FourD: “Do Developers Discuss Design?” Revisited

Abbas Shakiba
Robert Green
Robert Dyer

Bowling Green State University

supported in part by the US National Science Foundation under CCF-15-18776 and CNS-15-12947
Do developers discuss design decisions?

• Are design decisions only happening before implementation?

• Do design discussions/decisions show in the commit logs?
Prior work

• Brunet, João, et al. "Do developers discuss design?"
  11th Working Conference on Mining Software Repositories, 2014

• Selected set of 5 projects for analysis

• Analyzed:
  • commit logs
  • bug reports
  • discussions
Our Study

• Data from 2 software repositories
  • GitHub, SourceForge
• For each, 5 randomly selected projects
• Focus on commit logs
  • 200 randomly selected non-empty commits per project
• 2 x 200 x 5 = 2,000 commits total
• Train ML classifiers to identify commits discussing design
Tools Used

• Boa Language and Infrastructure
  • A language for analyzing ultra-large-scale software repositories

• Weka
  • Data Mining Tool written in Java

• Ruby on Rails
  • A web application framework written in Ruby
Approach

Getting Data (Boa)

Manual Classification (survey)

Pre-Processing (Weka)

Build Models (Weka)

Test Models (Weka)

Analyze Results
Approach (Cont'd)

- Boa queries
  - Randomly pick 5 projects (not shown)
  - Randomly pick 200 commits (shown)

```c
COMMITS: output top(200)[string] of string weight float;
ids := {"6176545", "6150849", "209281", "13151128", "1019785"};

isempty := function(s: string) : bool {
    s2 := trim(s);
    if (match(`^\s*$`, s2))
        return true;
    if (match(`^no message$`, lowercase(s2)))
        return true;
    if (match(`^*\*\* empty log message \*\*\*\*$`, lowercase(s2)))
        return true;
    return false;
};

exists (i: int; input.id == ids[i])
visit(input, visitor {
    before rev: Revision ->
    if (!isempty(rev.log))
        COMMITS[input.id] << rev.log weight rand();
});
```
Approach (Cont'd)

• Survey website for crowdsourcing

• Each log shown to 2-3 users

• Required 2 YES or 2 NO
• Convert data to ARFF format

• e.g., data1:
  “swapping the position of the input function <</>>”
  Classified: no

• e.g., data2:
  “reorganized a package structure to better reflect a layered approach”
  Classified: yes
Approach (Cont'd)

- Convert data to ARFF format
- Tokenization
  - Remove tokens without letters
- Stemming
  - Remove stop words
    - a, an, the, to, etc.
  - Eliminate prefix and suffix
    - -ing, -ed, -ly, etc.
Approach (Cont'd)

- Convert data to ARFF format
- Tokenization
  - Remove tokens without letters
- Stemming
  - Remove stop words
    - a, an, the, to, etc.
  - Eliminate prefix and suffix
    - -ing, -ed, -ly, etc.
Approach (Cont'd)

- Convert data to ARFF format
- Tokenization
  - Remove tokens without letters
- Stemming
  - Remove stop words
    - a, an, the, to, etc.
  - Eliminate prefix and suffix
    - -ing, -ed, -ly, etc.
Approach (Cont'd)

• Machine Learning Algorithms in Weka
  • Decision Tree
  • Random Forest
  • Naïve Bayes
  • Multinomial Bayes
  • Support Vector Machines
  • K-Nearest Neighbor
Difficulties

Different Data Distributions

Dataset 1

Dataset 2

Class: No

Class: Yes
Difficulties (Cont'd)

- Confusion Matrix
  - Add weight to cells
- Statistical measurements
  - F-Measure
  - G-Mean

\[
\text{Precision} = \frac{TP}{TP + FP} \quad \text{Recall} = \frac{TP}{TP + FN}
\]

\[
F_1 \text{score} = \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}
\]

\[
\text{Accuracy}^{\text{Yes}} = \frac{TP}{TP + FN} \quad \text{Accuracy}^{\text{No}} = \frac{TN}{TN + FP}
\]

\[
G_{\text{Mean}} = \sqrt{\text{Accuracy}^{\text{Yes}} \times \text{Accuracy}^{\text{No}}}
\]
### All Results

#### Decision Tree

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>86.18</td>
<td>98.3</td>
<td>12.2</td>
<td>82.8</td>
<td>68.50</td>
</tr>
<tr>
<td>sForge</td>
<td>87.39</td>
<td>97.6</td>
<td>25</td>
<td>84.7</td>
<td>74.81</td>
</tr>
<tr>
<td>Both</td>
<td>86.99</td>
<td>96.5</td>
<td>20.8</td>
<td>85.2</td>
<td>65.82</td>
</tr>
</tbody>
</table>

#### Random Forest

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>86.38</td>
<td>99.0</td>
<td>8.6</td>
<td>86.4</td>
<td>71.14</td>
</tr>
<tr>
<td>sForge</td>
<td>86.98</td>
<td>99.8</td>
<td>8.5</td>
<td>82.0</td>
<td>87.23</td>
</tr>
<tr>
<td>Both</td>
<td>86.44</td>
<td>99.5</td>
<td>5.7</td>
<td>81.4</td>
<td>75.01</td>
</tr>
</tbody>
</table>

#### Naive Bayes

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>75.17</td>
<td>76.2</td>
<td>69</td>
<td>78.6</td>
<td>54.72</td>
</tr>
<tr>
<td>sForge</td>
<td>81.48</td>
<td>84.2</td>
<td>65</td>
<td>83.2</td>
<td>61.33</td>
</tr>
<tr>
<td>Both</td>
<td>79.63</td>
<td>82.4</td>
<td>62.3</td>
<td>81.4</td>
<td>58.28</td>
</tr>
</tbody>
</table>

#### Support Vector Machine

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>85.48</td>
<td>93.7</td>
<td>34.5</td>
<td>84.7</td>
<td>64.95</td>
</tr>
<tr>
<td>sForge</td>
<td>86.27</td>
<td>93.7</td>
<td>40.7</td>
<td>85.7</td>
<td>68.21</td>
</tr>
<tr>
<td>Both</td>
<td>86.38</td>
<td>95.3</td>
<td>31.2</td>
<td>84.9</td>
<td>68.14</td>
</tr>
</tbody>
</table>

#### K-nearest neighbor

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>85.89</td>
<td>99.3</td>
<td>2.9</td>
<td>80.3</td>
<td>58.85</td>
</tr>
<tr>
<td>sForge</td>
<td>85.99</td>
<td>99.7</td>
<td>2.1</td>
<td>80.1</td>
<td>67.81</td>
</tr>
<tr>
<td>Both</td>
<td>86.18</td>
<td>99.9</td>
<td>1.8</td>
<td>80.1</td>
<td>80.16</td>
</tr>
</tbody>
</table>

#### Multinomial Bayes

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>74.27</td>
<td>74.8</td>
<td>71.2</td>
<td>78.0</td>
<td>54.33</td>
</tr>
<tr>
<td>sForge</td>
<td>70.57</td>
<td>70.0</td>
<td>73.5</td>
<td>85.5</td>
<td>51.85</td>
</tr>
<tr>
<td>Both</td>
<td>4.32</td>
<td>74.7</td>
<td>72.0</td>
<td>87.0</td>
<td>54.57</td>
</tr>
</tbody>
</table>

#### Build Models (Weka)

#### Test Models (Weka)

#### Pre-Processing (Weka)

#### Manual Classification (survey)
Interesting Results

### Random Forest

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>86.38</td>
<td>99.0</td>
<td>8.6</td>
<td>86.4</td>
<td>71.14</td>
</tr>
<tr>
<td>sForge</td>
<td>86.98</td>
<td>99.8</td>
<td>8.5</td>
<td>82.0</td>
<td>87.23</td>
</tr>
<tr>
<td>Both</td>
<td>86.44</td>
<td>99.5</td>
<td>5.7</td>
<td>81.4</td>
<td>75.01</td>
</tr>
</tbody>
</table>

### Support Vector Machine

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>85.48</td>
<td>93.7</td>
<td>34.5</td>
<td>84.7</td>
<td>64.95</td>
</tr>
<tr>
<td>sForge</td>
<td>86.27</td>
<td>93.7</td>
<td>40.7</td>
<td>85.7</td>
<td>68.21</td>
</tr>
<tr>
<td>Both</td>
<td>86.38</td>
<td>95.3</td>
<td>31.2</td>
<td>84.9</td>
<td>68.14</td>
</tr>
</tbody>
</table>

Increased False Negative Weight

<table>
<thead>
<tr>
<th>Data</th>
<th>Acc.</th>
<th>%TP</th>
<th>%TN</th>
<th>%F-M</th>
<th>G-mean</th>
</tr>
</thead>
<tbody>
<tr>
<td>GitHub</td>
<td>85.58</td>
<td>92.1</td>
<td>45.3</td>
<td>85.4</td>
<td>66.25</td>
</tr>
<tr>
<td>sForge</td>
<td>85.69</td>
<td>92.0</td>
<td>47.1</td>
<td>85.6</td>
<td>72.68</td>
</tr>
<tr>
<td>Both</td>
<td>85.34</td>
<td>90.1</td>
<td>50.8</td>
<td>85.5</td>
<td>64.61</td>
</tr>
</tbody>
</table>
F-measure and G-mean

GitHub

SourceForge
Future Work

• Move analysis completely into Boa
  • Pre-processing tasks
  • Machine learning models

• Do developers discuss other topics?
  • testing
  • debugging
  • etc.
To summarize...

Confusion Matrix

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th>Design</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td></td>
<td></td>
</tr>
<tr>
<td>No</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Survey Questions

Likelihood Table

<table>
<thead>
<tr>
<th>Data</th>
<th>Refactor</th>
<th>Fix</th>
<th>Test</th>
<th>struct</th>
</tr>
</thead>
<tbody>
<tr>
<td>Yes</td>
<td>14</td>
<td>3</td>
<td>2</td>
<td>18</td>
</tr>
<tr>
<td>No</td>
<td>6</td>
<td>16</td>
<td>31</td>
<td>39</td>
</tr>
</tbody>
</table>

Support Vector Machine

| | | | | | |
|---|---|---|---|---|
| Data | Acc. | %TP | %TN | %F-M | G-mean |
| GitHub | 85.48 | 93.7 | 34.5 | 84.7 | 64.95 |
| sForge | 86.27 | 93.7 | 40.7 | 85.7 | 68.21 |
| Both | 86.38 | 95.3 | 31.2 | 84.9 | 68.14 |
| Increased False Negative Weight |
| GitHub | 85.58 | 92.1 | 45.3 | 85.4 | 66.25 |
| sForge | 85.69 | 92.0 | 47.1 | 85.6 | 72.68 |
| Both | 85.34 | 90.1 | 50.8 | 85.5 | 64.61 |

To http://boa.cs.iastate.edu/