Regression Testing Practices

Per Runeson and Emelie Engström
Software Engineering Research Group
LUND UNIVERSITY
Sweden

Regression Testing

“Retesting...to ensure that the new version of the software has retained the capability of the old version”

[Burnstein02]
Multi-dimensional regression test

Coverage Items
- Expectations, Quality requirements
- Functional requirements, Design
- Interfaces, Implementation

Test cases
- Acceptance test
- System test
- Integration test
- Unit test

Versions (evolution in time)

Levels (views of the system)

Variants (variation in space)

Repetitive tests
- across versions
- across variants
- across levels

Redundancy?

Qualitative survey of regression testing practices

- Focus group discussions
  - 15 participants
  - 10 companies

- Questionnaire
  - 32 respondents
  - 29 companies
Goals and questions

RQ1 - What is meant by regression testing in industry?
RQ2 - Which problems or challenges related to regression testing exist?
RQ3 - Which good practices on regression testing exist?

Analysis of the results

What?

- Repetitive testing after changes
  - new configurations
  - fixes
  - changed solutions
  - new hardware...

- Find defects or obtain a measure of quality

- The amount and frequency determined by
  - the assessed risk
  - the size and type of the change
  - the amount of available resources
Analysis of the results

**When?**

- **At different levels**
  - system, integration, unit

- **At different stages**
  - As early as possible, as late as possible, continuously

**How?**

- **Systematic / ad hoc**
  - complete retest
  - prioritisation and selection
  - static and dynamic test suites
  - change analysis

- **Manual / Automated**
Analysis of the results

Challenges!

- Test case selection
  - Change impact assessment
  - Coverage assessment
  - Traceability requirement/test
- Automated vs. manual testing
  - Cost benefit
  - Environment
- Design for testability
  - Dependencies in software
  - Test selection/scoping

Good practices!

- Focus automation below the user interface level
- Regression test continuously
- Vary the test focus between different test rounds
- Visualise progress monitoring
- Connect software quality attributes to each test case
**Transparent and Efficient Regression Test Selection - TERTS**

**Identify**
- Context
- Factors
- Tools

**Implement**
- Procedure
- Tools

**Evaluate**
- Transparency
- Efficiency

---

**History-based regression testing**

\[ PR_k = \alpha \frac{f_{ck}}{e_{ck}} + \beta \cdot PR_{k-1} + \gamma \cdot h_k \]

\[ 0 \leq \alpha, \beta, \gamma \leq 1, k \geq 1 \]

- Historical effectiveness \((f_{ck} / e_{ck})\)
- Previous priority \((PR_{k-1})\)
- Execution history \((h_k)\)
Evaluation at Sony Ericsson

Fix-cache regression testing procedure

1. Identify fault-prone files
2. Link files to test cases
3. Recommend test cases based on cache content
Evaluation at ST-Ericsson

![Graph showing efficiency (fault/test case) over iterations]

<table>
<thead>
<tr>
<th>Iteration</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
</tr>
</thead>
<tbody>
<tr>
<td>RTC</td>
<td>27</td>
<td>41</td>
<td>99</td>
<td>1</td>
<td>11</td>
<td>78</td>
</tr>
<tr>
<td>XRTC</td>
<td>13</td>
<td>12</td>
<td>71</td>
<td>1</td>
<td>5</td>
<td>47</td>
</tr>
<tr>
<td>TXTC</td>
<td>552</td>
<td>480</td>
<td>1301</td>
<td>906</td>
<td>1203</td>
<td>1317</td>
</tr>
</tbody>
</table>

RTC = Recommended Test Cases, XRTC = Executed Recommended Test Cases, TXTC = Total Executed Test Cases

Conclusions from survey

- Definitions are the same – practices differ (RQ1)
- Challenges and good practices (RQ2, RQ3) relate to
  - Test selection,
  - General testing issues
  - Management practices
  - Design issues
Conclusions from TERTS

- Automating RTS requires good information
- Potential for improving efficiency – but studies are limited
- Transparency is a goal in itself

Ongoing research

- Test Scope Selection
- History Based Testing
- Alignment