IJCSE International Journal of Computer Sciences and Engineering Open Access
Review Paper Vol.-7, Issue-1, Jan 2019 E-ISSN: 2347-2693

Proposed 4S Quality Metrics and Automated Continuous Quality (ACQ) Metrics Dashboard to Quantify Software Product Quality

Dheeraj[1*], Kalpana Sharma[2]

[1,2]Department of Computer Science and Engineering, Bhagwant University, Ajmer, Rajasthan, India

[*]*Corresponding Author: dheeraj.chhillar@gmail.com , Tel.: +91 9873938888*

Available online at: www.ijcseonline.org

Accepted: 12/Dec/2018, Published: 31/Jan/2019

Abstract— The purpose of this paper is to propose a set of test metrics required to quantify the quality of the software. A detailed research was done to analyse the testing process including functional, performance, security and usability testing around latest technologies covering cloud computing, big data, machine learning, artificial intelligence and internet of things. Effort, schedule, productivity, defects, quality and cost are fundamental parameters of any project. There are several metrics around these parameters covering all phases of project including project initiation, planning, executing, monitoring and controlling and closing. There was a time when weekly, monthly, quarterly or yearly metrics reports were published based on collected data. Confirming authenticity of that collected data was also a challenge. In current scenario looking at the adaption of continuous software engineering we proposed a new term called Automated Continuous Quality (ACQ) Metrics Dashboard which will act as product stability index or project health indicator. This could be used by organizations to track and generate all the required reports at real time. Any individual could select any of the project parameters for any period of time to generate a report. It would use continuous data collected by continuous monitoring of the tools.

Keywords—Metrics, Continuous Testing, Continuous Delivery, Continuous Integration, 4S Metrics, ACT (Automated Continuous Testing), T Model

I. INTRODUCTION

As per the current scenario the usage of the software's has been drastically increased. The client list of the software usage consists of the individuals and business organizations as well. As the market has the availability of the software's for the client which has shift the attention from the development to the quality maintenance. Due to the advancement in the field of information technology the users are well aware of the needs of him/her from the software and also at the time of development and selection of the software product the user also goes for checking the quality and other performance related parameters of the software. The quality checking after the complete development of the product is just similar to remaking the complete product and also increases the cost and time for the complete development. For reducing the efforts for the evaluation and quality management of the software product the performance evaluation is to be considered on the basis of the developers perspective [1]. Software metrics are used for the evaluation of the performance of the software product. Where the metrics are defined as the computable or measurable which indicates the performance statistics of the software module. In the previous researches on software quality management the authors have considered the factors and subfactors for the quality evaluation and also have considered the different metrices for the same. This work tries to cover most of the quality related metrices which highly affects the quality of the software module.

Quality as a total part defines the metrices which defines the different measurement used for defining the software quality based on user. The most of the metrices considers the user satisfaction as the major part. The parameters considered in these metrices are like effectiveness, performance, productivity and security in the real time environment which falls under this particular segment. In the metrices the metrices are internal and external and the legacy of the external factors lies in the internal factors, the internal factors are like qualities considered by the user and hence the developer have to consider the internal segments to make effective availability of the best external one for the client[2]. In the case of dynamic metrics the factors which are considered at runtime for the software modules, segments of program and related systems. Tahir et al. [3], Sandhu et al and Choi et al., in their work discussed some of the software metrics which are based on the evaluation of the software quality and also being used for the prediction of the dynamic measures of the software. Aspect oriented concept is being for the development of dynamic analyser tool (Aspectj), which is being used for dynamically analysing the various applications of java which are being used for the data

collection at runtime which is being required for the dynamic cohesion metrics and for the applications of dynamic coupling tracers for the computation.

As per the research considered by Srinivasan et al [4] Object oriented design metrices and measures are defined as described under:

1. **Methods-Per-Class Factor (MPCF):** The Method-Per-Class Factor (MPCF) is defined as the ratio of the Number of Public Methods (NPM) to the sum of the Number of Public Methods (NPM) and Number of Non-Public Methods (NNPM) in the class.
2. **Attributes-Per-Class Factor (APCF):** The Attribute-Per-Class Factor (APCF) is defined as the ratio of the Number of Private (Protected) Attributes (NPA) to the sum of the Number of Private Attributes (NPA) and Number of Non Private Attributes (NNPA) in the class.
3. **Method Inheritance Factor (MIF):** The Method Inheritance Factor (MIF) is defined as the ratio of the Number of Inherited Methods (NIM) to the sum of the Number of Inherited Methods (NIM) and the Number of Defined Methods (NDM) in the class.
4. **Attribute Inheritance Factor (AIF):** The Attribute Inheritance Factor (AIF) is defined as the ratio of the Number of Inherited Attributes (NIA) to the sum of Number of Inherited Attributes (NIA) and the Number of Defined Attributes (NDA) in the class.
5. **Coupling Factor (CF):** NAC is being as the number of Actual Coupling for other classes and NPC is being defined as the number of Actual Coupling for the rest classes of the system Clearly, the number of possible couplings of a class with other classes of the system is one less than the number of classes.
6. **Lack-of-Cohesion Factor (LCF):** NDMP is the Number of Dissimilar Method Pairs in the class and NPMP is the Number of Possible Method Pairs in the class. If two methods access one or more common attributes, then these two methods are similar.
7. **Package Cohesion:** From the prior literature [5], it has been found that there is a strong interconnection between coupling and cohesion in the way of measuring its level.

I. Sub-factor Quality Metrics

1. **Measuring Structuredness:** Alan Gillies [6] in his work defined that the well-structured code will always be easier for the maintenance and also for adaptation part of the same and the average length of the code actually is being used for defining the calculation of the length of the modules code.
2. **Measuring Readability:** For having the accessibility, the documentation of the working module may help well for the usability. As per the work by Alan Gillies the term readability is being evaluated by Flesch –Kincaid Readability Index and Fog Index measurement methods.
3. **Measuring Reusability:** Gaffney and Durek [1989] in there work discussed a concept for reusability of the software module and also represented the cost of the reusability of the modules as mentioned below[7]:
   - **Measuring Reliability:** The probability that the system will not fail under the usability as expected by the client side over a specific time spam is being termed as the reliability of the software module [7].
   - **Measuring Portability:** As per the research of Mallikarjuna et al, the evaluation of the cost of the porting contains the cost for the evaluation of the different interfaces of the software module. A figure for degree of portability can then be computed for a specific software unit is measured as follows.
     DP = (cost to port / cost to redevelop)

II. STRUCTURAL METRICS FOR PROCESS MODELS

Below are some of the metrices which are from the category of structural metrices used for the process modelling, which are described in the prior literature Garcia et al [8] are as follows:

- **Number of nodes:** number of activities and routing elements in a model.
- **Diameter:** The length of the longest path from a start node to an end node.
- **Density:** Ratio of the total number of arcs to the maximum number of arcs.
- **The Coefficient of Connectivity:** Ratio of the total number of arcs in a process model to its total number of nodes.
- **The Average Gateway Degree:** Expresses the average of the number of both incoming and outgoing arcs of the gateway nodes in the process model.
- **The Maximum Gateway Degree:** Captures the maximum sum of incoming and outgoing arcs of these gateway nodes.
- **Separability:** Ratio of the number of cut-vertices on the one hand to the total number of nodes in the process model on the other.
- **Sequentially:** Degree to which the model is constructed out of pure sequences of tasks.
- **Depth:** Maximum nesting of structured blocks in a process model.
- **Gateway Mismatch:** The sum of gateway pairs that do not match with each other, e.g. when an AND-split is followed by an OR-join.
- **Gateway Heterogeneity:** Different types of gateways are used in a model.
- **Cyclicity:** The number of nodes in a cycle to the sum of all nodes.

- **Concurrency:** The maximum number of paths in a process model that may be concurrently activate due to AND-splits and OR-splits.

III. QUALITY METRICS

The literature presented in this work provides a description of the basic parameters of the quality consideration of the software product. The quality is the main goal in many of the industry like automobile, aviation, construction, etc., while in the case of the software development quality consideration it's a challenge for the organizations. The projects under the software development are generally tricky or we can say are runtime running applications where the integration is to be carried out quite carefully in various activities like technical and also in the case of the project management. The issue of quality in the case of the software development has started to be considered in just past years, while in the case current development requirement the quality consideration is the major part because of the segments like economic, social and competitiveness among organizations for software development.

In the case when the high-quality software development is being considered then there is unpredicted emphasis is attached with it [8]. In Nasib S. Gill, in his study author explained that the software systems which are poorly tested reduces the efficiency of the software and results in the negative effects for the development. The author has also discussed about the 'Software Reliability Measurement' and also different approaches defined by ISO for the software development and quality assurance of the same. The organizations dealing with the software development has to maintain good development environment in order to get the high efficiency in quality consideration of the software module. In the case of the software testing the team should focus more on the satisfaction of the customer other then only considering the errors detected and resolving the same. The author also has provided a discussion on the software quality management and factors that can be helpful in enhancing the improvements.

L H Rosenberg et al. [9] in his work explained that the now a days the field of the software quality assurance is facing many of the issues and challenges starting from the different ways for interpreting the quality assurance. The development team should have proper understanding of what actually a quality means, the description of the software module is also influenced by the environment where the software is to be used. There are many aspects related to the SQA as that of in the case of the software development lifecycle and those which spans many of the phases. SQA is being considered as the critical phase in the development which can also influence the ultimate output and the consideration of the SQA requires high quality software development and testing skills. In the core development field the segments like software safety and reusability are also being added to the list.

M. Agrawal et al. [10], in their work described the process of developing the high quality software product by defining the maturity of the product on-time and considered cost, for which the five CMM levels are to be considered which are defined from the organization's side. The ultimate goal of the work was to study the factors like effort, quality and time of the cycle. The author has used the linear regression technique on the data gathered from the 37 different level 5 software projects from 4 different companies. The maturity level of the software product is being defined by the CMM level 5 which actually reduces impact of the factors which previously are considered to be helpful for maintaining the quality, cost, size and cycle time. On the basis of the different model of development the efforts for the cycle time is around 12% and for the defect's consideration something around 49% efforts of the total considered for the development. In the work the author shown that the variance is caused majorly due to the factors other than the size of the software.

C. Woody et al. [11], in their work have described the requirement of the SQA just because of the growing need of the society over the software products. The work also considers the how and what type of queries related to the software product and also addresses the actions required for the same. The efforts estimation as per current scenario considers the rules of the SQA and also the issues for proper training on the same is being discussed in the work.

A. Janus et al. [12], in their work have discussed a new approach agile quality assurance where the concept of software metrices and agile metrices are combined for the better measurement of the quality of software in continuous integration. In the proposed methodology the continuous integration and continuous measurement are added to the subsequent activities of CI and forms Agile Quality Assurance.

Wei Li [13], in his work have explained the issues related to the component-based software systems. The complete spectrum of the QoS is being considered initially and then the physical and logical requirements are analysed by the means of different reconfiguration strategies. The software work is being classified on the basis of different characteristics of the QoS. Subsequently, a quantitative evaluation of the QoS assurance abilities is carried out and the results are discussed.

IV. PROPOSED 4S QUALITY METRICS

4S denotes Safety, Scalability, Stability and Serviceability to test Security, Performance, Functionality and maintainability respectively. This term is coined in relation to the software product and could be used in other fields also. Based on research we found 4S Quality metrics parameters are most important to ensure successful release of the software product in today's scenario. It touched all the major software quality attributes. There are a lot of other quality attributes at each level of software testing, however these are considered as most critical and could categorize all other attributes as well.

Safety: Safety metrics is to ensure security of the overall software product to protect it from internal and external threats and all vulnerabilities [14]. To ensure all authorizations work as expected, it is hack proof, virus proof etc. Security tool will be integrated with the ACT test bot to ensure data protection, data privacy[15].

Scalability: Scalability metrics is to consolidate all metrics related to performance parameters and non-functional requirements. Some of the basic performance parameters are considered as Access time, load time, run time, latency, throughput, capacity, efficiency etc. At design level and architecture level performance metrics should be captured to ensure scalability of the software product considering latest tools and frameworks of development used.

Stability: Stability metrics is to ensure that functionality of the product remains intact and lesser defects are leaked to production. Even if any minor defects are leaked to production assuming 100% testing is not possible, such defects should not impact the stability and functionality of the product. End users of the application should not be impacted. There should be a back-up mechanism of the previous stable build to production if end users encounter any blocker issues in production. It should be done at real time and that is quiet possible if Automated continuous software engineering concepts are used.

Serviceability: It mainly involves maintainability, testability, modularity, install ability, localizability and modifiability. This is import metrics parameter in today's scenario because everything is being used as a service be it software, infrastructure, platform from cloud. This would help in right selection of the cloud, tools, framework, process, programing language, test and development tools, architecture etc. At the feasibility study level OR design level it should be considered.

V. ACQ METRICS DASHBOARD

Automated Continuous Quality Metrics dashboard is the integration of T model [16] and Automated Continuous Testing process with the data analytics tool to generate required quality metrics. All reports related to functional testing, performance testing, security testing, CI/CD build details, resources time entry tools and development reports are consolidated and feed into the data analytics tool. Time entry report tools and project management tools are used in majority of the organization to track schedule and efforts spent by each team member in each project phase of the different software products. This would help in auto generation of metrics related to productivity, effort and time. This data analytics tool will be able to generate graphical representation of metrics based on inputs like Date, Release, build etc. Various metrics could be defined in the data analytics tool for which management wants to have report. Metrics related to effort, schedule, bugs, code and test activities could be generated using this. Test metrics like defect yield, test case execution rate, regression testing time, build health, defect density, test efficiency etc could be easily tracked using this tool. The benefit of integrating metrics dashboard with T model is that it will generate metrics at real time because overall process starting development, testing deployment and bug logging is automated in this. Major problems organizations face in generating metrics is due to actual data collection. In this case Reliability of the data will be there because whole process is automated using Continuation Integration tool, automated bug logging tools, Integration with Functional, Performance and Security tools, Business Intelligence tools and data analytics tools. Overall health of the build could be indicated after each build deployment and could be sent to the group via email. This would confirm whether build is ready to test, have critical defects OR is not at all ready for testing.

Algorithm to generate Metrics Report:

Step1: Start

Step2: Take auto feed of all Management tool Report, Technical reports, QA reports, Bug reports to Metrics Dashboard's Data Analytics tool.

Step2.1: Technical Reports related to Development, Configuration Management and Continuous Integration tool

Step2.2: Test Reports- For Functionality, Performance and Security from

Step3: Select required parameters/details on UI layer to generate metrics. Date From/To, Release Name(s), Build Number(s), Effort, Schedule, productivity, Team Name(s), Resources.

Step4: Select type of Report required: Graph, Trend chart, Bar chart, Pie chart and generate report.

Step5: If any error, Not Vaild OR parameter to be modified- Go to Step 3 and start again.

Step5: Save reports OR Email/notify Reports to Management OR resources that need to report.

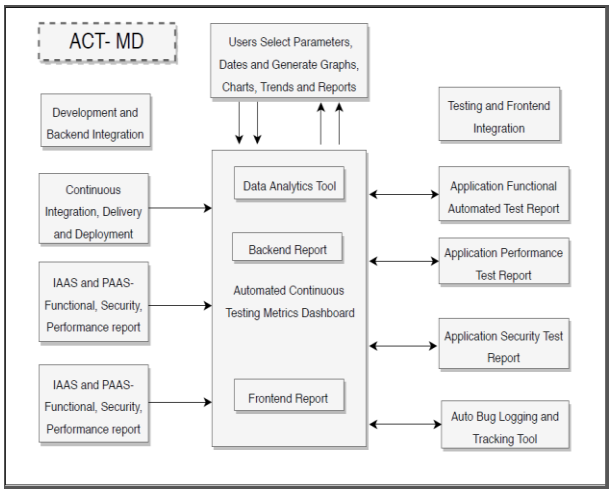

Figure 1. ACT Metrics Dashboard Framework

VI. CONCLUSION

A '4S quality metrics' is proposed as a software product stability index to quantify quality of the software. 4S denotes Safety, Scalability, Stability and Serviceability. Safety is to ensure that software is secure, which is a major threat these days; Scalability is to make sure software's performance remains intact in peak conditions. Stability is to represent bug free functionality and features of the software after each release whereas Serviceability is considered because everything in software world is being used as a service be it software, infrastructure OR platform. Also an ACQ Metrics dashboard framework is outlined to generate required metrics in various representations. ACT Metrics dashboard is based on Automated Continuous Testing and T model where data analytics tools are integrated with different types of Management, Dev, QA and bug reports at real time to have accurate and unambiguous reports. Further work could be done to implement this ACT Metrics dashboard framework and create a tool to be utilized across the industry.

REFERENCES

[1]. Aman Kumar Sharma, Dr. Arvind Kalia, Dr. Hardeep , "An Analysis of Optimum Software Quality Factors", IOSR Journal of Engineering, 2(4) ,2012 , 663-669.

[2]. C.Mallikarjuna, K. Sudheer Babu, P. Chitti Babu, "A Report on the Analysis of Software Maintenance and Impact on Quality Factors", International Journal of Engineering Sciences Research-IJESR , Vol 05, Article 01335, 2014.

[3]. Tahir, MacDonell, S.G., "A Systematic mapping study on dynamic software quality metrics", Proc. 28th IEEE International Conference on Software Maintenance ,Riva del Garda, Italy, 2012, 326-335.

[4]. K.P. Srinivasan, Dr. T.Devi, "A Complete and Comprehensive Metrics Suite for Object-Oriented Design Quality Assessment", International Journal of Software Engineering and Its Applications 8(2), 2014, 173-188.

[5]. G Bavota, A De Lucia, A Marcus, R Oliveto, "Using structural and semantic measures to improve software modularization", Emprical Software Engineering, 18(5),2013, 901-932.

[6]. Alan Gillies , Software Quality: Theory and Management , 3rd edition, Lulu.

[7]. C.Mallikarjuna, K. Sudheer Babu, P. Chitti Babu, "A Report on the Analysis of Software Maintenance and Impact on Quality Factors", International Journal of Engineering Sciences Research-IJESR , Vol 05, Article 01335, 2014

[8]. B. Kitchenham and S. Charters, Guidelines for performing Systematic Literature Reviews in Software Engineering. 2007.

[9]. Y. Levy and T. J. Ellis, "A systems approach to conduct an effective literature review in support of information systems research," Informing Sci. Int. J. Emerg. Transdiscipl., vol. 9, no. 1, pp. 181–212, 2006.

[10]. C. Wohlin, Experimentation in software engineering : an introduction. Boston: Kluwer, 2000.

[11]. M. J. Ordonez and H. M. Haddad, "The State of Metrics in Software Industry," in a Fifth International Conference on Information Technology: New Generations, 2008. ITNG 2008, 2008, pp. 453–458.

[12]. S. Misra and M. Omorodion, "Survey on Agile Metrics and Their Inter-relationship with Other Traditional Development Metrics," SIGSOFT Softw Eng Notes, vol. 36, no. 6, pp. 1–3, Nov. 2011.

[13]. D. Hartmann and R. Dymond, "Appropriate agile measurement: using metrics and diagnostics to deliver business value," in Agile Conference, 2006, 2006, p. 6 pp.– 134.

[14]. I. Hydara, N. Admodisastro, Information and Software Testing, Elsevier, Current state of research on cross-site scripting (XSS) – A systematic literature review, Volume 58, Feb 2015, Pages 170-186.

[15]. S.Subashini, Journal of Network and Computer Applications, Elsevier, A survey on security issues in service delivery models of cloud computing, Volume 34, Issue 1, January 2o11, Pages 1-11.

[16]. Dheeraj, International Journal of Electrical and Computer Engineering (IJECE)- Proposed t-model to cover 4S quality metrics based on empirical study of root cause of software failures, Vol 9, No-2, April 2019

Authors Profile

Mr. Dheeraj pursued Bachelor of Technology (Honors) from BM Institute of Engineering and Technology affiliated to MD University, Rohtak in 2005 and completed Master of Engineering from PDM College of Engineering affiliated to MD University Rohtak in 2009. He is currently pursuing Ph.D. and also working as a Senior Manager in Quality division with a US based company named CXC Infotech Private Limited. He has published research papers in reputed international journals including Scopus and UGC recognized journals. His main research work and interest include Software Testing, Automation Testing, Project Management, Software Engineering, Artificial Intelligence, Metrics, Agile and Software Architectures.

Dr. Kalpana Sharma pursued BA, M.Sc. in Computer Science, Masters of Computer Applications and Ph.D. She is currently heading Computer Science and Engineering Department at Bhagwant University, Ajmer, Rajasthan. She has published her research papers in lots of international journals of repute and presented papers at various conferences. Her research focuses on Software Engineering, Metrics, Security, Cloud, Artificial Intelligence, Metrics and Agile.

