

FAULT TOLERANT ISSUE FOR AGILE APPROACH

Dr. Brijesh Kumar Bhardwaj

Assistant Professor, Department of MCA, Dr. R. M. L. Avadh University, Ayodhya

Abstract : A fault Tolerant is a structural imperfection in a software system that may lead to ultimately system failure fault detection plays an important role in development of software. Here we are emphasizing on detection of fault in Agile Approach and software complexity after the careful analysis we got a conclusion that the overpowering number of faults creates much complexity in a software system. The given paper will discuss the detail of fault detection along with its prediction and an overview of system design with agile approach which leads to diminishing complexity in a software design.

Keywords: Agile approach, Fault detection and prediction issues, fault tolerant

I. INTRODUCTION

Agile approach is an incremental system of managing design and constructs the activities of industrial, information technology and other areas of business that is fruitful in a new product development plentiful of associations are at this time endeavouring to present this style of progression. A software process development passes through various activities for minimizing and detecting the errors The second another development approach is laden with fault, particularly if the. One way of decreasing these faults to utilize plan rules and metrics to survey the recently made outlines. The given paper consider the fault detection, fault prediction how to use the interface difficulty issues for the assessment of the Agile Approach[1, 11]. Complexity of software is one of the central factors that must be determined by clear procedures or methods of detection of faults that are utilized in software construction, over the last few decades enormous amounts of software systems have been developed. The quality assurance process of preventing mistakes, detection and disorder in a manufactured product which ISO 9000 Defines as a part of quality management [10].

II FAULT DETECTION AND PREDECTION

Fault detection is very important in a software because if a system is problematic it can create hurdles in working, stoppage of work, inefficient system, industrial accidents arise of faults in a software module. The detection of faults in the early hours detect in a software design can considerably improve the security, dependability, and monetary issue of power system. With the rising know-how fault detection problem can be with no trouble solved using an innovative approach which is based on easy accessible measurements, with no the need for luxurious equipment [8]. This paper describe the advantages and the dispute of using the technology of software fault detection problems, and also highlights the research results Fault of a class predicts the likelihood of the nearness of faults in that class. Assessing and keeping the faults early and exactly is the superior approach for declining the testing endeavours[2, 6]. In the event that fault inclined modules are known ahead of time, audit, analysis and testing endeavours can be focused on those modules [3, 9]. Fault detection Criteria have given the assistance to estimating and analysis of software quality, we can save a lot of time if those problems like lessons, inheritance, combination, connection, epitome which are fault detection and prediction activities could be tried for faults. Detection of Fault is an activity which performs to find the fault at construction stage.

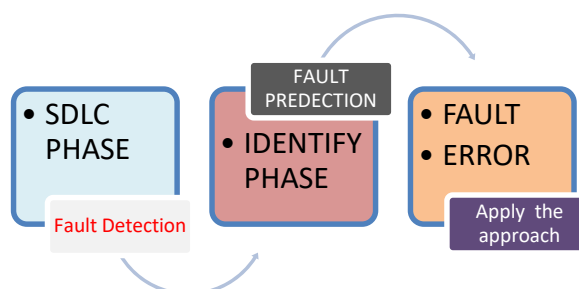


Fig 1 Fault Flow

III FAULT LEVEL

Software fault is one branch of software metrics which is centred on coordinating and estimating the qualities of software, instead of asymmetrical software method, for instance, the venture point of orientation status and announces structure for disappointment[4, 7]. There are a lot of software difficulty method for making the work easy, for example, source lines of code, to the obscure, In a perfect world, complexity method have together expressive and prescriptive segments. Engaging measures distinguish software that is mistake which is not easy to change, very hard to test. Prescriptive measures differentiate operational which help to manage software, for example the division of multifaceted modules into a a small number of less difficult ones, or presentation of the gauge of testing that ought to be perform on given modules [5].

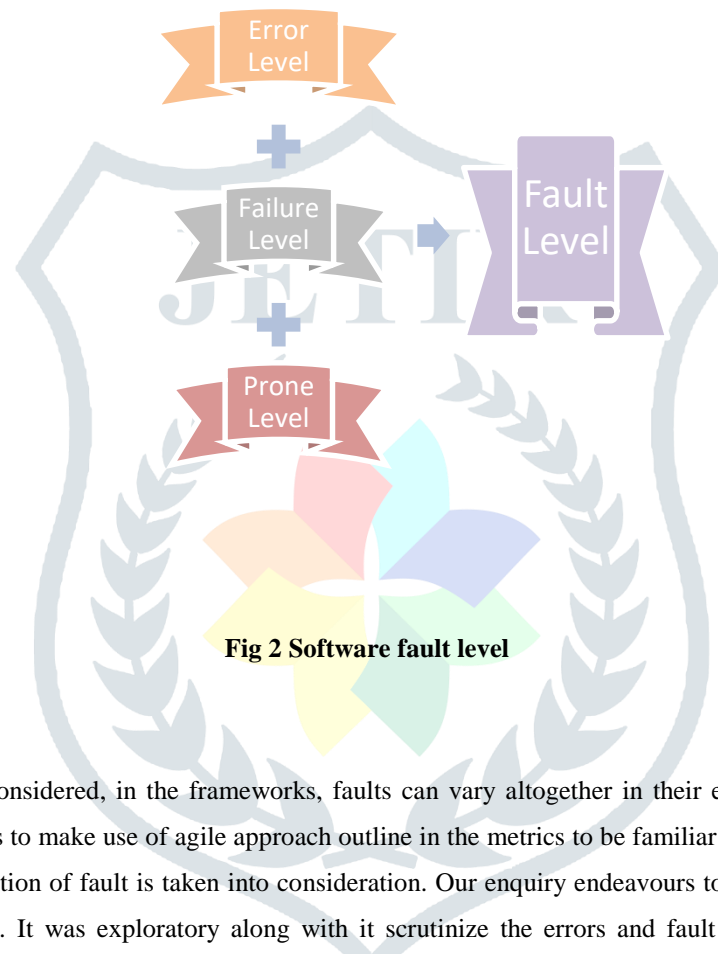


Fig 2 Software fault level

IV. CONCLUSION

All belongings are properly considered, in the frameworks, faults can vary altogether in their effect on the fault levels moves accordingly. It is advantageous to make use of agile approach outline in the metrics to be familiar with the fault-criteria of classes when the significance of detection of fault is taken into consideration. Our enquiry endeavours to provide the data concerning to complexity. In this correction. It was exploratory along with it scrutinize the errors and fault combined with abnormal state standard library classes were less fault inclined, belief and innovatissson than those collective in efforts with software classes.

REFERENCES

- [1]. Shatnawi R., 2012. Improving Software FaultPrediction for Imbalanced Data, IEEE Innovations in Information Technology (IIT), pp: 54-59.
- [2]. Bansiya, Jagdish, and Carl G. Davis. "A hierarchical model for object-oriented design quality assessment." Software Engineering, IEEE Transactions on 28.1 (2002): 4-17
- [3]. G. Booch, "Object-Oriented Design and Application", Benjamin/cummings, Mento Park, CA, 1991.
- [4]. Madeyski, L. And M. Jureczko, 2014. Which process metrics can significantly improve defect prediction models? An empirical study, Software Quality Journal, pp: 1-30
- [5]. Venkata U.B challagula, Farokh B. Bastani, "Emperical Assessment of Machine Learning based Software Defect prediction Techniques ", 10 th IEEE International Workshop on Object Oriented Real Time Dependable Systems, 2008.

- [6]. K. El Emam, S. Benlarbi, N. Goel, and S. N. Rai, "The confounding effect of class size on the validity of object-oriented metrics," *IEEE Transactions on Software Engineering*, vol. 27, no. 7, pp. 630–650, 2001.
- [7]. Basili V.R., (1980), *Qualitative Software Complexity Models: A Summary in Tutorial on Models and Methods for Software Management and Engineering*, Los Alamitos, Calif.: IEEE Computer Society Press.
- [8]. Cardoso, 2006, *Complexity Analysis of BPEL Web Processes*, *Improvement and Practice Journal*, John Wiley and Sons.
- [9]. Chis, (2008), *Evolutionary Decision Trees and Software Metrics for Module Defects Identification*, *World academy of science, engineering and technology* 28, 273-277.
- [10]. Anshul Mishra, D. Agarwal and M. H. Khan, "Availability Estimation Model: Fault Perspective", *International Journal of Innovative Research in Science, Engineering and Technology*, Vol. 6, Issue 6, June 2017.
- [11]. Dr. Brijesh Kumar Bhardwaj, "Examine Relationship Between Fault Factors And Agile Approach", *International Journal of Technical Innovation in Modern Engineering & Science (IJTIMES)*, Vol.5, Issue 01, Jan2019.

