Survey on Testing Developers Practices.

1. Survey on Testing Developers Practices

INTRODUCTION

Our research group at the University of Zurich together with the Delft University of Technology research group, is performing a study aimed at observing (and/or measuring) how developers perform bug fixing tasks during test activity. Specifically, tools such as Evosuite (http://www.evosuite.org/) can automatically generate test cases to help developers test Java classes and find possible bugs. Thus, it is our goal to better understand the bug fixing practice of developers during their testing activities when relying on generated testcases (by Evosuite).

We would be grateful if you could perform the 2 testing tasks on 2 Java Classes (Rational.java, and ArrayIntList.java) belonging to Math4J (http://sourceforge.net/projects/math4j/), and

Apache Commons Primitives (https://commons.apache.org/proper/commons-primitives/) respectively (they are two software system implemented in Java) you received vie e-mail. You should have received these Java classes via e-mail. For each task you can modify/add new test cases to those generated by Evosuite (http://www.evosuite.org/).

LET'S START

Note that you can perform the tasks in multiple rounds (e.g., the first 1 task today and the last tomorrow). The system will automatically propose you the right task at each round, hiding those that you already performed. TO ALLOW THIS, you MUST use the same PC and the same browser each time you connect to the survey. On the contrary, please do not interrupt while performing a task.

You received via e-mail the instructions to download an Eclipse workspace installation with the 2 projects mentioned above. Note that, together with the source code, you will also find, inside the workspace, the test cases generated by Evosuite for the selected Java classes.

When you have completed all tasks, and after having closed Eclipse, please zip the workspace folder (where also the projects are contained) and send it to the following e-mail addresses:

panichella@ifi.uzh.ch, a.panichella@tudelft.nl

THE DEADLINE TO COMPLETE THE QUESTIONNAIRE IS END of JULY.

Thank you very much for your effort,

Dr. Sebastiano Panichella and

Dr. Annibale Panichella

1.

All of the information that you provide will be treated as confidential and will only be used for research purposes. Some personal information may be collected about you if you choose to participate in the survey. In particular, your responses to survey questions and your personal details will be collected for the simply reason of linking your responses with the versions of the Java projects we sent to you via e-mail. We will not disclose your personal information to third parties.

Subject Information:

*

Name Participant:

Surname Participant:

2.You are a: *

- Bachelor's student
- Master's student
- Ph.D. student
- Faculty
- Professional Developer

3. You program in java since: *

- <1 year
- 1-2 years
- 3-6 years
- 7-10 years
- >10 years

4.How many years of testing experience do you have?

- less than one year
- <1 year
- 1-2 years
- 3-6 years
- 7-10 years

○ >10 years

5. Rate your programming skills in Java. A number from 1 (low) to 5 (high) *

○ 1 ○ 2 ○ 3 ○ 4 ○ 5

2. Task 1 (on Math4J):

a) Your TARGET class is: org.magee.math.Rational.java

b) Your GOAL is test the class as thoroughly as possible with the help of test case generated by EvoSuite (org.magee.math.TestRational.java).

c) TIMING: You have 45 (maximum) minutes time. You are finished earlier if you a) have written (if needed) test cases to cover all the code, and b) are sure there are no further bugs.

PLEASE use a chronometer to write down the time that was required by you to perform the task.

NOTE: Before performing the task (and start the count down) you can have a brief look at the packages of the system to be more familiar with the source code.

*

ADDITIONAL INFORMATION on Task1:

Notes on EvoSuite and jUnit

- We use EvoSuite to generate the initial test suite: "TestRational.java". We placed the test suite in: "org.magee.math.TestRational.java"

- These unit tests contain assertions that reflect the current behavior of the class under test. However, since the class under test is faulty (i.e., there are bugs), the generated assertions may be wrong reflecting the incorrect behaviour. Therefore, the assertions need to be analyzed and fixed.

- Feel free to remove assertions or to @Ignore tests you do not understand.

- Use jUnit in Eclipse to run the test cases.

Notes on Testing

There are bugs.

A test revealing a bug should fail.

Tests not revealing bugs should pass.

Notes on the Experiment

There are bugs. A test revealing a bug should fail. Tests not revealing bugs should pass. You may fix obvious bugs, but do not waste time debugging.

Notes on the Experiment

Please do not install additional plugins in Eclipse. Please do not change the build process. Please no navigate on-line during the task.

When you're finished write down the time that was required by you to perform the task.

*

NOW YOU CAN START THE TASK...

6.Do have experience with the project from which this class belong to? *

- Yes
- No

3. Remember to write down the time that was required to perform the task.

Please, remember to write down the time that was required to perform the task and report it in minutes and seconds (e.g. "25 minutes and 45 seconds").

7.Report the time required to perform the task in minutes and seconds (e.g. "25 minutes and 45 seconds"): *

8.

How do you judge the usefulness and comprehensibility of the provided test cases? A number from 1 (low) to 5 (high)

*

Very Low | Low | Medium | High | Very High

9.Why? *

4. Task 2 (on Apache Commons Primitives):

a) Your TARGET class is: org.apache.commons.collections.primitives.ArrayIntList.java

b) Your GOAL is test the class as thoroughly as possible with the help of test case generated by EvoSuite (org.apache.commons.collections.primitives.ArrayIntList.java).

c) TIMING: You have 45 (maximum) minutes time. You are finished earlier if you a) have written (if needed) test cases to cover all the code, and b) are sure there are no further bugs.

PLEASE use a chronometer to write down the time that was required by you to perform the task.

NOTE: Before performing the task (and start the count down) you can have a brief look at the packages of the system to be more familiar with the source code.

*

ADDITIONAL INFORMATION on Task2:

Notes on EvoSuite and jUnit

- We use EvoSuite to generate the initial test suite: "TestArrayIntList.java". We placed the test suite in: "org.apache.commons.collections.primitives.TestArrayIntList.java"

- These unit tests contain assertions that reflect the current behavior of the class under test. However, since the class under test is faulty (i.e., there are bugs), the generated assertions may be wrong reflecting the incorrect behaviour. Therefore, the assertions need to be analyzed and fixed.

- Feel free to remove assertions or to @Ignore tests you do not understand.

- If a generated test case does not compile, just delete it.

- Use jUnit in Eclipse to run the test cases.

Notes on Testing

There are bugs.

A test revealing a bug should fail.

Tests not revealing bugs should pass.

Notes on the Experiment

Please do not install additional plugins in Eclipse. Please do not change the build process. Please no navigate on-line during the task.

When you're finished write down the time that was required by you to perform the task.

*

NOW YOU CAN START THE TASK...

10.Do have experience with the project from which this class belong to? *

- Yes
- No

5. Remember to sign the time required to perform the task ONE MORE TIME.

Please, sign the time required to perform the task and report it in minutes and seconds (e.g. "25 minutes and 45 seconds") before to go in the next page.

You finished!

11.Report the time required to perform the task in minutes and seconds (e.g. "25 minutes and 45 seconds"): *

12.

How do you judge the usefulness and comprehensibility of the provided test cases? A number from 1 (low) to 5 (high)

*

○ Very Low ○ Low ○ Medium ○ High ○ Very High

13. Why? *

6. Thank you very much for your effort.

TO COMPLETE the three tasks, please zip the workspace folder (where also the projects are contained) and send it to the following e-mail addresses:

panichella@ifi.uzh.ch, a.panichella@tudelft.nl

14. In your opinion, which kind of source of information is more useful for the proposed testing tasks? *

The generated jUnit test cases by Evosuite.
Test cases comments (when available).
The source code itself.
The test cases I manually wrote.
Others

15. Why? *

16.

Please rank the following in order of importance from 1 to 5 where 1 is most important to you and 5 is least important to you.

*

	1	2	3	4	5
The generated jUnit test cases by Evosuite.	○	○	○	○	○
Test cases comments (when available).	○	○	○	○	○
The source code itself.	○	○	○	○	○
The test cases I manually wrote.	○	○	○	○	○
Others	○	○	○	○	○

17. Please justify the ranking that you provided. *

18.

For each of the following questions about test generation, please specify whether you fully agree, partially agree, partially disagree, or fully disagree.

*

	Fully Disagree	Partially Disagree	Partially Agree	Fully agree
I had enough time to finish my task.	○	○	○	○
It was easy to test the given class.	○	○	○	○

Adding/changing the generated tests leads to better tests.	○	○	○	○
I am certain I have found all bugs.	○	○	○	○
The class under test was easy to understand.	○	○	○	○

19.

For each of the following questions about test generation, please specify whether you fully agree, partially agree, partially disagree, or fully disagree.

*

	Fully Disagree	Partially Disagree	Partially Agree	Fully agree
Without comments, generated unit tests are difficult to read and understand	○	○	○	○
Generated unit tests are typically too long to understand.	○	○	○	○
Automatically generated unit tests only exercise the "easy" parts of the program.	○	○	○	○
Adding assertions to generated unit tests WITH comments is prohibitively difficult.	○	○	○	○
Adding assertions to generated unit tests	○	○	○	○

WITHOUT comments is prohibitively difficult.				
Difficulty in understanding generated unit tests depends on the complexity of the tested class, not the actual tests.	○	○	○	○

20.

Content adequacy

Considering only the content of the comments of jUnit test cases (when they were available) and not the way it is presented, do you think that the description?

*

- ○ Is not missing any information.
- ○ Is missing some information but the missing information is not necessary to understand the class.
- ○ Is missing some very important information that can hinder the understanding of the class.

21.If are missed some important information can you specify which information are missed:

22.

Conciseness

Considering only the content of the comments in the jUnit test cases (when they are available) and not the way it is presented, do you think that the description?

*

- ○ Has no unnecessary information
- ○ Has some unnecessary information
- ○ Has a lot of unnecessary information

23.

If the jUnit test cases comments have some unnecessary information can you specify which information

is useless:

24.

Expressiveness

Considering only the way the comments of jUnit test cases (when they were available) are presented and not its content, do you think that the description?

*

- ○ Is easy to read and understand
- ○ Is somewhat readable and understandable
- ○ Is hard to read and understand

25. Do you have any suggestions on improving the readability of generated unit test cases? *

26. Do you have any suggestions on making unit test cases more understandable?

27. If you have any comment, please write in the following text box: