

A New approach for fixing Bugs in Code Clones:

Fix It There Too (FITT)

Vaibhav Saini

Aftab Hussain

FIX IT THERE TOO

Area

Clone Detection for enhancing Fault Localization

75% of cases, the bug pattern associated with the code is duplicated as-is when the code is cloned.
(Hitesh et. Al, 2014).

While fixing a bug in a piece of code, it is very unlikely that the developer is aware of all the places where this code has been cloned in the project code

Research Problem

How could we simplify clone detection in our IDE so we could enhance fault localization?

Objective

To build a plugin for eclipse that will proactively inform developers about the existing clones of the piece of code on which they are currently working.

Proposal

IF there is a tool that gives us information on clones of a code segment *while* we are working on it **THEN** our awareness about the presence of clones in our project will be improved, which will enable us to take appropriate measures while we are correcting any of the clone siblings.

How often do they search for code-siblings after fixing code in one of them?

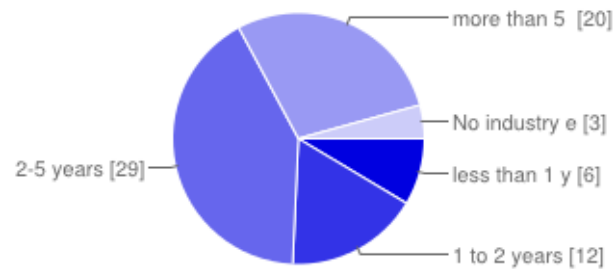
How much of cloning do they do anyway?

Would you really want this kind of a tool?

FIX IT THERE TOO

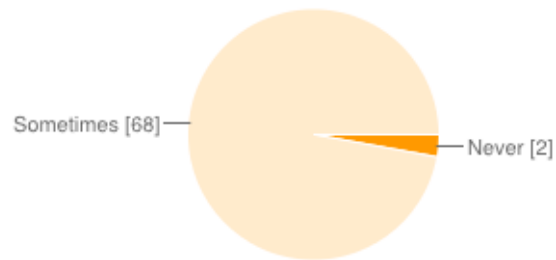
Evidence

1. How much industry experience do you have?



less than 1 year	6	9%
1 to 2 years	12	17%
2-5 years	29	41%
more than 5 years	20	29%
No industry experience	3	4%

2. How often do you copy some piece of code from the project that you are working on to use it in another module or functionality?

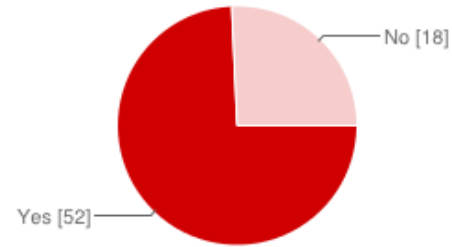


Never	2	3%
Sometimes	68	97%

FIX IT THERE TOO

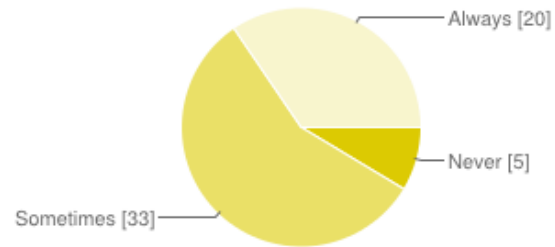
Evidence

3. While fixing a bug in a piece of code, do you actively search for similar pieces of code in the project?



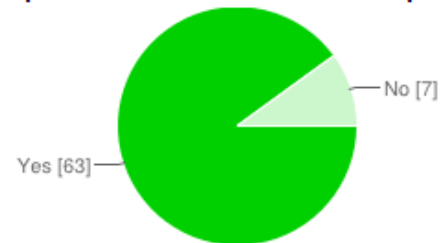
Yes	52	74%
No	18	26%

4. If your answer to question 3 was yes, how often do you search for similar pieces of code?



Never	5	9%
Sometimes	33	57%
Always	20	34%

5. Would you like to have a plugin for your IDE, that can highlight the pieces of code (like a function) that is similar to other pieces of code in the current project?



Yes	63	90%
No	7	10%

FIX IT THERE TOO

Scope

Eclipse - Java - Method Level
Clone Qualification Threshold of
80%

Overlap Similarity

=

$$\frac{\text{No. of Unique Common Elements}}{\text{No. of Elements in larger module}}$$

Evaluation Plan

We will hire two developers with
2-4 years of industry experience

Have them fill a survey based on a
month's use of our Plugin.

Related Work

* Kasper and Godfrey [1] presented evidence that clones may be intentional and improve developer productivity.

* Toomim et al. [3] showed that managing clones via linked editing to edit multiple cloned regions without much programmer intervention can be an efficient way of dealing with clones.

* Kim et. al [2] found that most of the clones are short lived and hence investment made in refactoring them may not be worth the effort.

* Rahman et al. [4] did not find any evidence that cloning is harmful based on their impact on defect occurrence.

References

- [1] **C. Kapser and M. Godfrey. "cloning considered harmful" considered harmful: patterns of cloning in software. Empirical Software Engineering, 13(6):645{ 692, 2008.**
- [2] **M. Kim, V. Sazawal, D. Notkin, and G. Murphy. An empirical study of code clone genealogies. In Proceedings of FSE, 2005.**
- [3] **M. Toomim, A. Begel, and S. Graham. Managing duplicated code with linked editing. In Visual Languages and Human Centric Computing, 2004 IEEE Symposium on, pages 173-180, 2004.**
- [4] **F. Rahman, C. Bird, and P. Devanbu. Clones: what is that smell? Empirical Software Engineering, 17(4-5):503{ 530, 2012.**

Thank You

:)