

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



RQ1 – Do organizations and developers pay attention to exception handling?

Do organizations and developers pay attention?

Specifications, policies, standards

73% – **no**

Tests specific for EH code

70% – **no**

Importance of EH in the design phase

61% – **irrelevant or of little importance**

16% – important or very important

Quality of EH code

14% – **bad or very bad**

40% – good or very good



Do organizations and developers pay attention?

Specifications, policies, standards

73% – **no**

Tests specific for EH code

70% – **no**

Importance of EH in the design phase

61% – **irrelevant or of little importance**

16% – important or very important

Quality of EH code

14% – **bad or very bad**

40% – good or very good

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

p-value = 0.02478

Why do developers use exception handling?

1. To create ways to tolerate faults	66%
2. To improve the quality of a functionality	63%
3. Importance of functionality	53%
4. Language requirement	43%
5. Organizational policies	21%
6. To debug a specific part of the code	17%
7. Does not use exception handling	2%

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

* Hina Shah, Carsten Görg, Mary Jean Harrold: Understanding Exception Handling: Viewpoints of Novices and Experts. IEEE Trans. Software Eng. 36(2): 150-161 (2010)

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



The screenshot shows the Eclipse Bugzilla interface for bug report 116223. The header includes the Eclipse logo and the word 'BUGS'. The title of the bug is '[refactoring] ProcessorBasedRefactoring does not handle exceptions thrown by participants'. Below the title is a navigation bar with links for Home, New, Browse, Search, and a search input field. A message indicates that this bug is not in the last search results. The main content area displays the bug title and various details:

- Status:** RESOLVED FIXED
- Product:** JDT
- Component:** UI
- Version:** 3.2
- Hardware:** PC Windows XP
- Importance:** P3 normal ([vote](#))
- Target Milestone:** 3.2 M5
- Assigned To:** Dirk Baeumer (CLA)
- QA Contact:**
- URL:**
- Whiteboard:**
- Keywords:**
- Depends on:**
- Blocks:**

Additional information on the right side includes:

- 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

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

This is.

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

Who answered the survey?

Java Experience

Most respondents have 7+ years

- 7.14% Less than 2 years
- 20.78% 2 to 5 years
- 14.94% 5 to 7 years
- 20.13% 7 to 10 years
- 37.01% More than 10 years

Project Size

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

- 14.03% Less than 20k LoC
- 20.13% 20k to 50k LoC
- 13.64% 50k to 100k LoC
- 52.20% 100k + LoC



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