Modelling Fault Detection Using SRGM in Agile Environment and Ranking of Models

Abhishek Srivastava, Amity School of Engineering and Technology, Amity University, New Delhi, India
P. K. Kapur, Amity Centre for Interdisciplinary Research, Amity University, Noida, India
Deepti Mehrotra, Amity School of Engineering and Technology, New Delhi, India
Rana Majumdar, Amity School of Engineering and Technology, New Delhi, India

ABSTRACT

Today’s software applications deployed in an enterprise to cater to the complex business processes, integrate various business units and address requirements of a global customer base. The traditional methodology of software engineering succumbs to the changing need of customer and technology advancement. On the behest of the customer, a software system should be designed in a way that it goes in concert with the present user needs. Agile methodology targets complex systems with its iterative, incremental, and evolutionary approach. There are numerous factors attributing towards the successful implementation of agile methodology. This led to adopting an approach of agile based on ‘lean’ principles over the traditional software development life cycle (SDLC) approach. Collaborative work is done with the project team on a priority list. The implementation is done through “SCRUM” an empirical framework for learning. It has multiple sprints which are deliverable products. This idea has substantially reduced the ‘time to market’ as the customer can decide which features of the software they would like to be delivered on a priority basis. To model trends of fault detection in each sprint, a growth model of software reliability is used. This research article presents a framework to analyze and measure the cumulative errors in an Agile Testing Process, the authors have applied modeling on various SRGMs to prove acceptability in an agile development process and finally compares these models using the Mahalanobis Distance Formula for Model ranking. The Mahalanobis distance criteria is easy to compute and that can be utilized to get the ranks and select the best model in view of an arrangement of contributing criteria.

KEYWORD
Agile Framework, Continuous Integration, Modeling, Release, Scrum, Software Reliability Growth Models (SRGM), Sprints

1. INTRODUCTION

Agile process is considerate of the ability of reacting quickly and effortlessly towards the changes. The Agile process is light, systematic and flexible. Earlier traditional approaches led a software be undergone through many sequential processes of a cycle involved in SDLC. Steps included requirement analysis, defining it properly, chalking a plan, start a build, testing and finally deployment. Over the period, practitioners have identified many pitfalls in the traditional approach of software development T. Addison et al. (2002), Belasis et al. (1996), Hughes et al. (2014). In agile client documentation are required to visualize the architecture. Then starts the implementation of it, followed by different

DOI: 10.4018/JCIT.2019040101

Copyright © 2019, IGI Global. Copying or distributing in print or electronic forms without written permission of IGI Global is prohibited.
types of testing and deploying it finally. On the contrary the agile method of software development is much less tedious than the conventional one.

Sprints are small time slices in which the phases are made to undergo regress testing. A time frame of 2-4 weeks is taken in a sprint. On table meeting is done with the owner of the product (the one who is requesting the needs) and the team developing it. Work is being decided amongst them. The development team finally decides when it comes to realistic evaluation of work to be accomplished during the sprint and the owner of product has to decide finally on what all criterion is required which are to be satisfied for the approval of work and to be given acceptance. The scrum master determines the sprint. Once the conclusion is drawn of what will be the duration of the sprint by the team all sprints of future should be the identical. Once the sprint begins, no interference of the product owner should be there for smooth functioning of the team to do their work. The power resides only in the hands of the scrum master or project manager. Completed work is presented at the end of the sprint to the project owner and the established criteria of acceptance or rejection is used by the project owner to take the decision and to ensure based on the given feedback, testing of modules & business prioritization, before beginning a new sprint. Therefore, it gives a guarantee of proper evaluation & response to any change in requirements of customer (see Figure 1).

Rana et al. (2011,2012,2013) in their work emphasize and addressed issues and challenges related to software development process in Agile environment. They also proposed a new framework for the same. So, it is clearly evident from literature review nowadays software practitioners are more inclined toward Agile than the conventional one due to its simplicity and less complex structure. Interestingly none of them apply SRGM concept during development to mitigate issues and this instills us to study and work on agile in corporation with SRGMs.

In the Past Yamada et al. (2015) has evaluated the forecasting mechanisms for making software products more reliable than ever before through software development via agile methodology. In a nutshell there is incremental and iterative development, where revisiting of the phases of the process are encountered time & again. More preference is given to adaptability according to changing requirements & continuous compatibility tests. Going simple with Agile, it’s a method of describing software solutions particularly practices of web, web sites, mobile applications that concentrates on high class deliverables of live software consistently and frequently, by simultaneously minimizing the overhead of the project leading to increasing business value. While modeling the routine hazard rates, the Weibull distribution may be a starting choice due to its negatively and positively skewed

Figure 1. Scrum process