An Empirical Study of Software Reuse vs. Defect-Density and Stability

Ivar Ursin Nikolaisen

October 30, 2012
Authors

- Parastoo Mohagheghi
  - Ericsson, NTNU, Simula

- Reidar Conradi
  - NTNU, Simula

- Ole M. Killi
  - NTNU

- Henrik Schwarz
  - NTNU

Published in 2004 in the “Proceedings of the 26th International Conference on Software Engineering IEEE” (ICSE’04).
Definition

- **Defect**
  - Faults, errors or failures. The software is not acting as it should.

- **Defect-density**
  - Number of defects divided by lines of code.

- **Stability**
  - Degree of modification. If a component goes through little to no modification between software releases it is termed “stable”.
The paper describes the results of an empirical study. A large-scale telecom system by Ericsson is studied. The study concerns Component-Based Software Engineering (CBSE).

The overall research question is the impact of reuse on software quality. Specifically:

- The impact of software reuse on defect-density and stability.
- The impact of component size on defects and defect-density.

Limitations:

1. Are defect density and stability good indicators of software quality?
2. Can we generalize the results?
■ Seeks to contribute because of “a lack of published, empirical studies on large industrial systems.”
■ While relevant data is often gathered, such data is often not analyzed much, or if it is then the results are not published.
A large-scale distributed telecom system developed by Ericsson.

Application specific components are non-reused components.

The reused components are reused both between applications A and B, and between successive releases of these applications.
- The study analyzes data from the defect reporting system for 12 product releases.
- Analysis of one release in particular is presented in detail in the paper.
- Data is extracted from a Trouble Report (TR) database.

- Face issues of incomplete TRs, for example inconsistent component naming, lacking information about which component the defect was in.
Hypothesis 1: Reuse and defect-density

- Studies the relation between component type (reused vs. non-reused) and defect-density.
- Reused blocks have lower defect-density than non-reused ones (with p-value less than 0.1). Almost 50% less.

<table>
<thead>
<tr>
<th>HypId</th>
<th>Hypothesis Text</th>
<th>Result</th>
</tr>
</thead>
<tbody>
<tr>
<td>H1</td>
<td><strong>H01:</strong> Reused components have the same defect-density as non-reused ones.</td>
<td>Rejected</td>
</tr>
<tr>
<td></td>
<td><strong>HA1:</strong> Reused components have lower defect-density than non-reused ones.</td>
<td>Accepted</td>
</tr>
</tbody>
</table>
Hypothesis 2: Reuse vs. component size and defects

- Non-reused components appear to have more defects than reused ones, even though the reused components are larger.
- Could perhaps be explained by different programming languages (predominantly Erlang in reused, C in non-reused components).

<table>
<thead>
<tr>
<th>H2</th>
<th>H02-1: There is no relation between number of defects and component size for all components.</th>
<th>Not rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>H02-2: There is no relation between number of defects and component size for reused components.</td>
<td>Not rejected</td>
</tr>
<tr>
<td></td>
<td>H02-3: There is no relation between number of defects and component size for non-reused components.</td>
<td>Rejected</td>
</tr>
</tbody>
</table>
Hypothesis 3: Reuse vs. component size and defect-density

- No relation between defect-density and component size is found.

<table>
<thead>
<tr>
<th></th>
<th>H03-1: There is no relation between defect-density and component size for all components.</th>
<th>Not rejected</th>
</tr>
</thead>
<tbody>
<tr>
<td>H3</td>
<td>H03-2: There is no relation between defect-density and component size for reused components.</td>
<td>Not rejected</td>
</tr>
<tr>
<td></td>
<td>H03-3: There is no relation between defect-density and component size for non-reused components.</td>
<td>Not rejected</td>
</tr>
</tbody>
</table>
Hypothesis 4: Reuse and stability

- Reused components were more stable (changed less between releases) than non-reused ones.
- Internal survey of 9 developers found that developers had similar opinions.

| H4 | H04: Reused and non-reused components are equally modified. | Rejected |
|    | HA4: Reused components are modified more than non-reused ones. | Rejected |
Validity threats

- Conclusion validity
  - Reused and non-reused components could have different functionality and constraints. For example non-reused components may have user interfaces.
  - Different developers with different experience levels. Not considered a problem for this study since all components are developed by the same development unit.
  - TR database does not contain all defects found (inspections, unit testing).

- Internal validity
  - Missing data. There could be a systematic problem where some classes of TRs have incomplete data. It is not believed that the missing data presents a systematic problem in this study.
Validity threats

■ Construct validity

☐ Defect-density and stability are commonly used as indicators for software quality.

■ External validity

☐ All data is taken from the same project.
☐ Generalization is possible for other releases of the same project, or even other projects in the same company.
☐ Should have more studies from other projects before broader generalizations can be made.
- The study finds that reused components have lower defect-density than non-reused ones.
- Reused components had a larger amount of high-severity defects before delivery, but less after delivery.
  - This could mean that the reused components were given a higher priority for fixes.
- Defects generally increase with component size, but not for reused components.
- Reused components were more “stable”, as in fewer modifications being made, between successive releases.
- No particularly surprising results.