

# Test Suite Reusability Measurement Based on Uses Frequently and Coverage of Reused Test Cases

Mochamad Chandra Saputra\*, Tetsuro Katayama\*, Yoshihiro Kita†, Hisaaki Yamaba\*, Kentaro Aburada\*, Naonobu Okazaki\*

*\*Interdisciplinary Graduate School of Agriculture and Engineering, University of Miyazaki,  
1-1 Gakuen-Kibanadai Nishi, Miyazaki, 889-2192 Japan*

*†Department of Information Security, Faculty of Information Systems, Siebold Campus, University of Nagasaki,  
1-1-1 Manabino, Nagayo-Cho, Nishi-Sonogi-gun, Nagasaki, 851-2195, Japan  
Email: chandra@earth.cs.miyazaki-u.ac.jp, kat@cs.miyazaki-u.ac.jp, kita@sun.ac.jp,  
yamaba@cs.miyazaki-u.ac.jp, aburada@cs.miyazaki-u.ac.jp, oka@cs.miyazaki-u.ac.jp*

## Abstract

Reusability of a test suite is one of important factors to decrease the cost in software testing. This research defines and measures the test suite reusability score expressed with the degree of reusability of the test suite. It is calculated by considering both frequency and code coverage of successful reused test cases in the test suite. Test suite reusability measurement provides useful information to improve the efficiency of software testing, especially in regression testing and automated testing.

*Keywords:* Test suite reusability, reused frequency, code coverage, reused test cases,

## 1. Introduction

White box testing knows as structural testing is one of the software testing approaches to test the program based on the internal structures of the program<sup>1</sup>. It is used to analyze internal logic and behavior, data structure, and code coverage. Test cases in the test suite is an important asset for software testing. The test cases are important on white box testing method to guarantee all independent paths or statements within a program executed at least once. Test cases should execute all true or false logical decisions and all loops at their or within the boundaries and ensure the validity of internal data structures<sup>2</sup>.

Currently, reusing assets in the software development process have a great purpose. The background of reusing assets in software development is to reduce cost and effort<sup>3</sup>. A reusable test suite at the testing phase is decreasing testing time and cost. Generally, the test suite effectiveness is depended on the capability of the test cases on error detection related to coverage of the test cases on the program.

The research on test cases reusability measurement considers to code coverage is reported by modifying the program with reducing branches, the test suite is highly reusable but, modifying the program such as splitting, and loop, the test suite is less reusable<sup>4</sup>. The research has shown test suite reusability measurement is needed to consider the code coverage.

The use of code clones is increased day by day due to the growth of the use of open-source software and variants<sup>5</sup>. Code clones are efficient in reducing the cost and time on software development that have similar requirements. Testing code clones is needed a strategy to achieve efficiency on it. Test suite reusability is one of the best strategies to test the code clones.

This research proposes the formula for the test suite reusability score to measure the degree of test suite reusability based on reused test cases on another program. The formula is considered to test suite reusability frequency and distinct of code coverage. The test suite reusability score measurement is including good

information to evaluate the efficiency of the test suite on reusability.

## 2. Principles of Reusability and Code Clones

The principle of software reusability is defined as the capability of an attribute to be reused in various objects<sup>6</sup>. The terminology of test suite reusability in this research is related to software reusability as the capability of test cases in the test suite to examine several or all paths of method should be tested on diverse objects. The test suite reusability additionally needs to consider code coverage on the reusability measurement. The case of test suite reusability measurement on white box testing, the capability of the test suite to examine different objects is not good enough. Consider the test suite can achieve 100% code coverage are one of criteria of a good test suite.

This research uses several types of code clone<sup>7,8</sup> such as code clone type 1 (exact clones) are identical clones except in white space and comments. The second type is code clone type 2, the differences from the original code are renamed identifiers, literals, types, layout, and comments but the structurally and syntactically is similar. The code clones type 3 are modified the statement such as statement insertions/deletions in addition to changes in identifiers, literals, types, and layouts. Code clone type 4 has been modified on code fragments to perform the same objective but different syntactic variants. The test suite reusability measurement is used on code clones because the test cases in the test suite need to use the same characters and by using the code clones we can have an identical or near identical program for uses of the test suite<sup>9</sup>.

## 3. Test Suite Reusability Measurement

Reusing assets on software testing will help to reduce the cost of software testing. The asset is defined as any kind of product from any part of the software process<sup>10</sup>. The test suite is not very valuable if is not reusable in testing another program. The test suite reusability measurement is important to know how reusable the test suite.

The test suite reusability measurement is conducted by investigating the test case execution on the Java programs. First information used on this measurement is the number of successful reused test cases in the test suite. The successful test cases are the examination result of the test cases on Java program that could achieve the objective of the testing without any error. The next information is distinct code coverage. The test cases in the test suite are possible to execute similar lines of code.

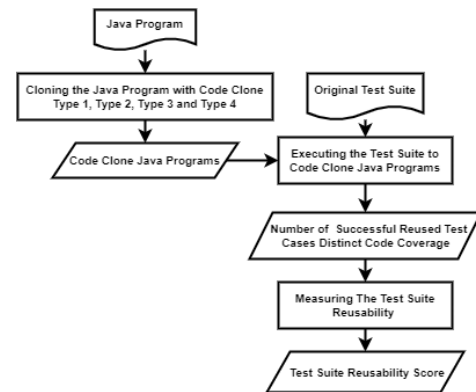


Figure 1. Test Suite Reusability Measurement Activity

The similar lines of code executed then count as one line of code from redundant lines of code or called distinct code coverage.

To simplify the formula, the research uses the following notation.

- SRTC: Successful reused test cases
- DCC: Distinct code coverage
- OT: Objects tested
- OLOC: Original line of code
- TC: Test Cases

By using the notation, the formula for test suite reusability score as follows.

$$TestSuiteReusabilityScore = \frac{\sum SRTC + \sum DCC}{(\sum OT \times \sum TC) + \sum OLOC} \quad (1)$$

The proposed formula for test suite reusability score has considering frequency and code coverage of successful reused test cases in the test suite.

## 4. Methodology

The test suite reusability measurement in this research is used on white box testing approach considered to the number of reused test cases in test suite to test the programs and code coverage. The test suite reusability measurement activity is shown in Figure 1. The activity of this experiment is using a Java program and given test suite to measure its reusability. Test suite reusability measurement based on code clones approach is used the number of successful reused test cases, and distinct code coverage from the test suite on calculation. The research uses code clones type 1, type 2, type 3, and type 4. Test suite reusability measurement results are informing the degree of test suite reusability.

## 5. Experiment

The research uses the *parallelogram* Java program as shown in Figure 2 with two given test suites. On the original *parallelogram* Java program, the test suites are

```

package parallelogram;
import java.io.BufferedReader;
import java.io.InputStreamReader;
public class Parallelogram {
    public static void main(String[] args) {
        double base,height;
        BufferedReader br = new BufferedReader(new
        InputStreamReader(System.in));
        System.out.println("This program calculate value the
        area of a trapezium");
        System.out.println("Enter the base and the height of
        the parallelogram");
        try{
            base=Double.parseDouble(br.readLine());
            height=Double.parseDouble(br.readLine());
        }catch (Exception e){
            System.out.println("An error occurred");
            return;
        }
        if(base<=0 || height<=0){
            System.out.println("Wrong Input");
            return;}
        System.out.println("Area = " + base*height );
    }
}

```

Figure 2. Parallelogram Source Code

examining for collecting the coverage information. The test suite reusability measurement on this experiment is excluding the result of original test suite examination data. The experiment is focused on the test suite reusability score on another Java program that reuses the test suite to testing the program. The original Java program is cloning by code clone type 1, type 2, type 3, and type 4 approach.

This research has two given test suites. The first test suite contains three test cases and the second test suite contains four test cases are examined on code clones. The information of successful reused test cases and distinct code coverage from test cases examination on code clones are used for test suite reusability measurement.

## 6. Result and Discussion

The research is examined the given test suites on code clones type 1, type 2, type 3, and type 4 to collect information number of reused test cases in the test suite and code coverage of the test suite. Table 1 shows that the given test cases on the test suite successfully examine the code clones. The distinct code coverage on code clones shows that several lines of code on code clones type 1, 2, and 3 are not executed by test cases on the test suites during the testing.

The result of code coverage information shows in Table 1. The value 1 means this line is executed and 0 is not executed. Table 1 also shows the result of distinct code coverage in another column. The value 1 means this line is executed and 0 is not executed during the testing by using one or more test cases. The result of the number of successful reused test cases shows in Table 2. The result obtains from test cases examination on code clones. Table 3 shows the result of test suite reusability

Table 1. The result from code coverage information on code clones

| No. LOC | Code Clone Type-1 |      |      |                        |              |      |      |      |                        |
|---------|-------------------|------|------|------------------------|--------------|------|------|------|------------------------|
|         | Test Suite-1      |      |      |                        | Test Suite-2 |      |      |      |                        |
|         | TC-1              | TC-2 | TC-3 | Distinct Code Coverage | TC-1         | TC-2 | TC-3 | TC-4 | Distinct Code Coverage |
| 1       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 2       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 3       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 4       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 5       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 6       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 7       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 8       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 9       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 10      | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 11      | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 12      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 13      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 14      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |
| 15      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |

| No. LOC | Code Clone Type-2 |      |      |                        |              |      |      |      |                        |
|---------|-------------------|------|------|------------------------|--------------|------|------|------|------------------------|
|         | Test Suite-1      |      |      |                        | Test Suite-2 |      |      |      |                        |
|         | TC-1              | TC-2 | TC-3 | Distinct Code Coverage | TC-1         | TC-2 | TC-3 | TC-4 | Distinct Code Coverage |
| 1       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 2       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 3       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 4       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 5       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 6       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 7       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 8       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 9       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 10      | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 11      | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 12      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 13      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 14      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |
| 15      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |

| No. LOC | Code Clone Type-3 |      |      |                        |              |      |      |      |                        |
|---------|-------------------|------|------|------------------------|--------------|------|------|------|------------------------|
|         | Test Suite-1      |      |      |                        | Test Suite-2 |      |      |      |                        |
|         | TC-1              | TC-2 | TC-3 | Distinct Code Coverage | TC-1         | TC-2 | TC-3 | TC-4 | Distinct Code Coverage |
| 1       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 2       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 3       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 4       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 5       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 6       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 7       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 8       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 9       | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 10      | 0                 | 0    | 1    | 1                      | 0            | 0    | 0    | 0    | 0                      |
| 11      | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 12      | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 13      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 14      | 0                 | 1    | 0    | 1                      | 1            | 1    | 1    | 0    | 1                      |
| 15      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |
| 16      | 1                 | 0    | 0    | 1                      | 0            | 0    | 0    | 1    | 1                      |

| No. LOC | Code Clone Type-4 |      |      |                        |              |      |      |      |                        |
|---------|-------------------|------|------|------------------------|--------------|------|------|------|------------------------|
|         | Test Suite-1      |      |      |                        | Test Suite-2 |      |      |      |                        |
|         | TC-1              | TC-2 | TC-3 | Distinct Code Coverage | TC-1         | TC-2 | TC-3 | TC-4 | Distinct Code Coverage |
| 1       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 2       | 1                 | 1    | 1    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 3       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 4       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 5       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 6       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |
| 7       | 1                 | 1    | 0    | 1                      | 1            | 1    | 1    | 1    | 1                      |

measurement uses the formula(1). The test suite reusability measurement is very common information to reduce the cost of software testing. The different number of lines of code is possible for code clones type 3 and 4, for code clone type 1 and 2 usually the number lines of

Table 2. Result of the number of successful reused test cases

|              | Code Clones | Number of Test Cases | Number of Successful Test Cases Tested |
|--------------|-------------|----------------------|--|
| Test Suite-1 | Type-1      | 3                    | 3                                      |
|              | Type-2      |                      | 3                                      |
|              | Type-3      |                      | 3                                      |
|              | Type-4      |                      | 3                                      |
| Test Suite-2 | Type-1      | 4                    | 4                                      |
|              | Type-2      |                      | 4                                      |
|              | Type-3      |                      | 4                                      |
|              | Type-4      |                      | 4                                      |

Table 3. Result of test suite reusability measurement

|              | Code Clones | Test Suite Reusability Score | Average |
|--------------|-------------|------------------------------|---------|
| Test Suite-1 | Type-1      | 100%                         | 100%    |
|              | Type-2      | 100%                         |         |
|              | Type-3      | 100%                         |         |
|              | Type-4      | 100%                         |         |
| Test Suite-2 | Type-1      | 84%                          | 87%     |
|              | Type-2      | 84%                          |         |
|              | Type-3      | 80%                          |         |
|              | Type-4      | 100%                         |         |

code are the same. The distinct code coverage is used to selecting the redundant line of code executed by the test cases on the test suite. By using the distinct code coverage, the calculation of percentage code coverage is represented the actual condition. The test cases in the test suite are possible to execute similar lines of code. The number of successful reused test cases on the test suite as shown in Table 2 is important for test suite reusability measurement. The number of successfully reused test cases is one of the important parameters for the test suite reusability measurement.

The information of code coverages and number of successful reused test cases in the test suite is used for test suite reusability measurement. The result of test suite reusability measurement uses the formula (1) shown in Table 3. The result of test suite reusability score shows the different scores for several test suites because they have the different number distinct code coverage by the test cases as shown in Table 1. The number of distinct code coverage is important to ensure the capability of the test suite to achieve 100% code coverage. The score of test suite reusability 100% means the test suite has the perfect capability to achieve 100% code coverage and reused on another program. The test suite reusability measurement provides useful information to reduce the cost of software testing, especially in regression testing and automated testing.

## 7. Conclusion

This research confirms the test suite reusability measurement is calculated by combining the number of successful reused test cases and code coverage. The

result shows the number of successful reused test cases in the test suite did not guarantee 100% code coverage on the test suite reusability measurement. The result of test suite reusability measurement is valuable information to reduce the cost of software testing, especially in regression testing and automated testing.

Our future works will focus on other measurement for test suite quality measurement.

## References

1. Sommerville, I. *Software Engineering*. Addison-Wesley Publishing Company, 2010, pp. 133–170.
2. Roger S. Pressman, P. D. *Software Engineering: A Practitioner's Approach*. McGraw-Hill, 2009.
3. Sutcliffe, A. *The Domain Theory Patterns for Knowledge and Software Reuse*. Lawrence Erlbaum Associates, Inc. Publishers, 2002.
4. Dong, Y., Wang, Y., Lau, M. F. & Lin, S. *Experiments on Test Case Reuse of Test Coverage Criteria*. 7th International Conference on Ubiquitous Intelligence & Computing and 7th International Conference on Autonomic & Trusted Computing, 2010, pp. 277–281.
5. Saini, N., Singh, S. & Suman. *Code Clones: Detection and Management*. Procedia Computer Science, 2018, pp. 718–727.
6. Younoussi, S. & Roudies, O. *All About Software Reusability: A Systematic Literature Review*. Journal of Theoretical and Applied Information Technology, 2015, pp. 64–75.
7. Bellon, S., Koschke, R., Antoniol, G., Krinke, J. & Merlo, E. *Comparison and Evaluation of Clone Detection Tools*. IEEE Transactions on Software Engineering, 2007, pp. 577–591.
8. Roy, C. K. & Cordy, J. R. *Survey on Software Clone Detection Research*. in Technical Report No. 2007-541, 2007.
9. Asaithambi, S. P. R. & Jarzabek, S. *Towards Test Case Reuse: A Study of Redundancies in Android Platform Test Libraries*. in Lecture Notes in Computer Science vol. 7925, 2013, pp. 49–64.
10. Ezran, M., Morisio, M. & Tully, C. *Practical Software Reuse*. Springer London, 2002.