The Influence of Organizational Structure on Software Quality: An Empirical Case Study

Nachiappan Nagappan
Microsoft Research
Redmond, WA, USA
nachin@microsoft.com

Brendan Murphy
Microsoft Research
Cambridge, UK
bmurphy@microsoft.com

Victor R. Basili
University of Maryland
College Park, MD, USA
basili@cs.umd.edu

Presentation by Etan Kissling
18.05.2010
SOFTWARE METRICS
Code Churn

• Software change history
• Large / recent changes

• Total added, modified and deleted LOC
• Number of times that a binary was edited
• Number of consecutive edits
Code Complexity

• Gathered from code itself
• Multiple complexity values

• Cyclomatic complexity
• Fan-In / Fan-Out of functions
• Lines of Code
• Weighted methods per class
• Depth of Inheritance
• Coupling between objects
• Number of subclasses
• Total global variables
Dependencies

• Components that a class uses
• Both data and call dependencies

• Incoming / outgoing direct / indirect dependencies to a binary
• Layer information: Distance of a binary from the system kernel
Code coverage

• Degree to which the source code is tested

```plaintext
code =

foo (x: INTEGER; y: INTEGER): INTEGER
local
c: INTEGER
begin
  c := y
  if x > 5 and y > 0 then
    c := x
  end
  Result := x * c
end
```
Statement coverage

- Has each node in the program been executed?

```plaintext
foo (x: INTEGER; y: INTEGER): INTEGER

local
    c: INTEGER

    do
        c := y
        if x > 5 and y > 0 then
            c := x
        end
    end

    Result := x * c
end
```

A testing suite which includes \texttt{foo(7, 1)} would cover all statements of this code.
Branch coverage

• Has each control structure been evaluated both to true and false?

```plaintext
foo (x: INTEGER; y: INTEGER): INTEGER
  local
    c: INTEGER
  do
    c := y
    if x > 5 and y > 0 then
      c := x
    end
  end
  Result := x * c
end
```

foo(7, 1) and foo(7, 0) together would cover this branch completely
Pre-release defects

- Number of pre-release bugs found in a binary
- Strong relationship between development defects per module and field defects per module
ORGANIZATIONAL METRICS
Total edits = 250  Total Engineers Editing = 32  Total Ex-Engineers (edited) = 0
Number of Engineers

- Touched a binary
- Still employed by the company

› The more people who touch the code the lower the quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Number of Ex-Engineers

- Touched a binary
- Left the company

- A large loss of team members affects the knowledge retention and thus quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Edit Frequency

• Number of edits

› The more edits to components the higher the instability and lower the quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Depth of Master Ownership (DMO)

- Level of ownership
- More than 75% of the edits done by engineers which report to the owner
  
  › The lower level is the ownership the better is the quality
Organizational Structure

Level 0
- AB
- AC
- AD

Level 1
- ABA
- ABB
- ABC
- ACA
- ADA

Level 2
- ABCA

Org Size
No. of Edits
- A
- 30
- 10

E1
- 30
E2
- 45
E3
- 60
E4
- 25
E5
- 30
E6
E7

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Percentage of Org contributing to development

- Number of people reporting at the DMO level
  Master owner org size

› The more cohesive are the contributors (organizationally) the higher is the quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Level of Organizational Code Ownership

• If there is an owner:
  Percent of edits from the owner’s organization

• If there is no owner:
  Percent of edits from the organization which made the majority of edits

› The more cohesive are the contributions (edits) the higher is the quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
Overall Organization Ownership

- Number of people at the DMO level making edits
  \[ \frac{\text{Total Engineers Editing}}{} \]

- The more the diffused contribution to a binary, the lower is the quality.
Organizational Structure

- Total edits = 250
- Total Engineers Editing = 32
- Total Ex-Engineers (edited) = 0

Diagram:
- Org Size
  - No. of Edits
- A
  - AB
    - ABA
    - ABB
  - ABC
  - AC
  - AD
- ABCA
  - 5 engineers edit
  - 190 edits done

Nodes:
- E1
- E2
- E3
- E4
- E5
- E6
- E7
- E8
- E9

Numbers:
- 5/32
- 30
- 200
- 40
- 40
- 30
- 10
- 30
- 25
- 30
- 30
Organization Intersection Factor

• Number of different organizations that contribute greater than 10% of edits

› The more diffused the different organizations contributing code, the lower is the quality
Organizational Structure

Total edits = 250
Total Engineers Editing = 32
Total Ex-Engineers (edited) = 0
CASE STUDY
Case study

- Windows Vista:
  3404 binaries
  50+ Million LOC

- Access to people management software to build tree maps for organizational metrics

- 50 random splits:
  2/3 to build prediction model
  1/3 to verify prediction accuracy
Precision and recall

<table>
<thead>
<tr>
<th>Actual</th>
<th>Predicted</th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Not Failure-prone</td>
<td>Failure-prone</td>
</tr>
<tr>
<td>Not failure-prone</td>
<td>A</td>
<td>B</td>
</tr>
<tr>
<td>Failure-prone</td>
<td>C</td>
<td>D</td>
</tr>
</tbody>
</table>

**Precision** = \( \frac{d}{b + d} \)
Percentage of correct failure-prone predictions

**Recall** = \( \frac{d}{c + d} \)
Percentage of correctly identified failure-prone binaries
## Comparization

<table>
<thead>
<tr>
<th>Model</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>Organizational Structure</td>
<td>86.2%</td>
<td>84.0%</td>
</tr>
<tr>
<td>Code Churn</td>
<td>78.6%</td>
<td>79.9%</td>
</tr>
<tr>
<td>Code Complexity</td>
<td>79.3%</td>
<td>66.0%</td>
</tr>
<tr>
<td>Dependencies</td>
<td>74.4%</td>
<td>69.9%</td>
</tr>
<tr>
<td>Code Coverage</td>
<td>83.8%</td>
<td>54.4%</td>
</tr>
<tr>
<td>Pre-Release Bugs</td>
<td>73.8%</td>
<td>62.9%</td>
</tr>
</tbody>
</table>
Threats to validity

• Internal validity:
  Influence of study to Windows

• Construct validity:
  Errors in measurement

• External validity:
  All data from one software system