Context-aware Regression Testing Techniques and Empirical Evaluations of Their Economic Impact

Hyunsook Do
North Dakota State University

Supported by the National Science Foundation

Overview of Presentation

- Background
  - regression testing
  - regression testing techniques
  - empirical studies
- Research problem
- Research projects
- Merit and impact of this research
Testing & Regression Testing

- Software Testing: validate software system

- Regression Testing: validate *modified software*  
  - Detect whether corrections and enhancements have introduced new faults  
  - Maintain the quality of subsequent releases of software

Regression Testing
Retest-all Approach

- Expensive:
  - An office application (1.8M LOC, 3128 tests):
    over 4 days to run tests
  - CISCO: $400M for five years
  - Microsoft application: 9M test cases
Regression Test Selection

Program and Modified Version

Procedure Avg
S1  count = 0
S2  fread(fptr,n)
S3  while (not EOF) do
S4   if (n<0)
S5    return(error)
S6    else          nums[count] = n
S7    endif         count++
S8    endwhile
S9    avg = mean(nums,count)
S10   return(avg)

Procedure Avg'
S1'  count = 0
S2'  fread(fptr,n)
S3'  while (not EOF) do
S4'   if (n<=0)
S5a   print("input error")
S5   return(error)
S6   else          nums[count] = n
S7   endif         count++
S8   endwhile
S9'  avg = mean(nums,count)
S10' return(avg)
Test History Information for Avg

<table>
<thead>
<tr>
<th>test</th>
<th>input</th>
<th>output</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>empty file</td>
<td>0</td>
</tr>
<tr>
<td>t2</td>
<td>-1</td>
<td>error</td>
</tr>
<tr>
<td>t3</td>
<td>1 2 3</td>
<td>2</td>
</tr>
</tbody>
</table>

CFG and Modified CFG
Identify tests that go through the changed code
⇒ depth-first traversal of two CFGs and look for code differences

Selected Tests

$T' = \{t2\}$

Dangerous Edge
Test Case Prioritization

A Simple Technique: Total Statement Coverage Prioritization

<table>
<thead>
<tr>
<th>test</th>
<th>stmts covered</th>
</tr>
</thead>
<tbody>
<tr>
<td>t1</td>
<td>s1, s2, s3, s9, s10, s11</td>
</tr>
<tr>
<td>t2</td>
<td>s1, s2, s3, s4, s5</td>
</tr>
<tr>
<td>t3</td>
<td>s1, s2, s3, s4, s6, s7</td>
</tr>
<tr>
<td></td>
<td>s8, s9, s10, s11</td>
</tr>
</tbody>
</table>

st-total: t3, t1, t2
Testing Suite Augmentation

Test Suite Reduction
Evaluation of Regression Testing Techniques

Studying the Regression Testing Process

Regression Testing $P_{V1}$

Apply Regression Testing Techniques
Studying the Regression Testing Process

Regression Testing $P_{V2}$

Apply Regression Testing Techniques

Studying the Regression Testing Process

Regression Testing $P_{V3}$

Apply Regression Testing Techniques
Studying the Regression Testing Process

Limitations for Regression Testing Research
Limitations for Regression Testing Research

- **Context factors**
  - Different types of applications:
    - safety-critical (dependability), business-critical (early market share)
  - Different testing processes:
    - big-bang, incremental, test-driven, continuous testing

Regression Testing Processes

- **Big-bang model:**
  - Safety-critical, legacy systems

- **Incremental model:**
  - Weekly, nightly testing

- **More time constraints**
Limitations for Regression Testing Research

- **Context factors**
  - Different types of applications:
    - safety-critical (dependability), business-critical (early market share)
  - Different testing processes:
    - big-bang, incremental, test-driven, continuous testing

- **Lifetime factors**
  - Cost-benefits tradeoffs considering entire system lifetime rather than individual releases

---

**Lifetime Factor**

Technique A > Technique B

Technique A < Technique B
Lifetime Factor

Cost-benefit tradeoffs considering system lifetime

* Cost-benefit tradeoffs of individual releases

Limitations for Regression Testing Research

- **Context factors**
  - Different types of applications:
    - safety-critical (dependability), business-critical (early market share)
  - Different testing processes:
    - big-bang, incremental, test-driven, continuous testing

- **Lifetime factors**
  - Cost-benefits tradeoffs considering entire system lifetime rather than individual releases

- **Cost-benefit models**
  - Considering context and lifetime factors
  - Considering common metric across other techniques
Lack of Common Metric for Techniques

- Test Case Selection
  - # of tests selected
  - Test execution time savings

- Test Case Prioritization
  - Rate of fault detection

- Cost-benefit model

Convert measures into common metric and compare them

Research Projects

- Create novel *regression testing techniques*
- Create novel *adaptive regression testing strategies*
- Create *comprehensive economic models*
- Evaluate and refine techniques, strategies and models through the *rigorous empirical approach*
Research Projects

• Create novel regression testing techniques

• Create novel adaptive regression testing strategies

• Create comprehensive economic models

• Evaluate and refine techniques, strategies and models through the rigorous empirical approach

Regression Testing Techniques that Address Context Factors

• Regression Testing Techniques for Large-scale Industrial Applications
  – Require a long period time to test
  – Practice mixed testing processes
  – E.g., Microsoft Dynamics Ax

• Regression Testing Approach for Frequent Patches
  – Require a short turn-around time
  – E.g., web applications
Regression Testing for Large-scale Industrial Applications

- Various phases of software development => produce numerous software artifacts
- Understanding the underlying relationships in massive data => help improve regression testing techniques and strategies

Analyzing massive data can be challenging!

Regression Testing for Large-scale Industrial Applications

- Data mining can address challenges that traditional data analysis techniques have faced  
  => e.g., massive, complex, heterogeneous data
Clustering approach to improving test case prioritization

- Test cases have common properties
  ⇒ Test cases within the same group may have similar fault detection ability
  ⇒ Engineers can manage regression testing activities more effectively using cluster-based prioritization

So, we investigated whether a clustering approach can help improve the effectiveness of test case prioritization

Prioritization with Clustering

1. Cluster test cases using their code coverage
2. Prioritize test cases within each cluster (coverage, complexity, fault history)
3. Generate a complete set of reordered tests
Empirical Study

- Research question:
  Does clustering improve prioritization?

- Object of analysis
  Microsoft Dynamics AX: 3 versions (650-707 KLOCs), tests (500-908), real faults (139-254)

Variables

- Independent Variable: prioritization technique

<table>
<thead>
<tr>
<th>Control</th>
<th>Heuristics (use clustering)</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tcov</td>
<td>Tcov-clst</td>
<td>Code coverage</td>
</tr>
<tr>
<td>Tcc</td>
<td>Tcc-clst</td>
<td>Code complexity</td>
</tr>
<tr>
<td>Tfb</td>
<td>Tfb-clst</td>
<td>Fault-based</td>
</tr>
<tr>
<td>Tcb</td>
<td>Tcb-clst</td>
<td>Combination of Tcc and Tfb</td>
</tr>
</tbody>
</table>

- Dependent Variable: fault detection rate
<table>
<thead>
<tr>
<th>Control (ctrl)</th>
<th>Fault D. R.</th>
<th>Heuristics (hrst)</th>
<th>Fault D. R.</th>
<th>Improvement (hrst/ctrl)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Version 1</td>
<td></td>
<td>Version 2</td>
<td></td>
</tr>
<tr>
<td>Tcov</td>
<td>50.78</td>
<td>Tcov-clst 73.82</td>
<td>66.97</td>
<td>44%</td>
</tr>
<tr>
<td>Tcc</td>
<td>48.59</td>
<td>Tcc-clst 73.82</td>
<td>67.09</td>
<td>47%</td>
</tr>
<tr>
<td>Tfb</td>
<td>50.37</td>
<td>Tfb-clst 64.67</td>
<td>56.01</td>
<td>17%</td>
</tr>
<tr>
<td>Tcb</td>
<td>48.25</td>
<td>Tcb-clst 71.86</td>
<td>67.31</td>
<td>46%</td>
</tr>
</tbody>
</table>

Discussion

- Clustering can help improve prioritization
  
  *Tests have common properties => similar fault detection ability*

- Practical implications for software industry
  
  *find defects early during daily (weekly) regression testing => early feedback to developers => fewer faults during the entire regression testing => quality & schedule control*
Regression Testing Approach for Frequent Patches

• Some types of systems undergo more frequent patches (e.g., web applications)
  – Security breaches
  – Features/functionality updates

• Often small patches or revisions

• Regression testing an entire system can require a lot of time and effort
  – A short turn-around time in releasing patches is critical because the applications have already been used.

• How to efficiently perform regression testing
Regression Testing Approach for Frequent Patches

- Focus only on the areas of code have been changed and regression test them
  ⇒ Companies can deliver patches more quickly and dependably

1) Identify the areas of the application impacted by the changes
2) Generate new test cases for the impacted areas of code using program slicing

PHP Analysis and Regression Testing Engine (PARTE)
Empirical Study

- PHP Analysis and Regression Testing Engine (PARTE)
- RQ: Can our approach efficiently generate test cases?

<table>
<thead>
<tr>
<th>Application</th>
<th>Version</th>
<th>Lines of Code</th>
<th>No. of Files</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAQForge</td>
<td>1.3.0</td>
<td>1806</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>1.3.1</td>
<td>1837</td>
<td>20</td>
</tr>
<tr>
<td></td>
<td>1.3.2</td>
<td>1671</td>
<td>18</td>
</tr>
<tr>
<td>osCommerce</td>
<td>2.2MS1</td>
<td>53510</td>
<td>302</td>
</tr>
<tr>
<td></td>
<td>2.2MS2</td>
<td>68330</td>
<td>506</td>
</tr>
<tr>
<td></td>
<td>MS2-60817</td>
<td>78892</td>
<td>502</td>
</tr>
</tbody>
</table>

Empirical Results

<table>
<thead>
<tr>
<th>Application</th>
<th>Version Pair</th>
<th>Total Paths of Entire Program</th>
<th>PARTE Paths</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAQForge</td>
<td>1.3.0 &amp; 1.3.1</td>
<td>612</td>
<td>4% 20</td>
</tr>
<tr>
<td></td>
<td>1.3.1 &amp; 1.3.2</td>
<td>1418</td>
<td>24% 335</td>
</tr>
<tr>
<td>osCommerce</td>
<td>2.2MS1 &amp; 2.2MS2</td>
<td>34924</td>
<td>5% 1630</td>
</tr>
<tr>
<td></td>
<td>2.2MS2 &amp; 2.2MS2-060817</td>
<td>35775</td>
<td>6% 2464</td>
</tr>
</tbody>
</table>
Summary

• Proposed an efficient regression testing technique for small changes in PHP web applications
  – Identifies the areas impacted by changes
  – Creates test paths using slices

• Main benefits
  – Deliver patches quickly and dependably
  – Can be beneficial when the major releases are tested (artifacts are accumulated over time)

Research Projects

• Create novel regression testing techniques

• Create novel adaptive regression testing strategies

• Create comprehensive economic models

• Evaluate and refine techniques, strategies and models through the rigorous empirical approach
Research Projects

• Create novel regression testing techniques

• Create novel adaptive regression testing strategies

Types of maintenance activities change across system lifetime
⇒ Differences in versions can involve different amounts and types of code modification
⇒ This variance can affect the cost and benefits of regression testing techniques

• Create comprehensive economic models

• Evaluate and refine techniques, strategies and models through the rigorous empirical approach
Adaptive Regression Testing (ART) Strategies

• No single regression testing technique is best to use across all versions

• Try to identify the most cost effective technique for each regression testing session

Adaptive Regression Testing (ART) Strategies

• ART strategies
  – evaluate regression testing techniques in terms of decision criteria (e.g., cost and benefit factors)
  – choose the best alternative (organization’s situations and feedback from prior testing sessions)

• Multiple Criteria Decision Making (MCDM) Problem (e.g., WSM, WPM, AHP, …)
Empirical Study

- Research Question

Is AHP effective for selecting appropriate test case prioritization techniques across system lifetime?
Objects of Study

<table>
<thead>
<tr>
<th>Objects</th>
<th>Versions</th>
<th>Classes</th>
<th>Size (Klocs)</th>
<th>Test Cases</th>
<th>Mutation Faults</th>
</tr>
</thead>
<tbody>
<tr>
<td>ant</td>
<td>9</td>
<td>914</td>
<td>61.7</td>
<td>877</td>
<td>412</td>
</tr>
<tr>
<td>xml-security</td>
<td>4</td>
<td>145</td>
<td>15.9</td>
<td>83</td>
<td>246</td>
</tr>
<tr>
<td>jmeter</td>
<td>6</td>
<td>434</td>
<td>42.2</td>
<td>78</td>
<td>386</td>
</tr>
<tr>
<td>galileo</td>
<td>16</td>
<td>68</td>
<td>14.5</td>
<td>912</td>
<td>2494</td>
</tr>
<tr>
<td>nanoxml</td>
<td>6</td>
<td>64</td>
<td>3.1</td>
<td>216</td>
<td>204</td>
</tr>
</tbody>
</table>

Variables

- **Independent variable**: test case prioritization technique application mapping strategy (Orig, Rand, Tcov, and Acov; assigns techs to versions)

- **Dependent variable**: relative cost-benefit value (relative to original order) obtained from EVOMO economic model
Five Mapping Strategies

Variables

- **Independent variable**: test case prioritization technique application mapping strategy (Orig, Rand, Tcov, and Acov)

- **Dependent variable**: relative cost-benefit value (relative to original order) obtained from EVOMO economic model
Economic Models for Regression Testing and Strategies

• EVOlution-aware economic MOdel for regression testing (EVOMO)
  
  – Consider various cost and benefit factors (e.g., setup, obsolete test, analysis, applying techniques …)
  
  – Consider system lifetime

EVOMO Economic Model
(EVOlution-aware economic MOdel for regression testing)

- Missed faults
- Test setup
- Identifying/repairing obsolete test cases
- Test result validation
- Supporting analysis
- Regression testing technique execution
- Test execution
- Delayed fault detection feedback

programmers’ salary

product revenue

common metric ($)

[Diagram showing various factors affecting economic models]
Practical Implications of the Results

Techniques selected by AHP can be more effective

- From prior empirical studies
  - Prioritization heuristics are more cost-effective than control
  - Other factors also affect the choice of techniques
    - Program size, test suite size, organization’s testing environment, etc.

Adopting different prioritization techniques by considering factors is a potential practical approach for organizations
Research Projects

• Create novel regression testing techniques
• Create novel adaptive regression testing strategies
• Create comprehensive economic models
• Evaluate and refine techniques, strategies and models through the rigorous empirical approach

Merit and Impact of This Research

• Provide new regression testing techniques and strategies that consider context factors
• Empirical data about those techniques and strategies
• Provide sound cost-benefit models and empirical approaches

Lay a foundation for evaluating cost-effectiveness of various regression testing techniques and strategies in practical ways
**Ongoing Work**

- Develop new regression testing techniques and approaches
  - Association rule-mining
  - Other MCDM approaches (e.g., WSM, WPM)
- Continued empirical assessment
- Refining and enhancing cost-benefit models
  - Empirical assessment
  - Sensitivity analysis