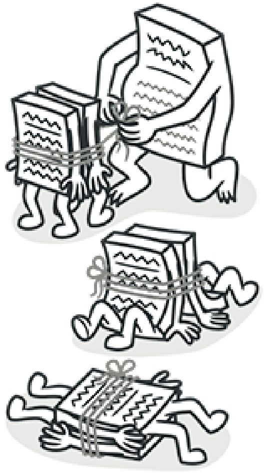
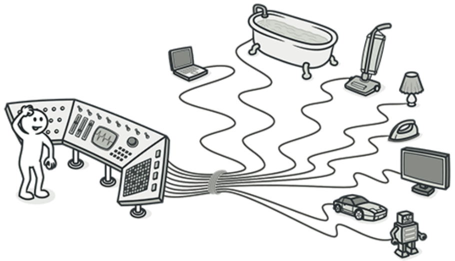
On the Influence of Human Factors for Identifying Code Smells

A Multi-Trial Empirical Study





Rafael Maiani de Mello
Department of Informatics, PUC-Rio

Roberto Oliveira
Department of Informatics, PUC-Rio

Alessandro Garcia
Department of Informatics, PUC-Rio

05/03/2023

Presented To

Dipok Chandra Das

Assistant Professor
Institute of Information Technology(IIT)
Noakhali Science and Technology University

Presented By

Prosanto Deb

ASH1925005M
Year 03, Term 02
Institute of Information Technology(IIT)
Noakhali Science and Technology University

Table of contents

Problem Domain

Code smells are symptoms of poor design problems observed in the low-level structure of a software system

The study investigates the influence of three human factors

Professional background

To reach high precision rates

Module knowledge

Previous background in code reviews is not essential to reach higher precision

Collaboration

Significantly increases the precision of code smell identification

Background and Related Works

M. Fowler, 1999

“Refactoring: Improving the Design of Existing Code”. A **Long Method** tends to **implement two or more concerns** that **should not be together** in a **single method**.

I Macia, 2012

“Are Automatically-Detected Code Anomalies Relevant to Architectural Modularity?” **Relied on the reports of the developers of the studied systems to support their analyses.**

J. Padilha, 2014

“On the Effectiveness of Concern Metrics to Detect Code Smells: An Empirical Study”. **Same as I Macia,2012**

M. Ferreira, 2014

“Detecting Architecturally-Relevant Code Anomalies: A Case Study of Effectiveness and Effort”. **Relied** on the **reports** of a **single reviewer.**

F. Palomba, 2014

“Do They Really Smell Bad? A Study on Developers' Perception of Bad Code Smells,” found that **developers without system knowledge** frequently reported **more code smells**, such as Feature Envies.

R. Oliveira, 2016

“Identifying Code Smells with Collaborative Practices: A Controlled Experiment” found that **reviewers working in collaboration** tend to **share** complementary **knowledge** and **reach** consensus about **smell occurrences.**

Strategy

Step 1
Population, Samples, and Systems

Step 2
Modules Selection and Code Smells

Step 3
Composition of the Reference List

Step 4
Experimental Design

Step 5
Validation

Population, Samples, and Systems

Text T1 describes a study conducted in a Software Engineering class from a Brazilian University with 28 bachelor students as subjects.

- The subjects were novice developers
- Had no previous background in industrial software projects
- Did not have any previous knowledge of such systems

while others had worked on an Agile Software Development course called Software Kaizen.

Modules Selection and Code Smells

In the scope of T1, the following subset of code smells was considered:

- Data Class
- Duplicated Code
- Lazy Class
- Long Method
- Message Chain

These types were selected because previous studies have reported that they occur most frequently.

Composition of the Reference List

The researchers involved in the development of the three systems established the reference list for the code smells detected by the smell detection tool in the selected modules. They,

- validated each code smell individually
- attended a meeting to reach a consensus on any conflicts

The reference list was not definitive and was updated after the execution of the identification tasks. The whole process took 92 man-hours.

Experimental Design

A **training session** was conducted to mitigate possible differences in awareness and knowledge on code smells.

The subjects were then **asked to identify code smells** in modules of each system along **three rounds** with a **duration of 60 minutes** each.

The authors adopted a crossed design to mitigate threats to validity concerning the influence of the learning curve over the results and the fact that one task could favor a specific treatment.

Experimental Design

<table>
<tr><th colspan="4">NoBg</th><th colspan="4">LitBg</th></tr>
<tr><th>Subject</th><th>R1</th><th>R2</th><th>R3</th><th>Subject</th><th>R1</th><th>R2</th><th>R3</th></tr>
<tr><td>T1S01</td><td>A</td><td rowspan="2">B</td><td rowspan="6">C</td><td>T1S15</td><td>A</td><td rowspan="2">B</td><td rowspan="6">C</td></tr>
<tr><td>T1S02</td><td>A</td><td>T1S16</td><td>A</td></tr>
<tr><td>T1S03</td><td>A</td><td rowspan="2">B</td><td>T1S17</td><td>A</td><td rowspan="2">B</td></tr>
<tr><td>T1S04</td><td>A</td><td>T1S18</td><td>A</td></tr>
<tr><td>T1S05</td><td>A</td><td rowspan="2">B</td><td>T1S19</td><td>A</td><td rowspan="2">B</td></tr>
<tr><td>T1S06</td><td>A</td><td>T1S20</td><td>A</td></tr>
<tr><td>T1S07</td><td rowspan="4">A</td><td>B</td><td rowspan="2">C</td><td>T1S21</td><td rowspan="4">A</td><td>B</td><td rowspan="2">C</td></tr>
<tr><td>T1S08</td><td>B</td><td>T1S22</td><td>B</td></tr>
<tr><td>T1S09</td><td>B</td><td rowspan="2">C</td><td>T1S23</td><td>B</td><td rowspan="2">C</td></tr>
<tr><td>T1S10</td><td>B</td><td>T1S24</td><td>B</td></tr>
<tr><td>T1S11</td><td rowspan="2">A</td><td rowspan="4">B</td><td>C</td><td>T1S25</td><td rowspan="2">A</td><td rowspan="4">B</td><td>C</td></tr>
<tr><td>T1S12</td><td>C</td><td>T1S26</td><td>C</td></tr>
<tr><td>T1S13</td><td rowspan="2">A</td><td>C</td><td>T1S27</td><td rowspan="2">A</td><td>C</td></tr>
<tr><td>T1S14</td><td>C</td><td>T1S28</td><td>C</td></tr>
</table>

Figure: Experimental Design From T1.

NoBg: No professional background
LitBg: Little background

The authors selected three functional software projects – named as A, B, and C

Experimental Design

Develoeprs with Module Knowledge				Developers without Module Knowledge			
Prof. Background	**Subject**	**R1**	**R2**	**Prof. Background**	**Subject**	**R1**	**R2**
MidBg	T2S01	D	D	HighBg	T2S14	F	F*
MidBg	T2S02	D		HighBg	T2S19		F*
MidBg	T2S03	D		HighBg	T2S15	F	F
HighBg	T2S04	E	E	HighBg	T2S18		F
MidBg	T2S05	E		MidBg	T2S17	F	F
HighBg	T2S06	E	E	MidBg	T2S22		F
HighBg	T2S07	E		MidBg	T2S24		F
HighBg	T2S08	F	F	HighBg	T2S16	F	F
HighBg	T2S09		F	MidBg	T2S20		F
HighBg	T2S10	G	G	HighBg	T2S21	F	F*
MidBg	T2S11		G	HighBg	T2S23		F*
HighBg	T2S12	H	H				
MidBg	T2S13		H				

Figure: Experimental Design From T2

Evalution of Human Factors

The precision of **smell identification** achieved by reviewers with **different levels of professional background** are **not different.**

The precision of **smell identification** achieved by reviewers **with and without module knowledge** are **not different.**

The precision of **smell identification** performed **individually** and **collaboratively** are **not different.**

The precision of **smell identification** achieved by **solo reviewers** with and without module knowledge are **different**.

The precision of **smell identification** achieved by **solo reviewers** with different levels of professional background **are different.**

The precision of **smell identification** achieved by **collaborative reviewers** with and without module knowledge are **different**.

Results and Findings

Sample	LitBg	MidBg	MHighBg
NoBg	0.547	*0.011*	0.086
LitBg	-	0.054	0.267
MidBg	-	-	0.400

Figure: P-VALUES: COMPARING PRECISION OF SOLO IDENTIFICATION BY PROFESSIONAL BACKGROUND.

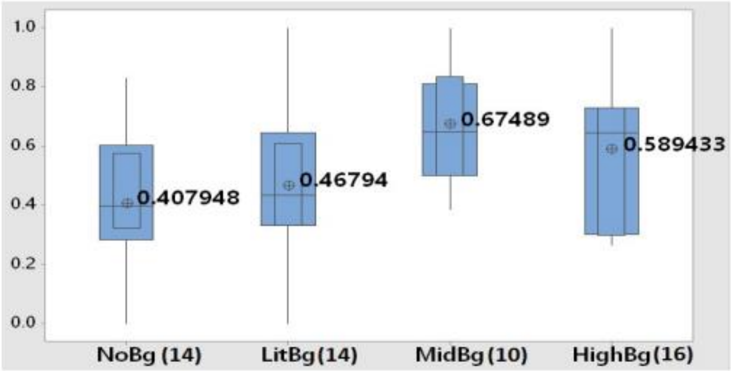


Figure: Precision by professional background: solo identification.

Even developers with **high professional background** tend not to reach **high precision** through **solo smell identification**

Results and Findings

Group	LitBg	MHighBg
NoBg	*0.029 (0.009)*	*0.039*
LitBg	-	0.790 (0.641)

Figure: P-VALUES: COMPARING PRECISION OF COLLABORATIVE IDENTIFICATION BY PROFESSIONAL BACKGROUND.

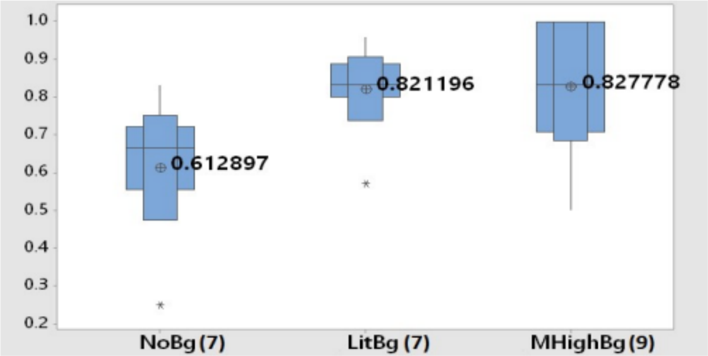


Figure: Precision by professional background: collaborative identification

Collaborative identification performed by developers with **some** professional background **tend to reach high precision**.

Results and Findings

Sample	HighBg-NoMK	HighBg-MK
LitBg-NoMK	*0.005*	0.072
HighBg-NoMK	-	0.123

Figure: P-VALUES: COMPARING PRECISION OF SOLO IDENTIFICATION BY MODULE KNOWLEDGE AND PROFESSIONAL BACKGROUND.

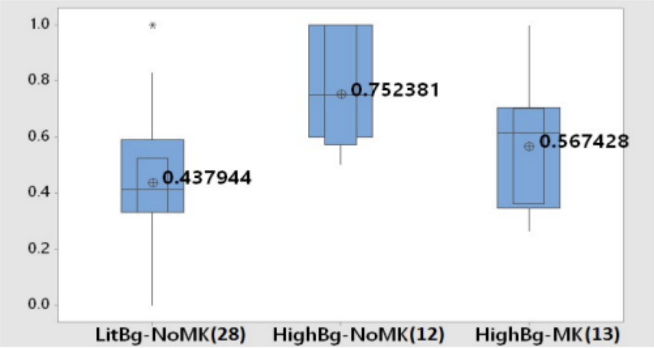


Figure: Precision by module knowledge: solo identification

Results and Findings

Sample	MHighBg-NoMK	MHighBg-MK
NoBg-NoMK	0.080	0.089
LitBg-NoMK	0.963	0.929
MHighBg-NoMK	-	0.969

Figure: P-VALUES: COMPARING PRECISION OF COLLABORATIVE IDENTIFICATION BY MODULE KNOWLEDGE AND BACKGROUND.

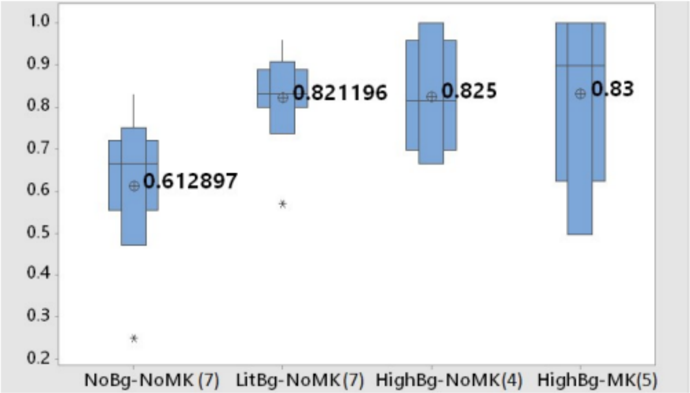


Figure: Precision by module knowledge: collaborative identification

Module knowledge lead **experienced reviewers** to focus on types of smells different from those **frequently reported by experienced reviewers** without this knowledge.

Results and Findings

Sample	Solo Round 2	Collaborative
Solo Round 1	0.994	*0.014*
Solo Round 2	-	0.072

Figure: P-VALUES: COMPARING PRECISION OF SOLO AND COLLABORATIVE IDENTIFICATION FROM T2.

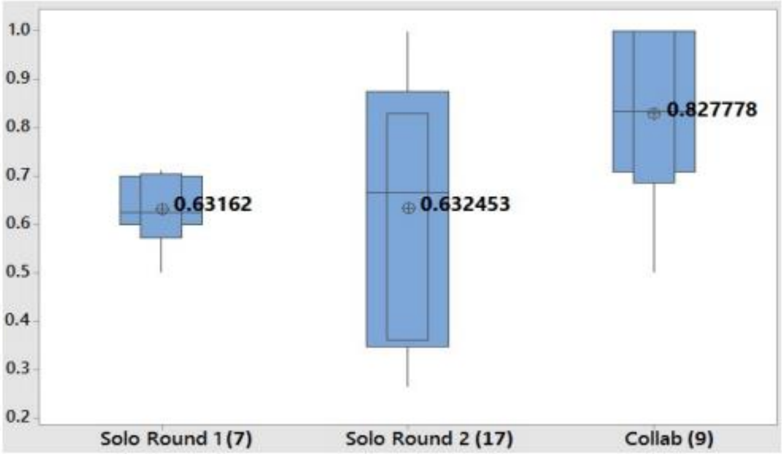


Figure: Precision in Trial 2: solo and collaborative identification.

Collaborative identification tasks performed by reviewers with a **medium-high professional** background tend to reach **high precision** when compared to **solo identification.**

Limitations

Potential Threats

To the validity of a study on identifying code smells in software development

Differences in the protocol

In the trials, such as the number of code smells and complexity of the modules

Time Limit

The limited time for the tasks is also a threat, but has minimal impact on the results.

Uneven Sample Size

Trial was limited to Brazilian students and developers were external threats to validity

Conclusion

These findings can help **researchers** and **project managers plan** and **allocate resources** for **smell identification** tasks more effectively.

Future Work

It includes **investigating** the influence of other **contextual factors** and developing **evidence based guidelines** to support the **planning and conduction** of smell identification tasks.

Thanks

Do you have any questions?

Also available at prosato2514@student.edu.bd