Implementing A Final Regression Testing Process

Jeff Tatelman
SQuAD Conference 2007
March 15th, 2007
Final Regression Testing Definition

The testing of software after a modification has been made to ensure that preexisting code is still working properly
  • Verification that what once worked still does
  • No new defects have been introduced
  • The “build” was created correctly

Completed after Function and End to End Testing
Adapting a Model

A framework that describes the key elements of an effective final regression testing process ...

An evolutionary improvement path from an adhoc state to a mature disciplined process.
Build Manual Tests

- Repeatable/Reusable
- Criticality to the business (Risk Based)
- Validate against requirements
- Stored using a Test Management tool
  - Capable of execution by multiple people
  - Results are visible
Example of an Inventory Control System
# Risk Based Testing

<table>
<thead>
<tr>
<th>Req ID</th>
<th>Business Requirement Name</th>
<th>Risk</th>
<th>END 1</th>
<th>END 2</th>
<th>END 3</th>
<th>END 4</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0.</td>
<td><strong>Customer Service</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1</td>
<td>Add A New Customer</td>
<td>H</td>
<td>X</td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.1</td>
<td>Select New Customer option</td>
<td>M</td>
<td></td>
<td>X</td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.1.2</td>
<td>Enter Required information</td>
<td>H</td>
<td>X</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>1.2</td>
<td>Delete A Customer</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>1.3</td>
<td>Modify Existing Customer</td>
<td>M</td>
<td>X</td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>2.0.</td>
<td><strong>Order Processing</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>2.1</td>
<td>Add an Order</td>
<td>H</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2.2</td>
<td>Modify an Order</td>
<td>M</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>2.3</td>
<td>Hold an Order</td>
<td>M</td>
<td></td>
<td></td>
<td>X</td>
<td></td>
</tr>
<tr>
<td>3.0.</td>
<td><strong>Inventory Management</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>3.1</td>
<td>Replenish Inventory</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>3.2</td>
<td>Audit Inventory Stock On Hand</td>
<td>M</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4.0.</td>
<td><strong>Shipping</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>4.1</td>
<td>Bill A Customer</td>
<td>H</td>
<td>X</td>
<td>X</td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>4.2</td>
<td>Ship To All Customers</td>
<td>H</td>
<td>X</td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.0.</td>
<td><strong>Print Listings and Reports</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>5.1</td>
<td>Order Review Report</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
<tr>
<td>5.2</td>
<td>Inventory Report by Material</td>
<td>L</td>
<td></td>
<td></td>
<td></td>
<td>X</td>
</tr>
</tbody>
</table>
Traceability Matrix Advantages

- Validation of requirements
- Risk based testing
- Ensure coverage
  - Knowledge of what is being testing
  - Knowledge of what is not being tested
- Reduce the number of test cases
- Trace changes:
  - Requirements to testing
  - Testing to requirements
- Manage regression for every release
Build Environment

- Isolated just for final regression testing
- Simulation of production instance
- Initial data setup
- Code and database migration management same process as production
Environment Profile
Preparing The Environment for Testing

- Development
- Test
- Staging
- Production

Code & DB Changes
Copy Code
Run Regression
Ready For Production

Development → Test → Staging → Production

Code & DB Changes
Separate Environment Benefits

- Early Warning of Bad News (Prior to Production)
- Control over all testing activities
- Tests can be repeated (regression tests) with reliable results
- Known data conditions
- No scheduling conflicts
- Not impacting other testing
Data Maturity

- Subset of test data from production
- Baselining
- Backup & Restore
- Modified test data to accommodate new releases
Build Master Data Strategy
Master Data Maintenance

Change

Data Refinement

Backup Baseline

Restore Baseline

Run V.1

Backup Baseline

Run V.2
Data Challenges

- Date and Time Requirements
- Database changes
- Data dependencies
  - Sequencing
  - Interfaces
Automating your Manual Tests

- Use the Traceability/Validation Matrix
- Proof of Concept
- Architecture Approach
- Avoiding Shelf ware
- Management Commitment
  - Right resources
  - Small wins
  - Levels 2-4 in place
How to Choose the Critical Applications

- Frequency of releases per year
- Recurring production problems
- Criticality to the business (revenue stream)
- Number of users impacted
- Visibility to customers
Final Regression Benefits

- **Documented Manual Test** are reused for the life of the product.
- **Separate test environment** assures tests are controlled and isolated.
- **Base-lining of Data** enables to receive the same test results for each run.

**Automated Test Scripts:**

- Mechanical comparison of test results
- Duplication of defects
- Increased number of test run coverage
Regression Testing Costs

- Documented Manual Test takes time to build
- Separate test environment requires additional instances for each application
- Base-lining of Data usually requires DBA support
- Automated Test Scripts can take 3 times longer to develop than manual tests
Summary

- Build Manual Tests that are repeatable
  - Traceability/Validation
- Create Separate Environment
- Baseline data
- Automate your manual test
Questions

Jeffcolorado@juno.com