MIS Review Vol. 15, No. 2, March (2010), pp. 75-90

Non Homogenous Poisson Process Model for Optimal Software Testing Using Fault Tolerance

Praveen Ranjan Srivastava, Chetan Mittal, Ajay Rungta, Vaibhav Malhotra, G Raghurama
Birla Institute of Technology and Science, Pilani, Rajasthan, India

ABSTRACT: *In software industry, it is important to prioritize the different modules of a software so that important modules are tested ahead of the lesser important ones. This approach is desirable because it is not possible to test each module regressively due to time and cost constraints. This paper proposes a way to prioritize several modules of a software product and calculates optimal time and cost for testing based on non homogenous poisson process. Sometimes it is more profitable for an organization to release software, even if it is not completely tested because of limited time and resources. This paper also tries to figure out whether the software could be released or not, after testing within a given time and cost.*

KEYWORDS: *Non Homogenous Poisson Process, Optimal Test Policy, Software Life Cycle Length, Testing Time, Module Test Prioritization, Fault Tolerance.*

1. Introduction

The essence of software testing is to find out any faults that might exist before releasing the product in the market. For this purpose, software product is tested carefully. The very primitive method of testing software is regression testing [15]. It is the process of testing software to make sure that old program still works with the new changes. Regression testing is any type of software testing which seeks to uncover software regressions. Such regressions occur whenever software functionality that was previously working correctly, stops working as intended. Typically regressions occur as an unintended consequence of program changes. Common methods of regression testing include re-running previously run tests and checking whether previously fixed faults have re-emerged [15]. But it is not feasible to perform regression testing on the software always, as it can be very expensive. In fact a large portion of the software maintenance budget can be consumed by regression testing [1]. That's why, a tester should find out what are the modules with greater importance so that they can be tested first and given more effort. It is impractical to test the software unless all the bugs are removed. The tester should also be aware of the optimal testing time and cost required to test the modules. When it is not possible to remove all the bugs with limited resources, then we have to accept limited faults in the software. For this reason, this paper attempts to provide an optimal boundary values for time and cost considering the actual percentage of