tests that validate the system architecture
  - Tests how the system was implemented
- White Box or Glass Box Tests
  - Structural testing
  - Testing based on knowledge of internal structure and logic
  - Usually logic driven

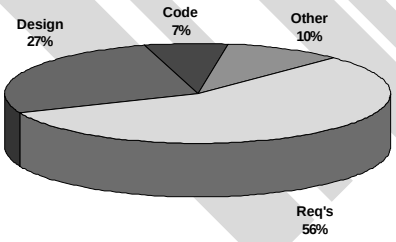
Test Terminology (Cont'd.) Structural Testing



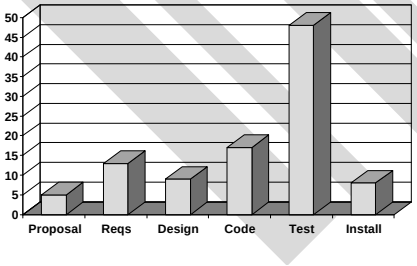
To effectively test systems, both functional and structural testing need to be performed.

The Economics of Testing - Making the Message to Management

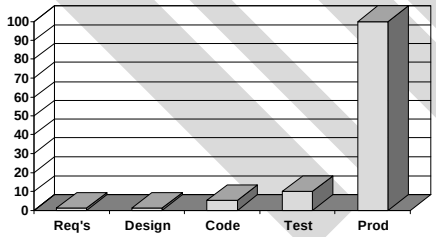
Where Defects Originate



Where Testing Resources are Used



The Relative Cost of Fixing Defects



The Bottom Line

- Most defects are created in the early stages of a project
- Most defects are found in the later stages of a project
- It costs 10 to 100 times as much to fix a defect in the later phases of a project.

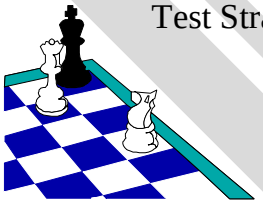
The Bottom Line

- If you want to reduce the cost of testing, spend time early in the system development (or purchase) process to make sure the requirements and design are correct.

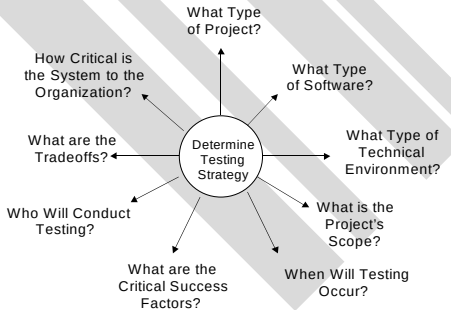
Basic Testing Principles

- Test early and often
- Involve everyone on the project
- Management support is critical
- The greater the risk, the more intense the test should be
- The higher the test coverage, the more confidence you'll have in the test

Test Strategy



Planning Step 1 - Determine Test Strategy

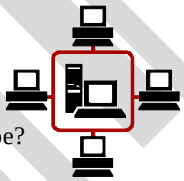


Test Strategy

- What Type of Project?
  - Traditional
  - Prototyping/CASE
  - Maintenance
- What Type of Software?
  - On-line
  - Real Time
  - Batch

Test Strategy

- What Type of Technical Environment?
  - Mainframe
  - Client/Server
- What is the Project's Scope?
  - New Development
  - System Maintenance



Test Strategy

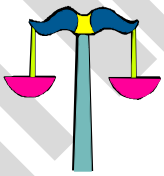
- When Will Testing Occur?
  - Requirements
  - Design
  - Testing
- What Are the Critical Success Factors?
  - Correctness
  - Reliability

Test Strategy

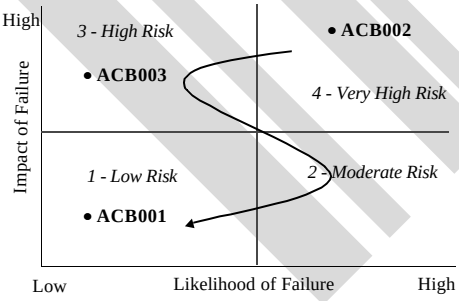
- Who Will Conduct Testing?
  - Users
  - Developers
- What Are the Tradeoffs?
  - Schedule
  - Cost/Resources
  - Quality

Test Strategy

- How Critical is the System to the Organization?
  - Risk Assessment



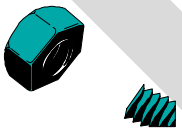
Risk Assessment



A Tool For Performing Risk Assessment

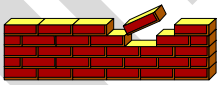
Recovery Triage Assessment												
#	Business Process/System/Function	Criticality to the organization's mission	Criticality and sensitivity to well-being, safety, or interest of general public, client, and customers	Criticality and sensitivity of data and information for: competitive advantage; customer confidence;	Fraud potential	Degree of dependance on system	Criticality of external interfaces with other systems or organizations	Size of user area affected	Availability and adequacy of backup and recovery procedure	Margin for error (i.e., is there reasonable time to make adjustments and corrections before the	Total score for process/function
1	Billing	5	3	3	5	3	3	1	1	3	27
2	Accounts Payable	3	1	1	5	1	3	1	1	3	19

Effective Testing Methods and Techniques



The QAI Testing Process

- Step 1 - Set Test Objectives
- Step 2 - Develop Test Plan
- Step 3 - Execute Tests
- Step 4 - Evaluate/Report Test Results

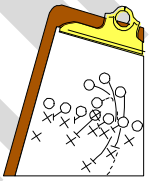


Step 1 - Set Test Objectives

- Select test team
- Perform risk assessment
- Define test objectives
  - A test objective is what the test is to validate.
  - There should be a one-to-one correspondence between system objectives and test objectives.

Step 2 - Develop Test Plan

- The better the test plan, the easier the test.
- Test plans should be specific, yet flexible for change.
- Test planning should be a team activity.
- Test plans should be reviewed just as any other project deliverable.
- The test plan should be easily read by management.



Major Elements of a Test Plan

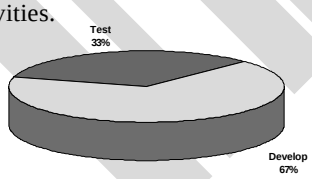
- Introduction
- Approach (Strategy)
- Test Objectives
- Description of the system or software to be tested
- Test environment
- Description of the test team
- Milestones/Schedule

Major Elements of a Test Plan (Cont'd.)

- Functions and attributes to be tested
- Evaluation criteria
- Data recording
- References
- Tests to be performed

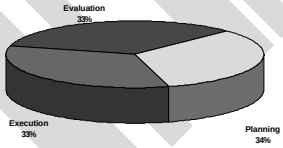
How Much Time Should be Spent on Test Planning?

- Many organizations report spending one-third to one-half of a project's time in test-related activities.



Planning Time Guidelines

- Of the total test time, roughly one-third of the time can be allocated each to:
  - Test planning
  - Test execution
  - Test evaluation



Tips for Test Planning

- Start early.
- Keep the test plan flexible to deal with change.
- Frequently have the test team review the test plan.
- Keep the test plan concise and readable.

Step 3 - Execute Tests

- Select test tools
- Develop test cases
- Execute tests

Step 3 - Execute Tests
Select Test Tools

- A test tool is any vehicle that assists in testing.
- May be manual or automated

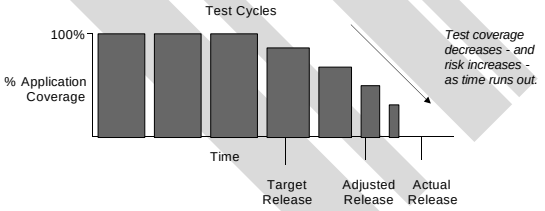
Automated Tools

- Not the complete solution, but an important part
- Requires:
  - a process
  - an understanding of testing in general
    - » Knowing what to test
    - » Defining test cases
    - » Knowing how to evaluate test results
  - cultural acceptance
- More than just capture/playback

Categories of Automated Tools

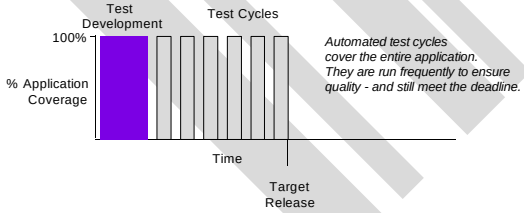
- Capture/playback or script execution
- Defect trackers
- Test management
- Test case generators
- Coverage analyzers
- Path and complexity analyzers

Manual Testing



Graphic courtesy of Rational Software, Burlington, MA

Automated Testing



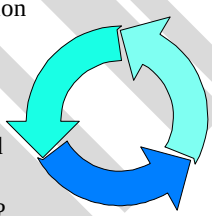
Graphic courtesy of Rational Software, Burlington, MA

Critical Success Factors

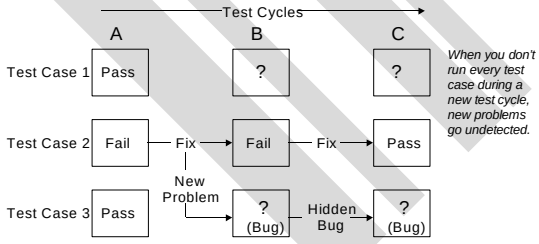
- Get senior management support for buying and integrating test tools
- Know your requirements
- Be reasonable in your expectations - Start small and grow
- Have a strong testing process that includes tools
- Don't cut the training corner

Regression Testing

- Why perform regression testing?
- The process
- The issues
- The role of automated testing tools
- How much is enough?

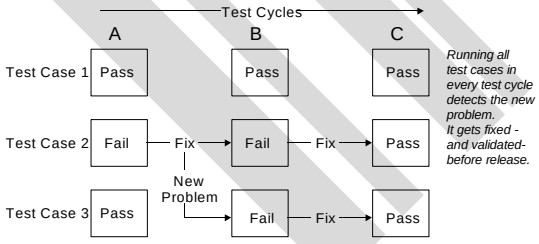


No Regression Testing: Hidden Defects



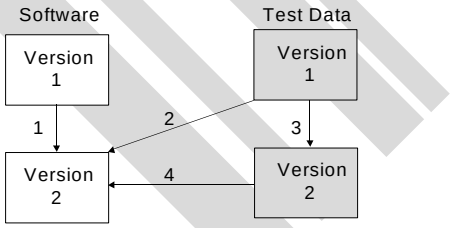
Graphic courtesy of Rational Software, Burlington, MA

Regression Testing: No Hidden Defects



Graphic courtesy of Rational Software, Burlington, MA

Regression Testing - The Process



Regression Testing Issues

- Test data must be maintained.
- There must be a way to conduct two *identical* tests.
- There must be a way to compare two identical tests.
- Some tests cannot use previous versions of test data.
  - Data conversion may be required
  - Date dependencies

Regression Testing Issues

- The greater the difference between versions, the less effective the regression test.
- There *must* be a stable baseline version for comparisons.

Regression Testing - How Much is Enough?

- The easy answer: "It depends."
- What does it depend on?
  - Risk
  - Scope of the change
  - System dependencies

Proving the Value of Regression Testing

- You need a benchmark of non-regression testing.
- Consider manualvs. automated.
- Consider initial investment in creating test environment and test scripts/procedures
- Measure time and defects.

Proving the Value of Regression Testing

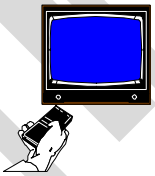
- Return on Investment (ROI) can include:
  - Shorter test times
  - More accurate testing
  - More consistent testing
  - Improved communication of defects
  - More effective testing (e.g. Fewer test cases needed to find more defects.)

Tips for Performing Regression Testing

- Control the scope of testing.
- Build a reusable test bed of data.
- Use automated tools.
- Base the amount of regression testing on risk.
- Build a repeatable and defined process for regression testing.

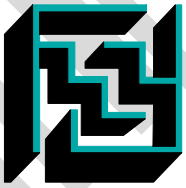
Step 3 - Execute Tests Develop Test Cases

- Functional Techniques
  - Requirements-based
  - Process-based
  - Data-oriented
  - Boundary value analysis
  - Decision tables
  - Equivalence partitioning



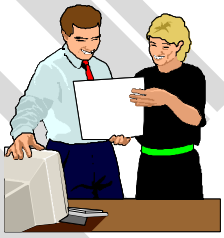
Step 3 - Execute Tests Develop Test Cases

- Structural Techniques
  - Complexity analysis
  - Coverage
    - » Statement
    - » Branch
    - » Condition
    - » Multi-condition
    - » Path



Step 4 - Evaluate/Report Test Results

- Occurs throughout the testing life cycle
- Tracks testing progress
- Keeps management informed of testing progress



Valuable Test Metrics

- Two key areas to measure
  - Time
    - » For future estimating
  - Defects
    - » For determining effectiveness of testing
    - » For improving the development and testing processes

Valuable Test Metrics

- Time
  - Time per test case
  - Time per test script
  - Time per unit test
  - Time per system test
- Sizing
  - Function points
  - Lines of code

Valuable Test Metrics

- Defects
  - Numbers of defects
  - Defects per sizing measure
  - Defects per phase of testing
  - Defect origin
  - Defect removal efficiency

Valuable Test Metrics

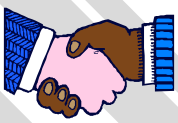
Defect Removal Efficiency =

Number of defects found in producer testing

Number of defects during the life of the product

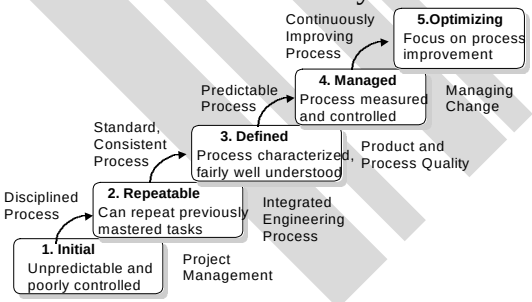
The Critical Path of Testing

What Must be in Place for Effective Testing?



- Management support
- A defined and repeatable process for testing
- Adequate tools
- Trained testers
- Cooperation between testers, developers and end users
- Maximum coverage with minimal test cases

The Five Levels of Software Process Maturity



Best Practices for Control/Test Management Processes