# False Test Case Selection : Improvement of Regression Testing Approach

Benjawan Srisura
Information Technology Department
Vincent Mary School of Science and Technology
Assumption University, Thailand
benjawan@scitech.au.edu

Adtha Lawanna
Information Technology Department
Vincent Mary School of Science and Technology
Assumption University, Thailand
adtha@scitech.au.edu

***Abstract*—Regression testing has been considered as a time-consumed process in software testing. In a recent year, one of interesting research work initiated for minimizing testing time is finding a technique in selecting test cases from a large test suit. Most of test cases selection technique in literature considers test cases that are related to the requirement's changed. During executing test cases that are related to the modified part, a set of fail test case is accidentally emerged and make test suit has become larger. Therefore, this paper proposes a technique in selecting suitable false test cases when they are generated in regression testing. However, in order to ensure that the quality and validity of using the proposed technique are acceptable, an experiment was systematically conducted in this study. And we also found that the false test case selection technique can minimize the size of test suit, effectively. (*Abstract*)**

***Keywords—Test case selection; Regression Test Case Selection; Regression Testing; Sotware Testing (key words)***

## I. Introduction

One of the important phases in the software development systems is software testing [1]. It evaluates the software's capability and reveals errors before delivering the desired software to users. Generally, regression testing is a repeating process which is executed in software testing to confirm the requirements have been implement correctly and also ensure that any applied bug's fix or a change of code have not affected previous functionalities and introduced any errors.

Generally, regression testing approach can be separated into three main processes - test case minimization, test case selection and test case prioritization. Test case prioritization process considers ordering of test cases for detection of faults at the earliest while test case minimization is carried out to eliminate the redundant test cases. Finally, test case selection process, its aim is to be able to select the subset of test cases from the test suite that has the potential to detect errors initiated from a change.

For a large scale software product, there are many numbers of test cases available in its test suite. Whenever a new version of software is released, re-running all test cases is executed with a potentially cost, time and effort [2, 3]. Therefore, a need of selecting a suitable subset of test cases from a large test suite has become an interesting issue to be improved.

Interestingly, considering the control flow of program module in selecting appropriated test cases has been concerned as high safe and precise techniques in the literature [4, 5, 6, 7, 8]. Unfortunately, in every test case execution, fail test cases set have been generated and then they will be totally delivered into the test suit for retesting next cycle, automatically. This situation may produce a test suit whose size is constant or become bigger.

As we consider the various types of delivered false test cases, each of them has its own level of defecting – such as some are effects from another, some are critical error. However, this information has never been recognized for improving consecutive test case selection. If considering false test case can be used to be another criterion in selecting the suitable test cases, test suit size may be minimized, significantly. Therefore, this paper proposes a false test case selection technique for supporting regression test selection in considering suitable test cases for the next execution.

## II. Related Works

According to the previous studies, this paper identifies pattern of regression test selection (RTS) techniques into two main approaches. The first approach, selecting test case with non-criteria – i.e. experts' judgement which tends to become unreliable with various type of software product, retest-all (AS) technique whose performance is quit poor due to the high cost of executing entire test cases, random selection technique that is ineffective in a large software scale.

Secondly, at least one criterion is carried out to be considered for selecting suitable test cases. RTS techniques of this group can be categorized based on how to identify test cases that affect to the modifications' part – including Dataflow technique which analyzes the data flow or data path of program's module [9, 10, 11] which is unsafe and imprecise because it does not consider the control dependency among program element, Slicing techniques [12, 13] selects test cases that generate different outputs which is precise because it does not involve test cases which do not produce different outputs, Firewall techniques [14, 15] select only test cases execute the modified module which may unsafe and imprecise, Differencing techniques (such as detecting code or textual difference) [16, 17] whose precision may poor, and Control flow technique [8] analyze control flow of modified module to

**978-1-4673-9749-0/16/$31.00 ©2016 IEEE**