Managing Large Regression Test Suites

By
Srinivas Yeluripaty PMP®
Senior Test Manager, Infosys Technologies Ltd.,

Prabhu Meruga, CSTE, CSQA
Test Manager, Infosys Technologies Ltd.,

ISQT Conference, Hyderabad, India   25/Jul/2008
Table of Contents

- Context

- Regression Testing Needs and Growth
  - Linear relationship between test suite and team size growth
  - IT Executive Sponsor’s concerns

- Strategies to Manage regression testing
  - Test Suite Management
  - Orthogonal Arrays
  - Risk Based Testing Approach
  - Automation Approaches

- Benefits & Conclusion
Context

Every IT Executive Sponsor will have major concerns on IT budgets to deal with growing regression testing needs of Information systems/IT applications. The key challenge here is the linear relationship between growing regression test requirements and effort/cost required to cover the same. It is always difficult to arrive at optimal strategy that addresses Optimized Cost, Better Quality and Improved Time to Market. This paper provides various proven strategies and methodologies to optimize the regression testing Cost while sustaining the Quality and Improving Time to Market.
Regression Testing Needs and Growth

Decide what should move into Regression

- Traceability
- Business Risk
- Technical Risk

Is regression test suite optimized?

Regression Testing

Iterative

Additional Regression Tests

Network Infrastructure changes
Operating System Patch updates
DB Migrations
Quick Fixes
Parallel Releases

Additional Regression Tests

Minor Enhancements
Defect Fixes

Release Specific Growth

Increasing Regression Test Suite Size – Month on Month, Year on Year
Linear relationship between Test Suite and Team Size Growth

- Work volume increase has a direct impact on the people required.
- The cost of QA increases from Functional Testing Scenario to Regression Testing Scenario due to regression test iterations.
- Dev-QA ratios vary between a new project and a maintenance based application.
- How to control the growing Regression Testing Cost?

Test Suite Size and Team Size – Linear Relationship

QA cost increase while moving from Functional testing to Regression Testing. Dev-QA ratios changing *

*Indicative Data Sample

*Data varies by project conditions
IT Executive Sponsor’s concerns – Which strategy to adopt?

**Key Focus Areas**
- Time to Market
- Optimize Cost
- Improve Quality

**Strategies**

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Impact</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Team Size</strong></td>
<td></td>
</tr>
<tr>
<td>Reduce</td>
<td>Low</td>
</tr>
<tr>
<td>Constant</td>
<td>High</td>
</tr>
<tr>
<td>Increase</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Coverage</strong></td>
<td></td>
</tr>
<tr>
<td>Reduce</td>
<td>Low</td>
</tr>
<tr>
<td>Complete</td>
<td>High</td>
</tr>
<tr>
<td><strong>Release</strong></td>
<td></td>
</tr>
<tr>
<td>On time</td>
<td>Low</td>
</tr>
<tr>
<td>Delay</td>
<td>Medium</td>
</tr>
<tr>
<td>On Time</td>
<td>High</td>
</tr>
<tr>
<td><strong>Time to Market</strong></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Cost</strong></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>Medium</td>
</tr>
<tr>
<td>High</td>
<td>Low</td>
</tr>
<tr>
<td><strong>Quality</strong></td>
<td></td>
</tr>
<tr>
<td>Low</td>
<td>High</td>
</tr>
<tr>
<td>Low</td>
<td>Low</td>
</tr>
</tbody>
</table>
Strategies to Manage Regression Testing

- Test Suite Management
- Optimization techniques – Addressing Cost, Time to market and quality
  - Orthogonal Arrays – Optimize test suite size
  - Risk Based Regression testing approach
  - Automation
Test Suite Management includes many critical activities...

Define

- Test Requirements Analysis
- Test Strategy documentation
- Freeze the Usage of test tools

Manage

- Traceability/Coverage
- Maintain centralized repository of test cases with backward traceability
- Upload / Document Screen Shots for results reference
- Track the test execution
- Defect Management
- Test data storage and reusability

Archive

- Backup and Data Recovery
- Reports and Graphs
Orthogonal Arrays – Optimize test suite size

- Orthogonal Array is a technique to minimize the test cases to an optimum number while maintaining an adequate level of test coverage.

- This technique is very useful when we need to test the combination and permutations of the test data conditions.

- This technique can be used for Unit, Integration, System or Acceptance testing.

**Ex:**
- A test condition with 4 input parameters each one having 3 distinct values needs 81 tests to provide complete coverage. **With orthogonal array**, the coverage can be achieved with just 9 tests achieving around 88% optimization.

**Savings and Optimization:**

<table>
<thead>
<tr>
<th>Parameters</th>
<th>Levels/Values for Parameters</th>
<th>Normal</th>
<th>Optimized</th>
<th>Savings</th>
<th>Normal</th>
<th>Optimized</th>
<th>Savings</th>
</tr>
</thead>
<tbody>
<tr>
<td>2</td>
<td>2</td>
<td>4</td>
<td>4</td>
<td>0%</td>
<td>9</td>
<td>8</td>
<td>11%</td>
</tr>
<tr>
<td>3</td>
<td>3</td>
<td>8</td>
<td>4</td>
<td>50%</td>
<td>27</td>
<td>9</td>
<td>67%</td>
</tr>
<tr>
<td>4</td>
<td>4</td>
<td>16</td>
<td>8</td>
<td>50%</td>
<td>81</td>
<td>9</td>
<td>89%</td>
</tr>
<tr>
<td>5</td>
<td>5</td>
<td>32</td>
<td>8</td>
<td>75%</td>
<td>243</td>
<td>18</td>
<td>93%</td>
</tr>
<tr>
<td>6</td>
<td>6</td>
<td>64</td>
<td>8</td>
<td>88%</td>
<td>729</td>
<td>18</td>
<td>98%</td>
</tr>
<tr>
<td>7</td>
<td>7</td>
<td>128</td>
<td>8</td>
<td>94%</td>
<td>2187</td>
<td>18</td>
<td>99%</td>
</tr>
<tr>
<td>8</td>
<td>8</td>
<td>256</td>
<td>12</td>
<td>95%</td>
<td>6561</td>
<td>18</td>
<td>100%</td>
</tr>
</tbody>
</table>

Reduce number of test cases before moving into Regression.
Risk Based Testing Approach

**Challenges**
- Need to increase QA Cycle duration
- Business criticality change with time
- Need for additional teams
- Varying needs for regression testing
- Increased test suite size

**Methodology**
- Historical Data Collection
- Test Case classification Analysis
- Stakeholder’s Review Sessions
- Top 3 Risks by each stakeholder
- Prerequisite
  - Classification of Test cases based on Criticality and Severity based on business risk, market impact.

**Results**
- Optimized Regression Test Suite based on Business Severity and Priority

**Prerequisite**
- Classification of Test cases based on Criticality and Severity based on business risk, market impact.

---

**Historical References approach**

1. **Step 1:** Identify similar releases
   - Complexity, Functionality, Modules, Timelines, Business situation
2. **Step 2:** Extract risks and experiences from that release by analyzing the QA Strategies and Results
3. **Step 3:** Identify risks mitigated, production failures along with root causes based on the release history
4. **Step 4:** Prioritize the risks
5. **Step 5:** Optimize the test suite by reprioritizing cases based on the need for regression (stability, business criticality/priority etc.,)
Automation Approaches – Optimized Regression Testing

Conventional SDLC

- Requirement Analysis
- High/Detailed Level Design & Development
- Unit and Integration Testing
- Testing (System & UAT)

Traditional Automation Process

- Test Strategy Definition
- Manual test planning & scripting
- Functional/Regression testing (Manual)
- Functional Automation Scripting & Execution

Accelerated Automation Process

- Test Strategy Definition
- Manual test planning & scripting
- Functional/Regression testing (Manual)
- Automation Testing planning
- Functional Automation Scripting & Execution

Reduced time to market

<table>
<thead>
<tr>
<th></th>
<th>QLT</th>
<th>CST</th>
<th>TTM</th>
<th>ROI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional Automation Approach</td>
<td>↑</td>
<td>↑</td>
<td>↓</td>
<td>↓</td>
</tr>
<tr>
<td>Accelerated Automation Approach</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
<td>↑</td>
</tr>
</tbody>
</table>

Win in the flat world
## Benefits and Conclusion

<table>
<thead>
<tr>
<th>Methodology</th>
<th>Description</th>
<th>Cost</th>
<th>Quality</th>
<th>Time to Market</th>
<th>ROI</th>
</tr>
</thead>
</table>
| Orthogonal Array testing Technique         | ♦ Focus is on the most important tests that will mitigate the **highest amount of risk**. This will increase test effectiveness while potentially resulting in **fewer tests**, thus **reducing cost**, while maintaining or increasing test effectiveness  
  ♦ Reduce overall cost of testing by identifying non-testable requirements and redundant test flows early in the lifecycle | ↓    | ↑       | ↑             | NA             |
| Risk based Testing / Test Case optimization | ♦ Increases test effectiveness while potentially resulting in fewer tests, thus reducing cost while maintaining or increasing test effectiveness  
  ♦ Prioritizes testing of high risk requirements, thereby resulting in testing the business critical requirements early in the testing lifecycle, which helps specifically in cases of schedule/resource crunch  
  ♦ Varying regression suites can be evolved based on the application / infrastructure releases thus focusing on specific needs of regression | ↓    | ↑       | ↑             | NA             |
| Accelerated Automation Approach             | ♦ Focus on quick and accurate execution  
  ♦ Minimize manual intervention so that effort can be better utilized  
  ♦ Increased regression test coverage  
  ♦ Increased time to Market with reduced test cycle size | ↑    | ↑       | ↑             | ↑              |

An immediately and Visual benefit
An immediate and Visual Benefit will be achieved only if a systematic implementation is followed

Initial cost will be medium to high; depending on the size of the application, significant ROI can be achieved
Thank You