Structural Testing Quality Based Estimation Approach Using Aspect Oriented Programming

[1]Annu, [2] Hema Rauthan
[1]Student, [2] Assistant Professor
[1]Deptt. Of Computer Science
[1] Indo Global college of engineering, Abhipur,Mohali.India

***Abstract* - It is an emerging technique used in most of business applications. Aspect Oriented Programming aims to ease maintenance and promotes reuse of software component. It increase maintainability and reusability of the system. Such code only represented in the modular form but also uses some libraries and external codes. Number of modules and external component access in a code affects the quality of a program code. We have collected some software metrics to take the relative decision. In this work the concern is given to main metrics called coupling, cohesion, module interference Metric. By assigning the weightage to these metrics an overall system metrics is defined. The work is in the form of a software tool that will perform the analysis on java program but the presented metrics are implementable for all Aspect Oriented languages**

***Keywords* - Aspect Oriented, Complexity analysis, Software Quality**

I. INTRODUCTION

Measurement allows the acquisition of information that can be used for developing theories and models, and devising, assessing, and using methods and techniques. Software measurement is a way to track the process. Software engineering is a young discipline, so its theories, methods, models and techniques still need to be fully developed and assessed.

II. SOFTWARE MEASUREMENT

Measurement is introduced by information technology organizations to better understand, evaluate, control and predict software processes. Measurement as the process by which numbers or symbols are assigned to attributes of entities in the real world in such a way as to describe them according to clearly defined rules [6]. Measurement can be categorized in two ways: - (a) Direct measures: It includes software process (e.g., cost and effort applied) and product (e.g., lines of code (LOC) produced, execution speed, and defects rate), and (b) Indirect measures: It includes the product functionality, complexity, efficiency, reliability, maintainability and many others [4] [5]. The goal of software measurement is certainly not limited to deriving measures. This body of knowledge can be used to understand, monitor, control, and improve software processes and products. Therefore, building measures is a necessary part of measurement, but not its final goal.

Software quality analysis is one of the key concerns for all software stakeholders under different aspects of software estimation. One of such aspect is the software reliability. Software testing is one of the effective approaches to estimate the software reliability or the software quality. Aspect Oriented programming generates the software modules under new perspective called aspects. Aspect basically defines the function or the event of the software system that itself collect number of attributes and functions incorporated to the individual entity. The structural analysis mechanism performed to analyze individual component of the software system. Once the individual module analysis is performed, at second stage, the module interaction analysis will be performed. This analysis includes the interaction of a module with other as well as the interfacing with the external environment will be analyzed. In the final stage, the collaborative analysis of all these modules will be performed to perform the system integration analysis. In this section, complete software quality will be analyzed. The analysis done on real time software application that will include the statistical analysis as well as metrics oriented analysis. The work is implemented on reaL time aspectJ project. The analysis is based on the structural and the work flow based analysis

III. ASPECT ORIENTED PROGRAMMING

It is an emerging technique that has profound impact in the area of software development. AOP aims to ease maintenance and promotes reuse of software components by providing mechanism for implementing cross-cutting concerns. Examples of cross-cutting concerns are readability, security etc.

Aspect-Oriented Software Development (AOSD) is a new paradigm which supports separations of concerns (SOC) in software development.. Researchers have explored many methodologies in order to assess the reusability of Object-Oriented (OO) software systems but the assessment of reusability of Aspect-oriented (AO) is under explored.AOSD aims to modularize crosscutting concerns in an application, which cannot be modularized otherwise using traditional approaches. By applying an aspect-oriented approach, concerns like security, readability can be isolated resulting in the increased maintainability and reusability of the system.

IV. RESULT

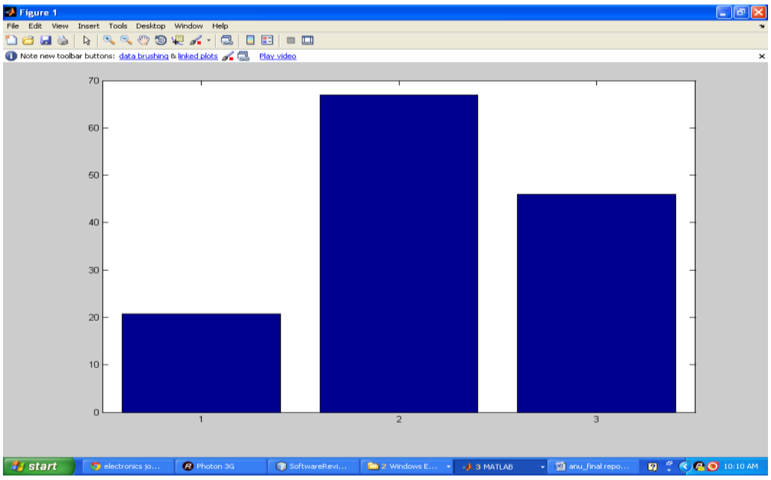

Fig 1 Complexity Analysis

FIGURE 1 showing the complexity analysis of an aspect oriented program under different module respective to components. The complexity is here analyzed in terms of attribute based analysis, method based analysis and nested component analysis

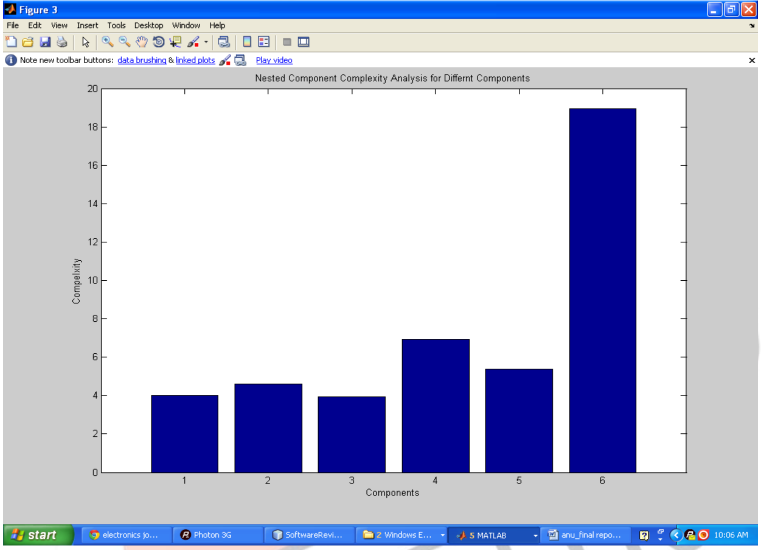

Fig 2 Nested Component Based Complexity Analysis

Here figure 2 is showing the nested component based complexity analysis of an aspect oriented program under different module respective to components. The complexity is here analyzed in terms of nested component based analysis. The figures shows the component is more complex in terms of nested component level complexity

V. CONCLUSION

The proposed work is about to estimate the software quality for aspect oriented program. The work is defined to analyze the software system in terms of integrated component analysis. The complexity analysis shows the effective analysis of software system.

VI. REFERENCES

[1] Lalji Prasad, and Aditi Nagar,” Experimental Analysis of Different Metrics (Object-Oriented and Structural) Of Software”,2009 First International Conference on Computational Intelligence, Communication Systems and Networks 978-0-7695-3743-6/09© 2009 IEEE.

[2] Zeeshan Ali Rana, Shafay Shamail, and Mian Muhammad Awais,” Ineffectiveness of Use of Software Science Metrics as Predictors of Defects in Object Oriented Software”, World Congress on Software Engineering, IEEE, 2009.

[3] Raed Shatnawi,” A Quantitative Investigation of the Acceptable Risk Levels of Object-Oriented Metrics in Open-Source Systems”, IEEE Transactions on Software Engineering 0098-5589/10@ 2010 IEEE.

[4] V. Krishnapriya, and Dr. K. Ramar,” Exploring the Difference between Object Oriented Class Inheritance and Interfaces Using Coupling Measures”,2010 International Conference on Advances in Computer Engineering 978-0-7695-4058-0/10© 2010 IEEE.

[5] Mr. U. L. Kulkarni,” Validation of CK metrics for Object Oriented Design Measurement”, Third International Conference on Emerging Trends in Engineering and Technology 978-0-7695-4246-1/10© 2010 IEEE.

[6] Qingfeng Du,” Software Power: A New Approach to Software Complexity Metrics”, 2010 Second WRI World Congress on Software Engineering 978-0-7695-4303-1/10© 2010 IEEE.

[7] Jianguo Chen,” Complexity Metrics for Component-based Software Systems”, International Journal of Digital Content Technology and its Applications.
[8] Dr. M.P. Thapaliyal Garima Verma,” Software Defects and Object Oriented Metrics – An Empirical Analysis”, International Journal of Computer Applications (0975-8887) Volume 9– No.5, November 2010.
[9] E Da-wei and Xiamen, ” The Software Complexity Model And Metrics For Object-Oriented”, 1-4244-1035-5/07@2007 IEEE.
[10] Tieng Wei Koh, Mohd Hasan Selamat, Abdul Azim Abdul Ghani, Rusli Abdullah,” Review of Complexity Metrics for Object Oriented Software Products”, IJCSNS International Journal of Computer Science and Network Security, VO 314 L.8 No.11, November 2008**.**