A Study on Developers’ Perceptions About Exception Handling Bugs

Felipe Ebert
Fernando Castor

Centro de Informática – Universidade Federal de Pernambuco
Recife, Pernambuco, Brazil

Some rights reserved
Motivation

Error Handling Code

- “Happy path” is happier
- Often written in an ad hoc manner?
- Rarely tested or documented?
- Low quality?
- Source of bugs?
Motivation

Error Handling Code

- “Happy path” is happier
- Often written in an ad hoc manner?
- Rarely tested or documented?
- Low quality?
- Source of bugs?

Developer, what are you thinking?

We conducted a **survey** to find out.
Exception Handling Bugs

Bugs whose cause is related to exception handling.
Exception Handling Bugs

Bugs whose *cause* is related to exception handling.

More specifically, bugs related to the ...

- definition
- propagation
- throwing
- handling
- documentation

... of exceptions.
Research Questions

RQ1 Do organizations and developers pay attention to exception handling?

RQ2 How commonplace are exception handling bugs?

RQ3 Are exception handling bugs harder to fix than other bugs?

RQ4 What are the main causes of exception handling bugs?
Research Questions

RQ1  Do organizations and developers pay attention to exception handling?

RQ2  How commonplace are exception handling bugs?

RQ3  Are exception handling bugs harder to fix than other bugs?

RQ4  What are the main causes of exception handling bugs?
About the Survey

- 24 questions
- More than 4000 emails sent
- 154 respondents
Study Results
RQ1 – Do organizations and developers pay attention to exception handling?
Do organizations and developers pay attention?

<table>
<thead>
<tr>
<th>Specifications, policies, standards</th>
<th>73% – no</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tests specific for EH code</td>
<td>70% – no</td>
</tr>
<tr>
<td>Importance of EH in the design phase</td>
<td>61% – irrelevant or of little importance, 16% – important or very important</td>
</tr>
<tr>
<td>Quality of EH code</td>
<td>14% – bad or very bad, 40% – good or very good</td>
</tr>
</tbody>
</table>
### Study Results

#### Do organizations and developers pay attention?

<table>
<thead>
<tr>
<th>Category</th>
<th>Percentage</th>
<th>Rating</th>
</tr>
</thead>
<tbody>
<tr>
<td>Specifications, policies, standards</td>
<td>73% – no</td>
<td></td>
</tr>
<tr>
<td>Tests specific for EH code</td>
<td>70% – no</td>
<td></td>
</tr>
<tr>
<td>Importance of EH in the design phase</td>
<td>61% – irrelevant or of little importance</td>
<td>16% – important or very important</td>
</tr>
<tr>
<td>Quality of EH code</td>
<td>14% – bad or very bad</td>
<td>40% – good or very good</td>
</tr>
</tbody>
</table>

More experienced developers (10+ years) tend to believe that the quality is worse.  

p-value = 0.02478
Why do developers use exception handling?

<table>
<thead>
<tr>
<th>Reason</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. To create ways to tolerate faults</td>
<td>66%</td>
</tr>
<tr>
<td>2. To improve the quality of a functionality</td>
<td>63%</td>
</tr>
<tr>
<td>3. Importance of functionality</td>
<td>53%</td>
</tr>
<tr>
<td>4. Language requirement</td>
<td>43%</td>
</tr>
<tr>
<td>5. Organizational policies</td>
<td>21%</td>
</tr>
<tr>
<td>6. To debug a specific part of the code</td>
<td>17%</td>
</tr>
<tr>
<td>7. Does not use exception handling</td>
<td>2%</td>
</tr>
</tbody>
</table>

Reasons 6 and 7 were the most often cited in a previous study*

RQ3 – Are exception handling bugs harder to fix than other bugs?
Developers’ Opinions

- Level of difficulty to fix bugs related to exception handling
  - 43% consider them to be easy or very easy to fix

- Average level of difficulty to fix non-EH bugs
  - Only 7% consider them easy or very easy to fix

- Priority/Severity of exception handling bugs
  - Less experienced developers consider it to be lower (p-value = 0.0474)
RQ4 – What are the main causes of exception handling bugs?
Main Causes

1. Lack of a handler that should exist
2. No exception thrown in a situation of a known error
3. Programming error in the catch block
4. Programming error in the finally block
5. Exception is caught unintentionally
6. Catch block where only a finally would be appropriate
7. Exception that should not have been thrown
8. Wrong encapsulation of exception cause
9. Wrong exception thrown
10. Lack of a finally block that should exist
11. Error in the exception assertion
12. Empty catch block

Can guide inspection and test activities
A Note on Swallowed Exceptions

Have you ever needed to fix bugs related to exception handling?

Yes ⇒ 83%

Please describe some of these situations...

19 answers (1 in 6) mentioned empty catch blocks as causes

- Sometimes swallowing is considered a good thing!
Future Work

- Analysis of bug reports

Bugzilla – Bug 116223  [refactoring] ProcessorBasedRefactoring does not handle exceptions thrown by participants

Status: RESOLVED FIXED
Product: JDT
Component: UI
Version: 3.2
Hardware: PC Windows XP

Importance: P3 normal (vote)
Target Milestone: 3.2 M5
Assignee: Dirk Baeumer

QA Contact:
URL:
Whiteboard:
Keywords:

Depends on:
Blocks:

Reported: 2005-11-14 09:20 EST by Markus Keller
Modified: 2006-01-03 12:06 EST (History)

CC List: 0 users
See Also:
Thank You!

A Study on Developers’ Perceptions About Exception Handling Bugs

Felipe Ebert and Fernando Castor

Contact: castor@cin.ufpe.br

Discussion Topic

Should developers worry about exception handling (bugs)?
Backup Slides
This is not (necessarily) an exception handling bug

Example

```java
public void buggyMethod() throws IOException {
    File f = null;
    .... // should have initialized f but did not
    return f.getName(); // NullPointerException
}
```
Example

public void buggyMethod() throws IOException {
    File f = null;
    .... // f is still not initialized
    return f.getName(); // NullPointerException
}

public void handlerMethod() {
    try {
        ...
    } catch (IOException e) {
        String fileName = this.buggyMethod();
        // will throw NullPointerException but
        // handlerMethod expects IOException.
    }
}
Threats to Validity

- Developers may have misunderstood the definition of EH bug
- Relatively small number of respondents
- Questionnaire may lack important questions
### Java Experience

Most respondents have 7+ years

<table>
<thead>
<tr>
<th>Experience Range</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 2 years</td>
<td>7.14%</td>
</tr>
<tr>
<td>2 to 5 years</td>
<td>20.78%</td>
</tr>
<tr>
<td>5 to 7 years</td>
<td>14.94%</td>
</tr>
<tr>
<td>7 to 10 years</td>
<td>20.13%</td>
</tr>
<tr>
<td>More than 10 years</td>
<td>37.01%</td>
</tr>
</tbody>
</table>

### Project Size

42% currently work on projects with more than 100k LoC

<table>
<thead>
<tr>
<th>Size Range</th>
<th>Percentage</th>
</tr>
</thead>
<tbody>
<tr>
<td>Less than 20k LoC</td>
<td>14.03%</td>
</tr>
<tr>
<td>20k to 50k LoC</td>
<td>20.13%</td>
</tr>
<tr>
<td>50k to 100k LoC</td>
<td>13.64%</td>
</tr>
<tr>
<td>100k to 200k LoC</td>
<td>9.09%</td>
</tr>
<tr>
<td>More than 200k LoC</td>
<td>33.12%</td>
</tr>
</tbody>
</table>
RQ2 – How commonplace are exception handling bugs?
Developers’ Estimates

- Percentage of bugs related to exception handling code in projects
  - Mean 9.72%
  - Median 5%

- Exception handling bugs are reported less often
  - p-value < 0.0001

- Exception handling bugs are less frequent
  - p-value < 0.0001