

Test Path Optimization Algorithm Compared with GA Based Approach

Aye Aye Kyaw and Myat Myat Min

Abstract Software testing performs a vital and basic role in developing software that is freedom from bugs and defects. Various levels of testing use Unified modeling language (UML) to design the tests. The test case generation from design specification is an essential work in testing phase. In recent trend, Model-Based testing approach attracts many researchers by using some search optimization techniques to produce an automated optimal test path. The best test path is got more beneficial than the other generated test paths because it can reduce time and cost more than others. In this paper, we compare and analysis different methodologies of model based best test path generation by using the UML activity diagram. We discuss the results the resulting test paths with respect to correctness and time consuming.

Keywords Model-Based testing · Test path · Test path generation · UML activity diagram

1 Introduction

A software project is made up of a series of phases. Generally, software projects comprise the phases as: i) Requirements gathering and analysis, ii) Design, iii) Development or Coding, iv) Testing, and vi) Deployment and Maintenance. Each and every phase has its own functionalities, milestones and deliverables to be achieved, in order to ensure a product that meet the needs of the customer.

This figure shows how defects from early phase add to the cost of the software project [1]. This implies that defects, when introduced in the requirements phase that might occur due to the misunderstanding or inconsistent requirements may inquire additional costs in detecting and correcting those errors. The defect can have continuously from requirement phase to testing phase. If the programmer knows having the defect at the testing phase, it raises the cost more and more at

A.A. Kyaw(✉) · M.M. Min
University of Computer Studies, Mandalay, Myanmar
e-mail: {ayeayekyaw2009,myatiimin}@gmail.com

© Springer International Publishing Switzerland 2016
T.T. Zin et al. (eds.), *Genetic and Evolutionary Computing*,
Advances in Intelligent Systems and Computing 387,
DOI: 10.1007/978-3-319-23204-1_46

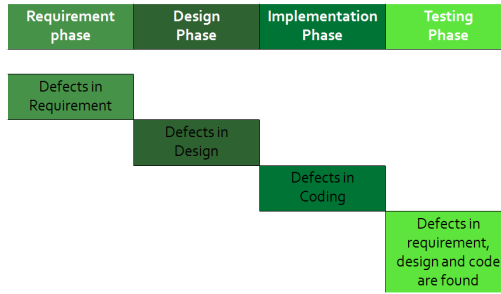


Fig. 1 Defects from early phase adding to the cost

every phase. To find defects early, testing activities shall be started as early as possible in the software or system development life cycle, and shall be focused on defined objectives. Good software testers cannot avoid models. MBT has emerged as a useful and efficient testing method for realizing adequate test coverage of systems.

Model based test case generation approaches identify faults in the implementation at early stage, reduce the software development time, and inspire developer to improve design quality. Unified Modeling Language (UML) has become the de facto standard for object oriented modeling and design. Most of researchers have used the following architecture for generating the test paths. There are various kinds of UML diagrams such as class diagram, activity diagram, sequence diagram, object diagram, and others. One or more UML diagram are used as an input, construct the dependency table based on these diagrams, and then create dependency graph according to the dependency table. Finally, produce all possible test paths. Some use data mining algorithms such as GA, Tabu Search, Ant Colony Optimization algorithm to find the best test path [2].

The rest of the paper is organized as follows: the related work is described in section 2. Test path optimization algorithm and GA-based algorithm to optimize the best test path are described in Section 3 and 4 respectively. Section 5 describes the experimental evaluation results with the different activity diagrams of the different domains. The paper is concluded with section 6.

2 Related Work

Many researchers have been successfully proposed test case generation for various software under many circumstances, using mainly search optimization techniques such as Genetic Algorithm, Ant Colony Optimization Algorithm, Tabu Search Algorithm, etc. The following papers are based work evolved under model based test case generation.

In [3], Shantghi and Mohan Kumar proposed an approach for generating test cases for object oriented software that uses Tabu Search technique to design and derive test cases. The Tabu search technique was applied to generate optimal test case.

Model Based Test Case Generation Technique Using Genetic Algorithms presented by M Sumalatha and Raju based on UML activity diagram. Finding all paths