How Do Developers Select and Prioritize Code Smells? A Preliminary Study

Natthawute Sae-Lim, Shinpei Hayashi, and Motoshi Saeki

Department of Computer Science
School of Computing
Tokyo Institute of Technology
INTRODUCTION
Code smell

An indicator of a design flaw or a problem in the source code

- One of the factors that cause technical debt 😞
- Increases code component’s fault-proneness 😞

Data Class

“Classes that have fields, getting and setting methods for the fields, and nothing else.”

Feature Envy

“Every time you make a kind of change, you have to make a lot of little changes to a lot of different classes.”

The number of code smell is overwhelming.
Related Work

**Code Smells Prioritization**

<table>
<thead>
<tr>
<th>ICPC 2016</th>
<th>MTD 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Context-Based Code Smells Prioritization for Prefactoring</strong>&lt;br&gt;Sae-Lim <em>et al.</em></td>
<td><strong>Towards a Prioritization of Code Debt: A Code Smell Intensity Index</strong>&lt;br&gt;Fontana <em>et al.</em></td>
</tr>
</tbody>
</table>

**Code Smells Filtration**

<table>
<thead>
<tr>
<th>CSMR 2004</th>
<th>ICSE 2015</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Using history information to improve design flaws detection</strong>&lt;br&gt;Ratiu <em>et al.</em></td>
<td><strong>Filtering Code Smells Detection Results</strong>&lt;br&gt;Fontana <em>et al.</em></td>
</tr>
</tbody>
</table>
Related Work

Code Smells Prioritization

- Task relevance
- Smell severity

Code Smells Filtration

- Historical information
- False positive
No empirical evidence on how developers handle code smells
RQ1: What are the factors used by developers in the code smell selection process?

RQ2: What are the factors used by developers in the code smell prioritization process?
STUDY DESIGN
Data Collection

Selection

This smell should be solved because ...

Class A  God
Class B  Data
Class C  Blob
...
...

Code smells

Prioritization

This smell should be solved (in this order) because ...

1. Class C  Blob
2. Class A  God
...
...

Code smells

x10

x10
It involves many issues.

It is not a Blob Class after looking into the code.

3 issues came from this single class. This class is too generic.

Response

Task relevance

False positive

Task relevance, Smell severity

Codes
RESULTS
15 Final Codes

- Smell severity
- Smell coupling
- Co-located smells
- Smell false positive

- Task relevance
- Task importance
- Task implementation cost
- Task implementation risk

- Testability
- Readability
- Maintainability
- Understandability

- Module importance
- Module dependency

- Refactoring cost
RQ1: Selection Process

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task relevance</td>
<td>33</td>
</tr>
<tr>
<td>Smell severity</td>
<td>11</td>
</tr>
<tr>
<td>Task implementation cost</td>
<td>5</td>
</tr>
<tr>
<td>Testability</td>
<td>5</td>
</tr>
<tr>
<td>Co-located smells</td>
<td>4</td>
</tr>
</tbody>
</table>

**Factors considered together**

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Task relevance, Smell severity</td>
<td>9</td>
</tr>
<tr>
<td>Task relevance, Testability</td>
<td>5</td>
</tr>
</tbody>
</table>
### RQ2: Prioritization Process

#### Top 5 Factors

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module importance</td>
<td>14</td>
</tr>
<tr>
<td>Task relevance</td>
<td>10</td>
</tr>
<tr>
<td>Testability</td>
<td>5</td>
</tr>
<tr>
<td>Smell severity</td>
<td>4</td>
</tr>
<tr>
<td>Maintainability</td>
<td>3</td>
</tr>
</tbody>
</table>

#### Factors considered together

<table>
<thead>
<tr>
<th>Code</th>
<th>Number of responses</th>
</tr>
</thead>
<tbody>
<tr>
<td>Module importance, Task relevance</td>
<td>4</td>
</tr>
<tr>
<td>Module importance, Testability</td>
<td>3</td>
</tr>
</tbody>
</table>
CONCLUSION
### How do developers select and prioritize code smells?

#### Selection:
- Task relevance
- Smell severity

#### Prioritization:
- Module importance
- Task relevance
Factors that have been considered
- Smell severity
- Task relevance
- Smell false positive

Factors that have not been considered
- Testability
- Readability
- Smell coupling
- Maintainability
- Task importance
- Refactoring cost
- Co-located smells
- Understandability
- Module importance
- Module dependency
- Task implementation risk
- Task implementation cost