Volume 7, Issue 1, January 2017 ISSN: 2277 128X

International Journal of Advanced Research in Computer Science and Software Engineering

Research Paper

Available online at: www.ijarcsse.com

Software Quality Metrics in Quality Assurance to Study the Impact of External Factors Related to Time

T John Vijay[*1], Dr. M. Gopi Chand[2], Dr. Harika Done[3]

[1]Research Scholar, JNTU, Hyderabad, Telangana, India

[2]Professor, VCE, Hyderabad, Telangana, India

[3]Asst. Professor, SVEC, Tadepalligudem, Andhra Pradesh, India

DOI: 10.23956/ijarcsse/V6I11/0114

Abstract: Software quality assurance is a process for guesstimating and documenting the quality of the software products during each phase of the software development lifecycle. Measurement provides a way to assess the status of any program to determine if it is in trouble or in need of corrective action and process of improvement. This assessment helps to diagnose the status in terms of time, completion, risks and also the quality and thus ensures to take meaningful, effective remedial action (e.g., controlling requirements changes, improving response time to the developer, relaxing performance requirements, extending your schedule, adding money, or any number of options). Successful software quality assurance is highly dependent on software metrics. The purpose of this paper is to give a clear picture about the application of Software Quality Metrics (SQM) in quality life cycle and to focus on the significance of SQM in the role of external factors, mainly concentrating on time, which will influence the quality of the final product and plays a vital role in the software quality.

Keywords: Software quality assurance, Software Quality Metrics (SQM), Software quality, Time.

I. INTRODUCTION

Software quality assurance is a formal process for evaluating and documenting the quality of the work products during each stage of the software development lifecycle. The practice of applying software metrics to operational factors and to maintain factors is a complex task. Successful software quality assurance is highly dependent on software metrics. It needs linkage the software quality model and software metrics through qualityfactors in order to offer measure method for software quality assurance [1]. Software quality assurance (SQA) is a technique to help achieve quality and is becoming a critical issue in software development and maintenance [2]. Software metric deals with the measurement of software product and software product development processand it guides and evaluates software development [3]. It is an essential research subject in software engineering [4] & [5]. Test Metrics are useful to take the decision for next phase of activities such as, estimate the cost &amp; schedule of future projects, understand the kind of improvement required to success the project and also to take decision on process or technology to be modified etc. and thus, Test Metrics are the most important to measure the quality of the software.

II. APPLICATION OF TEST METRICS

If metrics are not followed, then the work completed by the test analyst will be subjective i.e. the test report will not have the proper information to know the status of the work/project. If Metrics are involved in the project, then the exact status of the work with proper numbers/data can be published, i.e. in the Test report, one can publish the details of the project or programme like:

1. How many test cases have been designed per requirement?
2. How many test cases are yet to design?
3. How many test cases are executed?
4. How many test cases are passed/failed/blocked?
5. What is the total estimated time?
6. How far the project has been completed by the day evaluation?
7. How many risks were identified?
8. Whether the project requires any other sources to deliver it with in the stipulated time? etc.

Software metrics are of various types like:Product metrics are measures of the software product at any stage of its development, fromrequirements to installed system. Product metrics may measure the complexity of thesoftware design, the size of the final program, or the number of pages of documentationproduction. Process metrics are measure of the software development process, such asoverall development time, the average level of experience of the programming staff, or typeof methodology used. The test process metrics provide information about preparation fortesting, test execution and test progress. Some test product metrics are number of testcases design, % of test cases execution, or %

test cases failed. Test product metricsprovide information of about the test state and testing status of a software product and aregenerated by execution and code fixes or deferment. Some rest product metrics areEstimated time for testing, average time interval between failures, or time remaining tocomplete the testing.The software maintenance phases the defect arrivals by time interval and customer problemcalls. The following metrics are therefore very important: Fix backlog and backlogmanagement index, fix response time and fix responsiveness, percent delinquent fixes, and fix quality [1].

III. EARLIER WORKS DONE RELATED TO THE PRESENT STUDY

Quality of software is the reason of success and decline of a software related organization. Many researches have been done regarding quality management, measurement and assurance in different countries. Florac (1992) [6] discussed software quality measurement and the related problems and defects. The defects were used in understanding and predicting the software quality product and software efficiency.

Devnani-Chulani (1998) [7] is discussing cost, schedule and quality with the help of two sub-models of quality model. This study discussed initial stages of the model which tells great deal about the foundation of numerical problems related to the model parameters. And different models are discussed related to the measurement parameter.

Iftikhar and Ali (2011) [8] deliberated the role of quality measurement in software industries of Pakistan. It was discussed how quality assurance is measured in different industries and how they are compared. In 2007, Imam et al.,[9] explained that it is difficult to improve industrial software quality standards only by relaying on few approved models.

Stavrinoudis and Xenos (2008) [10] discussed how to measure the internal and external software quality. Quality metrics and surveys were conducted to measure customers' software quality reviews. Mekprasertvit (2004) [11] discussed how software quality assurance plan are made. In this study, different techniques and steps were discussed. Jørgensen (1999) [12],Farooq*et al*. (2011) [13]explained the importance of software quantitative and qualitative metrics. Boegh*et al*. (1999) [14] discussed internal and external quality characteristics and the link between them. Squid data model was discussed. Some specification relating software quality was also discussed.

Test product metrics provide information about the test state and testing status of a software product. Using these metrics the products test state and indicative level quality, useful for product release decision can be measured [15&16].

Some of the software testing metrics are:

1. Test Efficiency = (DT/(DT+DU)*100
2. Test Effectiveness = (DT/(DF+DU)*100
3. Test improvement TI = number of defects detected by the test team during / sourcelines of code in thousands
4. Test time over development time TD = number of business days used for product testing / number of business days used for product
5. Test cost normalized to product size (TCS) = total cost of testing the product in dollars / source lines of code in thousands
6. Test cost as a ration of development cost (TCD) = total cost of testing the product in dollars / total cost of developing the product in dollars
7. Test improvement in product quality = Number of defects found in the product after release / source lines of code in thousands
8. Cost per defect unit = Total cost of a specific test phase in dollars / number of defects found in the product after release
9. Test effectiveness for driving out defects in each test phase = (DD/(DD+DN)*100
10. Performance test efficiency (PTE) = requirement during perform test / (requirement during performance time + requirement after signoff of performance time) * 100%
11. Cost per defect unit = Total cost of a specific test phase in dollars / number of defects found in the product after release
12. Estimated time for testing
13. Actual testing time
14. % of time spent = (actual time spent / Estimating time)*100

Where

DD : Number of defects of this defect type that are detected after the test phase.
DT : Number of defects found by the test team during the product cycle
DU : Number of defects of found in the product under test (before official release)
DF : Number of defects found in the product after release the test phase
DN : Number of defects of this defect type (any particular type) that remain uncovered after the test phase.

Application of metrics related to Time as external quality factor in Life Cycle Assessment:

Based on the above metrics, test lead/manager will get the understanding of the below mentioned key points.

a) %age of work completed
b) %age of work yet to be completed
c) Time to complete the remaining work
d) Whether the project is going as per the schedule or lagging? etc.

The practice of applying software metrics to different operational factors and to maintain factors is a complex task. In spite of the difficulties it is important to do metrics since this assessment helps to diagnose the status in terms of time, completion and also the quality and thus ensures to take meaningful, effective remedial action (e.g., controlling requirements changes, improving response time to the developer, relaxing performance requirements, extending the project schedule, adding money, or any number of options). Further, based on the metrics, if the project is not going to complete as per the schedule, then the manager will approach the client and other stake holders by providing the reasons for lagging to avoid the last minute surprises.

Thus, time or duration of the project is an important factor that may potentially interfere with successful completion of the project. Application of metrics to measure the time period as will certainly help the project team for successful delivery i.e., delivery by the targeted time. It also helps to avoid financial losses due to delay of the delivery, if any by avoiding through proper actions.

Aside from the occurrence of errors or defects, and their number (if any), the major metric for quality is the mean time between their occurrence. Whether you record time to failure, time intervals between failures, cumulative failures in a given time period, or failures experienced in a given time interval, the basic metric of reliability is *time*.

A survey has been conducted for the present study to identify the impact of time over the quality of the project since time is also have considerable impact as an external factor in the delivery and the quality of the project. 50 projects of 5 categories, 10 projects for each (in terms of project completion time) were selected for the study as: more than enough, Enoughwith-out difficulty, enough with difficulty, possible with extreme difficulty and not possible even with extreme difficulty. Quality of the project was estimated by giving %age to the Customer satisfaction like:

*Well satisfied: 81-100%, Satisfied: 61-80%, Good: 41-60%, Average: 21-40%, Not satisfied: <20%

Table-1: Project duration Vs Quality of the Project

S.No	Project Category	Number of projects taken	Level of estimated time	Average % of work completed by the time	Quality of the project observed after delivery (in terms of Customer satisfaction)
1	A	10	More than enough	>85	Satisfied
2	B	10	Enough with-out difficulty	>85	Well Satisfied
3	C	10	Enough with some difficulty	80-85	Good
4	D	10	Possible with Extreme difficulty	<=85	Average
5	E	10	Impossible even with extreme difficulty	<=50	Not Satisfied

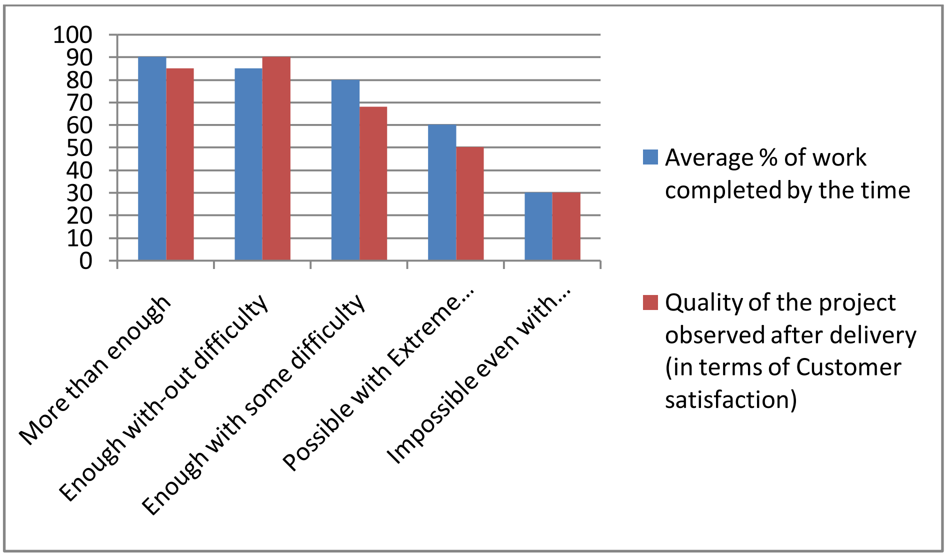

Figure-1: Project duration Vs Quality of the Project

Based on the results shown by the study it was noticed that time duration of a project has much influence over the quality of the project. It was also noticed that the customers were not satisfied with the delivered project when the project was completed with-in short time i.e., impossible even with extreme difficulty. In the projects with enough time the quality of the project was observed as good and satisfied.

IV. CONCLUSION

Software Quality Metrics Provide a systematic way to assess quality based on a set of clearly defined rules and also provide an “on-the-spot” rather than “after-the-fact” insight into the software development.To the end user of an application, the only measures of quality are in the performance, reliability, and stability of the application or system in everyday use. This is "where the rubber meets the road," as users often say.In order to achieve customer satisfaction by improvising the quality of the project future research is need to be extended in the area of software testing metrics to develop appropriate metrics to evaluate the project considering the external factors on the delivery and also the future applicationof the project.

REFERENCES

[1] Ming-Chang Lee, *National Kaohsiung University of Applied Sciences, British Journal of Applied Science & Technology, 4(21): …………., 2014. www.sciencedomain.org*

[2] Vennila G, Anitha P, Karthik R, Krishnamoorthy P. A study of evaluation information to determine the software quality assurance, International Journal of Research and Reviews in Software Engineering. 2011;1(1):1-8.

[3] Ma Y, He K, Du D, Liu J, Yan Y. A complexity metrics set for large-scale object oriented software system, In proceedings of the Sixth IEEE International Conference on Computer and Information Technology, Washington, DC, USA. 2006;189-189.

[4] Aggarwal, K.K., Sing, Y., Kaur, A and Malhotra, R., 2006, “Software Design Metrics for Object oriented Software, “ Journal of Object Technology, Vol.6, No. 1, January-February, pp.121-138.

[5] Zuse, H., 1997, “ Reply to: Property-Based Software Engineering Measurement, “ IEEE Transactions on Software Engineering, Vol. 23, No.8, August, pp.533.

[6] Florac, W.A., 1992. Software quality measurement: A framework for counting problems and defects: DTIC document. Technical Report: CMU/SEI-92-TR-022, ESC-TR-92-022.

[7] Devnani-Chulani, S., 1998. Modeling software defect introduction. Proceeding of California Software Symposium'97. Center for Software Engineering, University of Southern California.

[8] Iftikhar, A. and S.M. Ali, 2011. Software quality assurance a study based on Pakistan’s software industry. Pak. J. Eng. Technol. Sci., 1(2): 65-73.

[9] Imam, A., S.A. Khoja and I. Shariff, 2007. Improving Software Quality: A Benchmarking Approach. Software Engineering Research and Practice CSREA Press, ISBN: 1-60132-034-5.

[10] Stavrinoudis, D. and M. Xenos, 2008. Comparing internal and external software quality measurements. Proceedings of the 8th Joint Conference on Knowledge-Based Software Engineering (JCKBSE). IOS Press, Piraeus, Greece, August 25-28, pp: 115-124

[11] Mekprasertvit, C., 2004. Applying broadcasting/ multicasting/secured communication to agentmom in multiagent-systems. CIS 895-MSE Project, Kansas State University.

[12] Farooq, S.U., S. Quadri and N. Ahmad, 2011. Software measurements and metrics: Role in effective software testing. Int. J. Eng. Sci. Technol., 3(1): 671-680.

[13] Jørgensen, M., 1999. Software quality measurement. Adv. Eng. Softw., 30(12): 907-912.

[14] Boegh, J., S. Depanfilis, B. Kitchenham and A. Pasquini, 1999. A method for software quality planning, control and evaluation. IEEE Software, 16(2): 69-77.

[15] Farooq SU. Quadri SMK, Ahmad N. Software measurements and metrics: role ineffective software testing. Internal Journal of Engineering Science and Technology.2011; 3(1):671-680.

[16] Premal BN, Kale KV. A brief overview of software testing metrics, International Journal of Computer Science and Engineering. 2011;1(3/1):204-211.