Context-Based Code Smells Prioritization for Prefactoring

Natthawute Sae-Lim, Shinpei Hayashi, and Motoshi Saeki

Department of Computer Science
Tokyo Institute of Technology
INTRODUCTION
Prefactoring[1]

Method A()
{
    __________
    __________
    __________
}

I need to implement feature X in method A()

Blob

Method A()
{
    __________
}

Refactoring

Easier!

Problem

Code smell detection results

I need to implement feature X in method A()

Problem: Results from existing smell detector do not fit prefactoring phase
Goal

Our technique

Original code smell detection result

Proposed code smell detection result

1st
2nd
3rd

.
.
.
nth

Smells that are relevant to developers’ context
Key Idea

Severity-based prioritization\textsuperscript{[1][2]}

I need to improve overall quality of the source code

Context-based prioritization

I need to implement feature $X$ in method $A()$

---

PROPOSED TECHNIQUE
Developers’ context

- Developers’ context = modules to be modified

Open issue #1
Fix a bug in login page

Open issue #50
Add “forgot password” feature

- This list is used to estimate developers’ context
Impact analysis

- Identify modules in source code that are likely to be affected by the changes
- Impact analysis → Change prediction

Change description #1

There is a bug in `LoginPage` that user can login successfully if leave the password field blank.

<table>
<thead>
<tr>
<th>Relevant modules</th>
</tr>
</thead>
<tbody>
<tr>
<td><code>LoginPage.login()</code></td>
</tr>
<tr>
<td><code>LoginPage.Reset()</code></td>
</tr>
<tr>
<td><code>UserPage.setPassword()</code></td>
</tr>
<tr>
<td><code>FormField.getPassword()</code></td>
</tr>
<tr>
<td>...</td>
</tr>
<tr>
<td><code>UserPage.ShowError()</code></td>
</tr>
</tbody>
</table>
Approach overview

Change descriptions
Source code

Bug 123
When click...

Main()
xxx;

Impact analysis
Code smell detection

TraceLab[1]
inFusion[2]

List of smells

Scoring

Prioritized smells

Scoring

Context Relevance Index
- Accumulating the score of matched modules in IA result

Code smell detection results

<table>
<thead>
<tr>
<th>Smell</th>
<th>Level</th>
<th>Module</th>
<th>CRI</th>
</tr>
</thead>
<tbody>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>Blob</td>
<td>Method</td>
<td>LoginPage.login()</td>
<td>0.4</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

Impact analysis results

#1

<table>
<thead>
<tr>
<th>Relevant modules</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>FormField.getPassword()</td>
<td>0.5</td>
</tr>
<tr>
<td>LoginPage.login()</td>
<td>0.1</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>

#50

<table>
<thead>
<tr>
<th>Relevant modules</th>
<th>Score</th>
</tr>
</thead>
<tbody>
<tr>
<td>UserPage.Reset()</td>
<td>0.7</td>
</tr>
<tr>
<td>LoginPage.login()</td>
<td>0.3</td>
</tr>
<tr>
<td>...</td>
<td></td>
</tr>
</tbody>
</table>
EMPIRICAL STUDIES
Empirical Studies

RQ2: Does the accuracy of IA affect quality of the ranking?

RQ3: Does Context-based smell prioritization provide more relevant results than the severity-based one?
Use Dit et al.’s benchmark dataset[1]

**Metric**

- **nDCG (Normalized Discounted Cumulative Gain)**
  - Metric for evaluating the quality of ranking documents
  - Relevant documents in **higher rank** are **more useful** than the ones in lower rank

- **Calculate nDCG for:**

- **Oracle**
  - Smells occurring in the modules *actually* modified during two releases
Empirical Studies

RQ2 : Does the accuracy of IA affect quality of the ranking ?

RQ3 : Does Context-based smell prioritization provide more relevant results than the severity-based one?
Impact analysis[1]

VSM | LSI | VSM+Dyn | LSI+Dyn

Cut points
5
10
20
30
40

VSM = Vector Space Model  
LSI = Latent Semantic Indexing  
Dyn = Dynamic Analysis

RQ2: Does the accuracy of impact analysis affect quality of the ranking?

Spearman’s correlation coefficient

Evaluate the association between two variables

Accuracy of IA tends to affect quality of the ranking by our technique
RQ2: Does the accuracy of IA affect quality of the ranking?

RQ3: Does Context-based smell prioritization provide more relevant results than the severity-based one?
RQ3: Does context-based smell prioritization provide more relevant results than the severity-based one?

YES!
### Baseline

<table>
<thead>
<tr>
<th>Rank</th>
<th>Smell Type</th>
<th>Class Name</th>
<th>Severity</th>
<th>#Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Blob</td>
<td>GeneratorCSharp</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>Blob</td>
<td>GeneratorJava</td>
<td>8</td>
<td></td>
</tr>
<tr>
<td>3</td>
<td>God</td>
<td>FigAssociation</td>
<td>8</td>
<td>5</td>
</tr>
<tr>
<td>4</td>
<td>Blob</td>
<td>ParserDisplay</td>
<td>8</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>Blob</td>
<td>GeneratorPHP4</td>
<td>7</td>
<td></td>
</tr>
<tr>
<td>6</td>
<td>RPB</td>
<td>FigClassifierRole</td>
<td>7</td>
<td>3</td>
</tr>
<tr>
<td>7</td>
<td>Blob</td>
<td>Modeller</td>
<td>7</td>
<td>1</td>
</tr>
<tr>
<td>8</td>
<td>SC</td>
<td>Import</td>
<td>6</td>
<td></td>
</tr>
<tr>
<td>9</td>
<td>God</td>
<td>CoreFactoryMDRImpl</td>
<td>5</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>RPB</td>
<td>StylePanelFigText</td>
<td>5</td>
<td></td>
</tr>
</tbody>
</table>

### Our approach

<table>
<thead>
<tr>
<th>Rank</th>
<th>Smell Type</th>
<th>Class Name</th>
<th>CRI</th>
<th>#Issues</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>God</td>
<td>Project</td>
<td>7.90</td>
<td>3</td>
</tr>
<tr>
<td>2</td>
<td>God</td>
<td>ProjectBrowser</td>
<td>4.04</td>
<td>7</td>
</tr>
<tr>
<td>3</td>
<td>Blob</td>
<td>ProjectBrowser</td>
<td>4.04</td>
<td>7</td>
</tr>
<tr>
<td>4</td>
<td>SC</td>
<td>StylePanel</td>
<td>2.43</td>
<td>1</td>
</tr>
<tr>
<td>5</td>
<td>God</td>
<td>FigNodeModelElement</td>
<td>2.18</td>
<td>4</td>
</tr>
<tr>
<td>6</td>
<td>God</td>
<td>UMLMutableGraphS</td>
<td>1.54</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>Blob</td>
<td>GeneratorCSharp</td>
<td>1.04</td>
<td></td>
</tr>
<tr>
<td>8</td>
<td>God</td>
<td>FigEdgeModelElement</td>
<td>0.94</td>
<td>3</td>
</tr>
<tr>
<td>9</td>
<td>God</td>
<td>ExtensionMechanism</td>
<td>0.91</td>
<td>1</td>
</tr>
<tr>
<td>10</td>
<td>God</td>
<td>CoreFactoryMDRImpl</td>
<td>0.80</td>
<td>1</td>
</tr>
</tbody>
</table>
CONCLUSION
Conclusion

Context-based code smells prioritization

Prefactoring   Automated

Accuracy of IA tends to impact the results

More relevant results than severity-based