ISSUE REPORT ANALYSIS

Simon Curty
Bachelor Thesis / Seminar Project
Issue Report Classification
(Bachelor thesis)

Bug Report Quality Assessment
(Seminar Project)
Misclassification of Issue Reports

It’s Not a Bug, It’s a Feature:
How Misclassification Impacts Bug Prediction

Kim Herzig
Saarland University
Saarbrücken, Germany
herzig@cs.uni-saarland.de

Sascha Just
Saarland University
Saarbrücken, Germany
just@st.cs.uni-saarland.de

Andreas Zeller
Saarland University
Saarbrücken, Germany
zeller@cs.uni-saarland.de

Abstract—In a manual examination of more than 7,000 issue reports from the bug databases of five open-source projects, we found 33.8% of all bug reports to be misclassified—that is, rather than referring to a code fix, they resulted in a new feature, an update to documentation, or an internal refactoring. This misclassification introduces bias in bug prediction models, confusing bugs and features. On average, 39% of files marked as defective actually never had a bug. We discuss the impact of this misclassification on earlier studies and recommend manual data validation for future studies.

Index Terms—Mining software repositories, bug reports, data

<table>
<thead>
<tr>
<th>TABLE I</th>
<th>PROJECT DETAILS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Maintainer</td>
<td>Tracker type</td>
</tr>
<tr>
<td>HTTPClient</td>
<td>APACHE</td>
</tr>
<tr>
<td>Jackrabbit</td>
<td>APACHE</td>
</tr>
<tr>
<td>Lucene-Java</td>
<td>APACHE</td>
</tr>
<tr>
<td>Rhino</td>
<td>MOZILLA</td>
</tr>
<tr>
<td>Tomcat5</td>
<td>APACHE</td>
</tr>
</tbody>
</table>
The commons release process (http://jakarta.apache.org/commons/releases.html) is a good starting place, but is out of date. When we do our own releases, there are some other steps particular to maven, the test-local and the webapp tests that must be documented.

Still need to link to it from the nav bar.
It’s Not a Bug, It’s a Feature: How Misclassification Impacts Bug Prediction

Kim Herzig
Saarland University
Saarbrücken, Germany

Sascha Just
Saarland University
Saarbrücken, Germany

Andreas Zeller
Saarland University
Saarbrücken, Germany
Goal

• Issue Report Classifier: A program that tells you, if an issue report describes a bug or not

• Better precision than reporters (over 66%)
**Type**: Issue category on tracker

**Label**: Issue category according to paper
Issue Report Classifier

Issue Report → Vectorization → FNN → Output
The combination of SSL tunnelling, host authentication, and disabled persistent connection support (HTTPD KeepAlive off) causes an infinite loop in HttpMethodDirector.

The problem has been reported on the httpclient-dev list by Rindress MacDonald <RMacDona at enterasys.com>
**Details**

<table>
<thead>
<tr>
<th>Type</th>
<th>Status</th>
<th>Resolution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Bug</td>
<td>Closed</td>
<td>Fixed</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Priority</th>
<th>Affects Version/s</th>
<th>Component/s</th>
<th>Labels</th>
<th>Environment</th>
<th>Bugzilla Id</th>
</tr>
</thead>
</table>
| Major    | 2.0 Alpha 1      | HttpClient (classic) | None   | Operating System: All  
|          |                  |             |        | Platform: All   | 16421       |

**Description**

The commons release process ([http://jakarta.apache.org/commons/releases.html](http://jakarta.apache.org/commons/releases.html)) is a good starting place, but is out of date. When we do our own releases, there are some other steps particular to maven, the test-local and the webapp tests that must be documented.

**Activity**

*Jeff Dever* added a comment - 26/Jan/03 15:22

The first pass is in there, I'll refine it over time.  

Still need to link to it from the nav bar.
Vector Component Examples

• Stack trace
• Test cases
• Code samples
• Code reference (reference to a specific place in the code)
• Priority/Severity
• Keywords

• 21 in total
Vectorization

Issue Report

Stack trace heuristic

0

Code sample heuristic

1

... 

Test case heuristic

0

Heuristic

(0, 1,..., 0)
FNN

- Input layer
- Hidden layers (25 hidden nodes)
- Output layer
Experiment Setup

• 100 independent runs
• Dataset is split into 75% for training and 25% for testing (random sampling)
• Balancing of dataset
• Early stopping (max. 1000 epochs)
• Fixed learning rate of 0.1
### Results

<table>
<thead>
<tr>
<th></th>
<th>Bug</th>
</tr>
</thead>
<tbody>
<tr>
<td>Precision</td>
<td>0.9</td>
</tr>
<tr>
<td>Recall</td>
<td>0.72</td>
</tr>
<tr>
<td>F1 Score</td>
<td>0.8</td>
</tr>
</tbody>
</table>

Precision, recall and F1 Score of the category **Bug** with binary classification over a dataset with **matching label and type**
Problems and Limitations

• Vectorization is bug centric

• Only for Java

• Some elements hard to detect
Bug Report Quality

What Makes a Good Bug Report?

Thomas Zimmermann, Member, IEEE, Rahul Premraj, Nicolas Bettenburg, Member, IEEE, Sascha Just, Member, IEEE, Adrian Schrötter, Member, IEEE, and Cathrin Weiss

Abstract—In software development, bug reports provide crucial information to developers. However, these reports widely differ in their quality. We conducted a survey among developers and users of APACHE, ECLIPSE, and MOZILLA to find out what makes a good bug report. The analysis of the 466 responses revealed an information mismatch between what developers need and what users supply. Most developers consider steps to reproduce, stack traces, and test cases as helpful, which are, at the same time, most difficult to provide for users. Such insight is helpful for designing new bug tracking tools that guide users at collecting and providing more helpful information. Our CUEZILLA prototype is such a tool and measures the quality of new bug reports; it also recommends which elements should be added to improve the quality. We trained CUEZILLA on a sample of 289 bug reports, rated by developers as part of the survey. The participants of our survey also provided 175 comments on hurdles in reporting and resolving bugs. Based on these comments, we discuss several recommendations for better bug tracking systems, which should focus on engaging bug reporters, better tool support, and improved handling of bug duplicates.

Index Terms—Testing and debugging, distribution, maintenance, and enhancement, human factors, management, measurement.
This bug has been reported on the HttpClient user list by Ilya Kharmatsky <ilyak at mainsoft.com>
Mismatch between what developers deemed important and what was provided by reporters.

<table>
<thead>
<tr>
<th>Developers</th>
<th>Reporters</th>
</tr>
</thead>
<tbody>
<tr>
<td>steps to reproduce (83%)</td>
<td>steps to reproduce (98%)</td>
</tr>
<tr>
<td>stack traces (57%)</td>
<td>observed behavior (96%)</td>
</tr>
<tr>
<td>test cases (51%)</td>
<td>expected behavior (94%)</td>
</tr>
<tr>
<td>observed behavior (33%)</td>
<td>product (94%)</td>
</tr>
<tr>
<td>screenshots (26%)</td>
<td>version (91%)</td>
</tr>
<tr>
<td>expected behavior (22%)</td>
<td>operating system (90%)</td>
</tr>
<tr>
<td>code examples (14%)</td>
<td>summary (90%)</td>
</tr>
<tr>
<td>summary (13%)</td>
<td>component (87%)</td>
</tr>
<tr>
<td>version (12%)</td>
<td>severity (77%)</td>
</tr>
<tr>
<td>error reports (12%)</td>
<td>build information (60%)</td>
</tr>
<tr>
<td>build information (8%)</td>
<td>screenshots (60%)</td>
</tr>
<tr>
<td>product (5%)</td>
<td>test cases (56%)</td>
</tr>
<tr>
<td>operating system (4%)</td>
<td>error reports (53%)</td>
</tr>
<tr>
<td>component (3%)</td>
<td>stack traces (50%)</td>
</tr>
<tr>
<td>hardware (0%)</td>
<td>hardware (48%)</td>
</tr>
<tr>
<td>severity (0%)</td>
<td>code examples (36%)</td>
</tr>
</tbody>
</table>
Goal

- Bug Report Quality Estimator: A program that informs the reporter about the usefulness of the report

- Intuitive score as output

- Why would/should a reporter care?
Bug Report Quality Estimation

Bug Report → Vectorization → Bayesian Network → “usefulness” → Mapping → Quality Score (1-5)
Mapping

The graph shows the relationship between the number of reports and the usefulness of a service, with categories ranging from very bad to very good. The usefulness is quantified along the x-axis, while the number of reports is shown on the y-axis. The curve indicates that as the number of reports increases, the perceived usefulness also increases up to a certain point, after which it begins to decrease.
Problems and Limitations

• Dataset from the paper incomplete -> missing ground truth

• Quite a bit of speculation involved