

ISE08

International Conference on
Innovation in Software Engineering

 ISWC 2008

4th International Workshop
on Semantic Web Enabled Software Engineering



UNIVERSITÉ
Concordia

UNIVERSITY

Mining Bug Repositories for Textual Quality

Philipp Schügerl, Juergen Rilling, Philippe Charland*

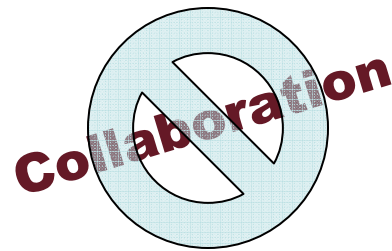
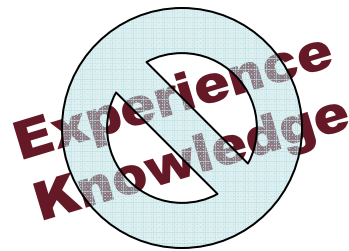
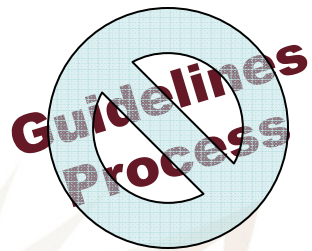
Ambient Software Evolution Group,
Concordia University, Montreal
aseg.cs.concordia.ca

*System of Systems Section
Defence R&D Canada Valcartier
Quebec, Canada

Motivation

- Software Engineering is a systematic and disciplined approach ¹

...so why does the performance of software engineers vary?



1: IEEE Standard Glossary of Software Engineering Terminology," [IEEE](#) std 610.12-1990, 1990

Manifesto

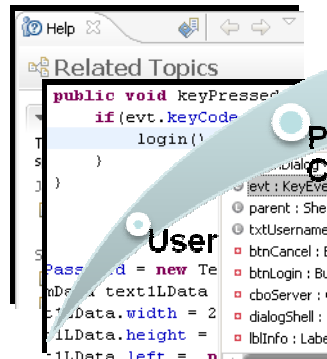
1. We need to make process and guidelines explicit and verifiable
2. We need better ways to capture explicit and implicit domain knowledge/experience and provide it to developers when they need it
3. We need to improve collaboration and workflows
4. A solution must be easy to use (integrated)

SE-ADVISOR System

Context

SE Ontology

Advice

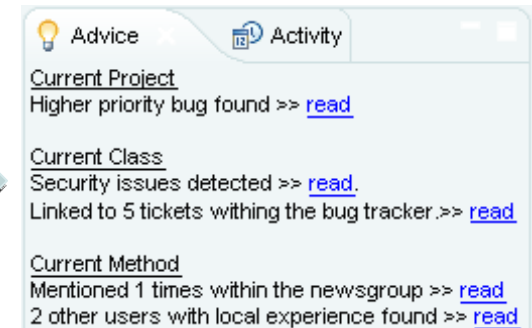


```
public void keyPressed(  
    if (evt.getKeyCode() == KeyEvent.VK_ENTER) {  
        login();  
    }  
}
```

Related Topics

- Project, Class
- User

Process step



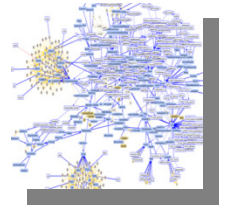
Advice Activity

Current Project
Higher priority bug found >> [read](#)

Current Class
Security issues detected >> [read](#).
Linked to 5 tickets withing the bug tracker.>> [read](#)

Current Method
Mentioned 1 times within the newsgroup >> [read](#)
2 other users with local experience found >> [read](#)

SE Ontology



- Unified representation of resources
 - Sourcecode, versioning, bugtracking,... sub-ontologies
 - “The following documents, source code parts and wiki pages are affected by a change in this class”
- Formal representation of domain specific knowledge
 - For example: define concept “bad programmer”
 - “This class has a higher chance of containing a bug because a bad programmer has originally written it [and it hasn’t been modified]”

Back to...

MINING BUG REPOSITORIES FOR TEXTUAL QUALITY

Contribution

- Defining bug report quality attributes
- Automatic quality assessment for bug reports
- Evaluating classification performance

Bug Quality – General Guidelines

- Be precise
- Explain it so others can reproduce it
- One bug per report
- Clearly separate fact from speculation
- No abuse or whining about decisions

[Bettenburg 2008]

[Tatham 2008]

Bug Quality Guidelines 1/2

- **Certainty.** There should be no speculation in a bug description.

To Me it seems not to be possible to create a class within a diagram from a different package?

(ArgoUML Bug# 378)

- **Focus.** There should be no off-topic discussions, complaints or personal statements. Only one bug should be described per report.

Hi, I'm a very new user to ArgoUML. I found it exciting.....Direct to the point.....

(ArgoUML Bug# 236)

Bug Quality Guidelines 2/2

- **Reproducibility.** There should be a clear way to reproduce the bug (context in which the bug occurred).

After adding a diagram (class/state), I couldn't delete it from the project...

(ArgoUML Bug# 269)

- **Observability.** The observed negative behavior should be described.

*GUI **hangs** when attempting to bold text...*

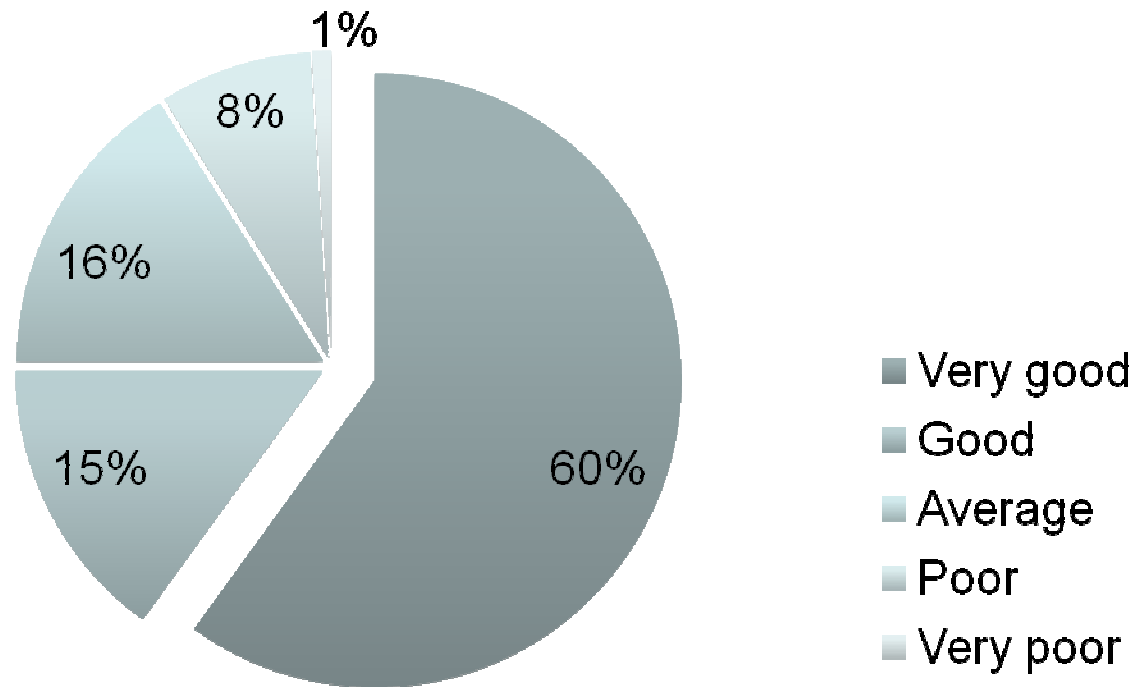
(ArgoUML Bug# 364)

Bug Quality Classification

Predicted	Observed					<i>Precision</i>
	Very good	Good	Average	Poor	Very poor	
Very good	15	4	2	0	1	86%
Good	29	36	7	5	2	91%
Average	1	6	2	0	1	80%
Poor	0	6	3	3	3	60%
Very poor	1	2	5	1	3	33%
<i>Recall</i>	96%	85%	63%	44%	60%	

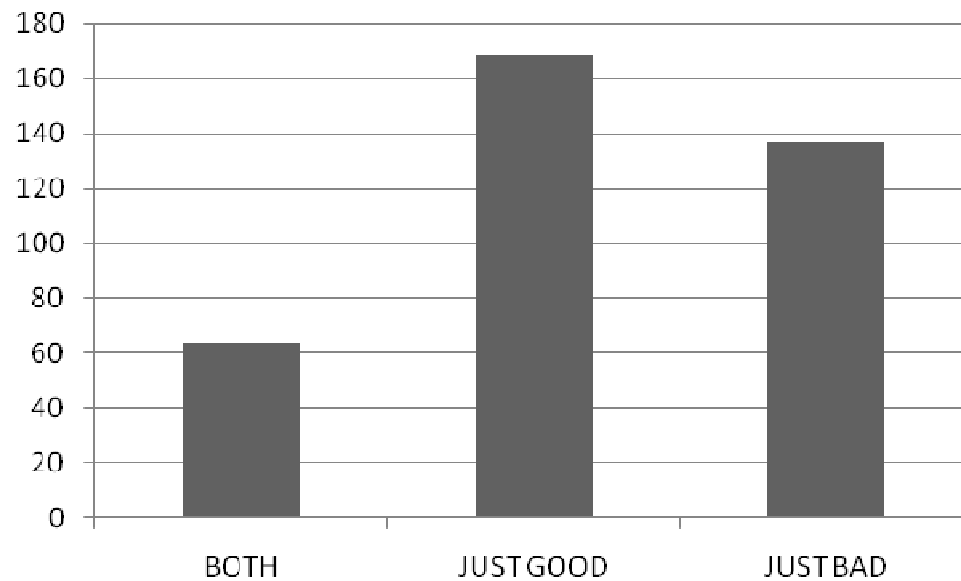
Accuracy: 81%

Automatic Quality Assessment



ArgoUML bugtracker

Reporters per Quality Level



Ambient Software Evolution Group

aseg.cs.concordia.ca



UNIVERSITÉ
Concordia

UNIVERSITY

www.concordia.ca

