Manage Risk by Risk-Driven Continual Regression Testing

Yanping Chen
School of Information Technology and Engineering, University of Ottawa
Outline

- Risk and risk-based testing
- Regression testing and risk-based continual regression testing
- Risk-based regression test case selection
- Risk-based end-to-end scenario selection
- Real experience to date
- Summary and recommendations
- Reference
Risk and Risk-based Testing

- Risk: event that has some probability of happening, and that if it occurs, will result in some loss
- Risk-based testing: do heavier testing of those parts that may bring higher risk
- Risk-based testing actions
  - Identify risk for functions or features
  - Quantify risk and create ranked list of functions or features
  - Design test cases based on ranked list
Why Risk-based Testing?

- All testing is motivated by risk: Tester’s job is finding high-priority problems to avoid risk
- Traditional testers have always used risk-based testing, but in ad hoc fashion based on their personal judgment [4]
- Using risk to measure quality of test suite is reasonable

“Risk-based testing” vs. “Food-based living” [Air]
Questions to asked for risk-based approach

- Which areas are significant?
- How much testing is enough for average area?
- What are risks involved in leaving certain bug unresolved?
- At what point can product be considered adequately tested and ready for market?
Continual Regression Testing

- *To ensure that new or modified features do not cause current release to regress after incorporating fixes into product -- ensure customer’s business won’t be at risk*
- Essential to ensure software quality
- In software maintenance: validate modified software
- In O-O software development
  - Ensure quality of successive increments
  - Assess quality of re-used components
- Continual regression testing: execute regression tests every day or on every new build
Typical Regression Test Selection

Test Suite $T$

$T''$: New test cases

$T'$: Regression test cases selected from $T$

Regression Test Suite $T'''$

Program $P$

New Features

Changed Features

Program $P'$
A Simple Risk Model [2]

- Two elements of Risk Exposure ($RE_f$):
  - Probability of fault
  - **Cost** (consequence or impact) of fault in corresponding function if it occurs in production

- $RE_f = P_f \times C_f$
  - $RE_f$: Risk Exposure of function $f$
  - $P_f$: probability of fault occurring in function $f$
    - In our model, we consider severity of defects to assess probability
      - **Note:** $P(f)$ is extended to severity probability
    - $C_f$: cost if fault occurs (in production) in function $f$
Risk-based Regression Testing Approach

Model-based Tests Selection Method:
Step 1. Assess cost $C_t$ for each test case
Step 2. Derive severity probability $P_t$ for each test case
Step 3. Calculate Risk Exposure $RE_t$ for each test case
Step 4. Select test cases with top $RE_t$ as regression test cases
Assess Cost $C_t$

- **Two kinds of costs**
  - $C_t (c)$: Consequences of fault as seen by customer, i.e., losing market place
  - $C_t (v)$: Consequences of fault as seen by vendor, i.e., high software maintenance cost

- $C_t$ is categorized on 1~5 scale (1- low, 5 - high)
  - Weight $C_t (c)$ and $C_t (v)$ equally
  - $C_t = (C_t (c) + C_t (v))/2$
Assess Cost $C_t$ (Cont’d)

- $C_t (c)$
  - Test case takes one, specific control flow and includes some data
  - Create questionnaire with questions for both control flow and data
  - Score each test case based on answers for questionnaire as $C_t (c)$, on 1~5 scale (1- low, 5 - high)

- $C_t (v)$
  - Cost to fix bugs is dependent on system complexity
  - Use proper questionnaire in assessment
  - Measure $C_t (v)$ on 1~5 scale (1- low, 5 - high)
Derive Severity Probability $P_t$

- Summarize number of defects opened for each test case after running full test suite
- Calculate average severity of defects for each test case
- *Use result of Number of Defects ($N_t$) times Average Severity ($S_t$) $N_t \times S_t$ to assess severity probability*
- $P_t$ falls into 1~5 range (1 - low, 5 - high)
  - Test cases without any defects in full testing, $P_t = 1$.
  - Test cases with the top 25% estimate $N_t \times S_t$, $P_t = 5$
  - Test cases with the bottom 25% estimate $N_t \times S_t$, $P_t = 2$
Calculate *Risk Exposure* $RE_t$

**Step 1:**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>$C_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0010</td>
<td>5</td>
</tr>
<tr>
<td>t0020</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>$t_n$</td>
<td>3</td>
</tr>
</tbody>
</table>

**Step 2:**

<table>
<thead>
<tr>
<th>Test Case</th>
<th>$N_t$</th>
<th>$S_t$</th>
<th>$P_t$</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0010</td>
<td>3</td>
<td>2</td>
<td>5</td>
</tr>
<tr>
<td>t0020</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>...</td>
<td>.....</td>
<td>.....</td>
<td>.....</td>
</tr>
<tr>
<td>$t_n$</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

$$RE_t = P_t \times C_t$$
Select Test Cases with Top $RE_t$

- Choose test cases with highest value of $RE_t$
- Reach pre-defined coverage target (e.g., 30% of full test suite)

<table>
<thead>
<tr>
<th>Test Case</th>
<th>Full Test Suite</th>
<th>Regression Test Suite (30%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0010</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t0020</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t0030</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t0040</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t0050</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>t0060</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>t0070</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>
**Risk-based End-to-end Regression Test Scenario Selection**

- **Test Scenario**
  - Simulate common user profiles of system use
  - More customer-directed
  - Highly effective at finding regression faults
  - Covers sequence of test cases -- *Traceability*

- **Selection rules**
  - Select scenarios that contain most critical test cases
  - Have test suite of scenarios cover as many test cases as possible
Risk-based Regression Test
Scenario Selection

End-to-end Test Scenario Selection Method

To start: Create traceability matrix between scenarios and test cases

Step 1. Calculate Risk Exposure $RE_s$ for each scenario

Step 2. Select scenario with highest $RE_s$ as regression tests

Step 3. Update traceability matrix and re-calculate $RE_s$

Step 4. Repeat Steps 2 and 3 until out of time and resources
End-to-end Test Scenario
Selection Method with Example

Step 1. Calculate Risk Exposure
$RE_s$ for each scenario

$RE_s = \sum RE_{t_i}, \{1 \leq i \leq n | \text{test case } t_i \text{ is covered by this scenario}\}$

Step 2. Select scenario with highest $RE_s$ for regression testing
<table>
<thead>
<tr>
<th></th>
<th>s001</th>
<th>s002</th>
<th>s003</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0010</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0020</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0030</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0040</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>t0050</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0060</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0070</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Scenario**

<table>
<thead>
<tr>
<th>Scenario</th>
<th>RE&lt;sub&gt;s&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>s001</td>
<td>985</td>
</tr>
<tr>
<td>s002</td>
<td>463</td>
</tr>
<tr>
<td>s003</td>
<td>732</td>
</tr>
<tr>
<td>s004</td>
<td>213</td>
</tr>
<tr>
<td>s005</td>
<td>195</td>
</tr>
<tr>
<td>s006</td>
<td>127</td>
</tr>
<tr>
<td>s007</td>
<td>70</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

**Table (excerpted):**

<table>
<thead>
<tr>
<th></th>
<th>C&lt;sub&gt;f&lt;/sub&gt;</th>
<th>P&lt;sub&gt;f&lt;/sub&gt;</th>
<th>RE&lt;sub&gt;f&lt;/sub&gt; = P&lt;sub&gt;f&lt;/sub&gt; × C&lt;sub&gt;f&lt;/sub&gt;</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0010</td>
<td>5</td>
<td>5</td>
<td>25</td>
</tr>
<tr>
<td>t0020</td>
<td>2</td>
<td>4</td>
<td>8</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>t&lt;sub&gt;n&lt;/sub&gt;</td>
<td>3</td>
<td>1</td>
<td>3</td>
</tr>
</tbody>
</table>
Step 3. Update traceability matrix and re-calculate $RE_s$

- When running chosen scenario, some test cases will be covered – not necessary to cover again
- Thus, after chosen scenario has been executed
  - Delete column for chosen scenario
  - Delete rows for all test cases that have been covered by this scenario
- Based on updated relation table, re-calculate $RE_s$ for rest scenarios and re-build Risk Exposure table

Step 4. Repeat Steps 2 and 3 until out of time and resources

- Size of test suite is dependent on time and resources
<table>
<thead>
<tr>
<th>t0010</th>
<th>s001</th>
<th>s002</th>
<th>s003</th>
<th>...</th>
</tr>
</thead>
<tbody>
<tr>
<td>t0020</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0030</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0040</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0050</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0060</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>...</td>
</tr>
<tr>
<td>t0070</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>...</td>
</tr>
</tbody>
</table>

Next choice -- s003

<table>
<thead>
<tr>
<th>Scenario</th>
<th>REs</th>
</tr>
</thead>
<tbody>
<tr>
<td>s002</td>
<td>356</td>
</tr>
<tr>
<td>s003</td>
<td>611</td>
</tr>
<tr>
<td>s004</td>
<td>176</td>
</tr>
<tr>
<td>s005</td>
<td>180</td>
</tr>
<tr>
<td>s006</td>
<td>96</td>
</tr>
<tr>
<td>s007</td>
<td>68</td>
</tr>
</tbody>
</table>

...
Case Study with historical data of IBM WebSphere

- Three components of IBM WebSphere with different characters
  - Component One: Focus on functionality
  - Component Two: Focus on data
  - Component Three: Both functionality and data are important
- Each component was owned by one experienced tester
- 306 test cases in total
Real Experiences to Date

- High Risk Exposure coverage and average Risk Exposure
- Acceptable specification coverage not our focus
- Only requires straightforward calculation – *can be automated*
- Systematic – *not subjective*!
- Powerful in selecting effective test cases and finding defects
  - Caught all defects
  - Omitted fewer test cases that failed in execution

<table>
<thead>
<tr>
<th></th>
<th>Risk-based Test Suite</th>
<th>Manual Test Suite</th>
<th>Compared Well</th>
</tr>
</thead>
<tbody>
<tr>
<td>Defects Detected (%)</td>
<td>100%</td>
<td>84.1%</td>
<td>√√</td>
</tr>
<tr>
<td>Defect-revealing Test Cases Selected (%)</td>
<td>93.9%</td>
<td>83.1%</td>
<td>√</td>
</tr>
</tbody>
</table>
Summary

- New risk-based regression test technique
  1. *Risk-based* regression test case selection
  2. *Risk-based* regression test scenario selection
- New objective selection criteria that has good potential to guide regression test selection, even for new or less-experienced test personnel – SYSTEMATIC APPROACH!
- An EFFECTIVE means of QUANTIFYING quality of test suite
Recommendations for Adoption in Process

- Highlight & motivate **RISK**
  - Analysis
  - Planning
  - Results

- Collect risk data
  - Test plan
    - Cost of test cases
    - Scenarios vs. test cases
  - Test profile
    - Number of defects by test case
    - Defect severity

- Measure efficiency & effectiveness
  - % defect detection
  - % defect-revealing test case coverage
Reference


