

SOFTWARE QUALITY IMPROVEMENT THROUGH STATISTICAL ANALYSIS ON PROCESS METRICS

Karuna Prasad, Divya MG, Sarat Chandrababu and Mangala N

C-DAC, Bangalore, Karnataka, India

ABSTRACT

Software Quality can be considered as totality of features and characteristics of a product or service that bears its ability to satisfy stated or implied needs. The Quality of any software can be achieved by following by well-defined software process. These software process results into various metrics like Project metrics, Process metrics and Product metrics. Process metrics are very useful from management point of view. Process metrics can be used for improving the software development and maintenance process for defect removal and also for reducing the response time.

This paper describes on importance of capturing the Process metrics during the quality audit process and also attempts to categorize them based on the nature. To reduce such defect, corrective actions are recommended.

KEYWORDS

Software Metrics; ISO; Software Quality Audit; Process Metrics

1. INTRODUCTION

The quality of software is of utmost importance in the field of software engineering. Software quality also depends on the process which is carried out to design and develop the software. Even after the process is followed with minute care, the errors and defects may still exist. The quality of a software product is mainly determined by the quality of the process used to build it. Measurement and analysis will help in determining the status of the software process in terms of whether the process is followed and the functioning is as intended. Verification is the similar type of control from the management perspective. To meet such goals, quality audit for software process are conducted time to time. By measuring the errors and defects, we can take steps to improve the process.

The improvement of process will depend on metrics captured in the lifecycle of software. Software metrics can be classified into Project metrics, Process metrics and Product metrics [1]. Process metrics are management metrics which are used for improving the software development and maintenance process for defect removal and reducing response time of the process. Process metrics are invaluable tool for an organization who are wanting to improve their process. Usually

these process metrics are not used mostly because of uncertainty about which metrics to use, how to perform measurements and how to overcome such defects.

For software process improvement, there are many models which are available for example Capability Maturity Model (CMM), Bootstrap, Personal Software Process (PSP), IT Infrastructure Library (ITIL), IEEE, Six Sigma and ISO 9000 quality management system. These models evaluate the software product, quality and their drawback. Moreover locally designed actions can be initiated in areas where improvement is needed. The software process must be defined and documented. In addition to the processes, standards for the different work products to be defined, e.g. coding and document standards.

The rest of this paper is organized as follows. In section II we have presented our approach and objectives. In section III we have presented the literature review which is basis of our work. In section IV quality process is explained, in next section categorization of errors and defects are presented. In section VI we have presented corrective actions. In section VII data collection methodology is explained. In section VIII results and the analysis are discussed. Finally we have provided conclusion in section IX.

2. APPROACH AND OBJECTIVE

In this paper we have applied statistical quality assurance to the errors and defects reported during the quality audit for the year 2015 and 2016 in our organization. This has been done in view to improve the quality of software development process and hence the software products. We are presenting that by measuring the errors and defects we can take actions to improve them. We are also presenting how each and every errors and defects are grouped. There after each of them is categorized with severity like minor, moderate or serious. The data collected over a period of two years has been analysed and presented. The analysis also describes recommended actions for the corrective action. The idea has been inspired from the software engineering practitioners Roger S Pressman and Bruce R Maxim [2].

Broadly we are trying to address 3 objectives namely quality improvement, categorizing of errors and recommendation of corrective actions.

3. RELATED WORK

In [3] the authors have presented the mechanism of how software engineering capabilities relate to the business performance. They have proposed a structural model including the Software Engineering Excellence indicator which consisted of deliverables, project management, quality assurance, process improvement, research and development, human resource development and customer contact.

In [4] the author has shared how NASA's Johnson Space Center developed a 'statistical method' to determine sample size for the number of process tasks to be audited by SQA. The goal of this work is to produce a high quality product which is cost effective.

In [5] authors have said that technological choices are fundamental for project planning, resource allocation, and quality of the final software product. For analysis they have taken open source web applications available in SourceForge. Authors aim to provide tools to support project

managers. They have said that there is need to set thumb rule to guide technological choices to increase the quality of software artifacts.

This paper [6] is related to software product quality modelling and measurement. The outcome of the research is grouped as system-level software quality models, source code element-level software quality models and applications of the proposed quality models.

Our work focuses on applying statistical quality assurance to improve the quality of software products.

4. QUALITY PRACTICES BEING FOLLOWED AT OUR ORGANIZATION

International Organization for Standards (ISO) is an independent body that provides requirements, specifications, guidelines, characteristics etc that can be used consistently to ensure that materials, products, processes and services are fit for their function. ISO International Standards ensure that products and services are reliable and of good quality. The technical committees are made up of experts from the relevant industry, but also from consumer associations, academia, NGOs and government [7].

ISO 9001:2008 standards set out the criteria for a quality management system and are the only standard in the family that can be certified to. It can be used by any organization, large or small, regardless of its field of activity. In fact is implemented by over one million companies and organizations in over 170 countries. This standard is based on a number of quality management principles including a strong customer focus, the motivation and implication of top management, the process approach and continual improvement [7].

Our organization is ISO 9001:2008 certified. For the development of software, ISO 9001 process is been followed. The ISO related activities are mainly carried out by the quality assurance team. The main role of quality assurance team is ensuring Quality Management System conformance, promoting customer focus, and reporting on Quality Management System performance. A quality manager is traditional employee who has been given this responsibility. Monitoring the quality objectives that have been established and reporting this to top management is another traditional role of the quality manager. Having one person focus on the management of this important activity is a good idea to provide focus and direction.

Quality manager is also responsible for internal audit planning & management. Internal audit is the disciplined approach to evaluate and improve the effectiveness of software quality processes. The scope of internal audit is mainly risk management, control and governance of software processes.

5. CATEGORIZATION OF ERRORS AND DEFECTS

Software metrics is a standard of measure of a degree to which a software system or process possesses some property. It can be classified into three categories: Project metrics, Process metrics and Product metrics [1]. Project metrics are those that describe the project characteristics and execution example resource requirement, hardware requirement etc. Process metrics are statistical software quality assurance (SQA) data or management metrics which are used for improving the software development and maintenance process. The Process metrics is usually

captured in the software quality audit process such as deviation from process, effectiveness of defect removal during development, propagation of error from phase to phase. Product metrics focus on the quality metrics of deliverables and are used to measure the properties of software like lines of code (LOC), defects/KLOC, defect density, customer satisfaction etc.

Process metrics is collected through the SQA audits. The error and defects so found are categorized in 12 types [2]. Most of the categories are self-explanatory however we have listed here them for the purpose of more clarity. All the errors and defects reported in “Auditor Note Sheet” are categorized as IID, IES etc depending upon the nature of error and defect.

1. Incomplete or erroneous specifications (IES) - Any specification incompleteness is captured in this category. Any deviations from the process manual or specification like approval missing, partial implementation etc are included. If any missing metrics in the specification/template is also considered as IES.
2. Misinterpretation of customer communication (MCC) - Any deviation from customer requirement, feedback, suggestion etc not captured are categorized in this category.
3. Intentional deviation from specification (IDS) - IDS relates to deviation from process manual, software requirement specification etc due to lack of suitable reasons.
4. Violation of programming standards (VPS) - Any deviation from standards or introduction or modification can be counted in this category.
5. Error in data representation (EDR) - Any deviation from data formats as declared in specification.
6. Inconsistent competent interface (ICI) - Any deviation from recommended interface related errors.
7. Error in design logic (EDL) - Any deviation from committed logic eg DFDs, UML or ER diagram.
8. Incomplete or erroneous testing (IET) - Any errors and defects reported in testing by stakeholder/ customer/ third-party user etc. after completion of testing.
9. Inaccurate or incomplete documentation (IID) - Any missing sub sections of process manual or incomplete documentation.
10. Error in programming language translation of design (PLT) - Any design feature not captured while implementation which can cause defects in products.
11. Ambiguous or inconsistent human/computer interface (HCI) - Any error or defects in graphical user interface.
12. Miscellaneous - Any other errors and defect not captured in above mentioned categories.

All of the above categories are further classified based on the severity of the error/defects. They are labelled as minor, moderate and serious. It is classified as minor if the error/ defect not critical

to impact the process. Similarly, the defect is classified as moderate if the process is observed to be followed but cannot be evidenced. If the error or defect is observed to have major deviation from process then it is categorized as serious.

6. CORRECTIVE ACTIONS

For each of the error and defect categorized above, a corrective action is recommended as discussed below;

1. Incomplete or erroneous specifications (IES) - Effective Peer Review to be conducted.
2. Misinterpretation of customer communication (MCC) - Effective implementation of requirement gathering techniques to be adhered to improve the quality of customer communication and specification.
3. Intentional deviation from specification (IDS) - Reasons to be captured for intentional deviation and same to be reviewed.
4. Violation of programming standards (VPS) - Reason to be captured for intentional violation and same to be reviewed.
5. Error in data representation (EDR) - Recommend to use tools for data modelling also perform more stringent data design reviews.
6. Inconsistent competent interface (ICI) - Recommend more appropriate technical reviews and trainings.
7. Error in design logic (EDL) - Recommend more appropriate technical reviews and trainings.
8. Incomplete or erroneous testing (IET) - Recommend to adopt more appropriate testing methodologies with proper test plans.
9. Inaccurate or incomplete documentation (IID) - Recommend to use tools for documentation and reviews.
10. Error in programming language translation of design (PLT) - Cross reference with design requirements and appropriate tools usage to be recommended.
11. Ambiguous or inconsistent human/computer interface (HCI) - Graphical user requirement techniques and technology to be recommended.

7. DATA COLLECTION : A USE CASE

At C-DAC [8] quality audit is conducted every quarter. Audit is conducted for every project which is in design & development phase or maintenance phase. Quality assurance team rolls out the schedule with project name, auditee, auditor, date, time, and venue. With this auditee will keep ready all document and details required for audit. After the audit auditor will submit "Auditor Note Sheet" to quality assurance team. Auditor note sheet contains audit errors and

defects, if any. Quality assurance team publishes the entire “Auditor Note Sheet” in ISO related intranet web site where all C-DAC members have access to these Note Sheets.

Table 1. Error categorization for year 2015.

Error Type	Serious Errors	Moderate Errors	Minor Errors
MCC	0	0	0
IES	1	2	6
VPS	0	0	0
EDR	0	0	0
ICI	0	0	0
EDL	0	0	0
IET	0	0	0
IDS	3	1	0
IID	0	0	0
PLT	0	0	0
HCI	0	0	0
MIS	0	0	0
	4	3	6

Table 2. Error categorization for year 2016.

Error Type	Serious Errors	Moderate Errors	Minor Errors
MCC	0	0	0
IES	1	2	11
VPS	0	0	0
EDR	0	0	0
ICI	0	0	0
EDL	0	0	0
IET	0	0	0
IDS	2	1	0
IID	0	0	0
PLT	0	0	0
HCI	0	0	0
MIS	0	0	0
	3	3	11

For our experiment we have taken 2 years data namely Year 2015 and Year 2016. Based on our quality assurance guidelines of our organization these errors and defects are grouped as serious, moderate and minor which is described in section V. Also based on its nature every error or defect is categorized as IID, IES etc, same is recorded in the Table 1 and Table 2. Figure1 and Figure2 capture the severity of the errors thus categorized.

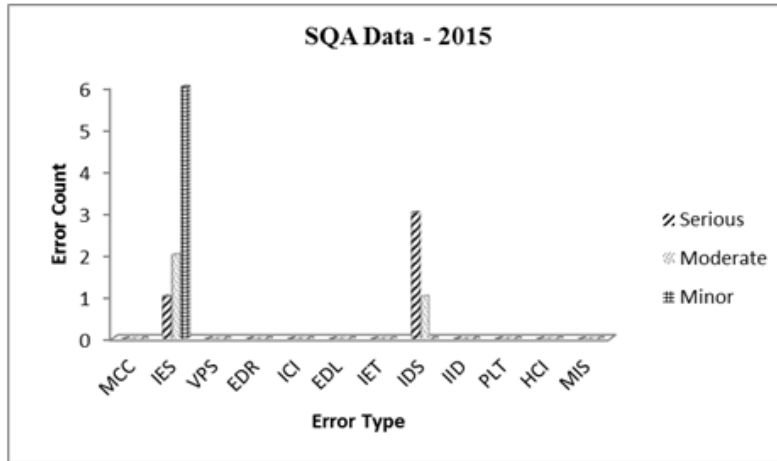


Figure 1- Severity of errors captured for year 2015

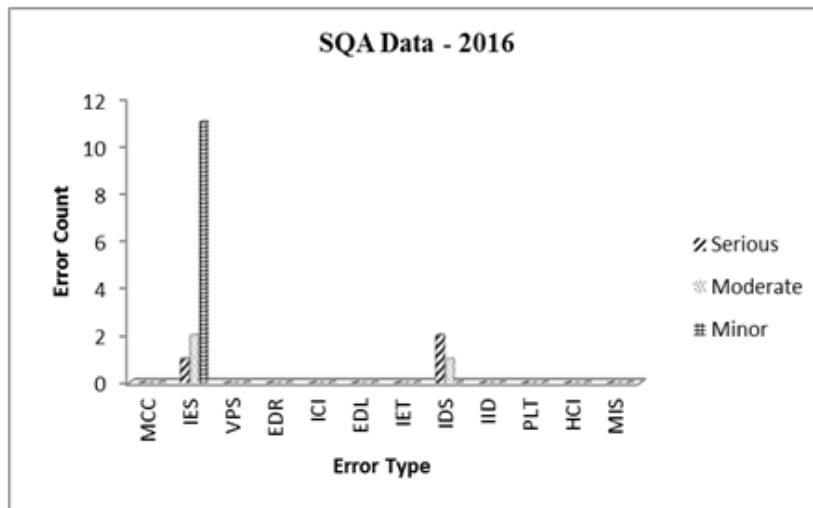


Figure 2- Severity of errors captured for year 2016

8. ANALYSIS AND RESULT

Every year three internal audits and one external audit's are conducted. Internal audit is conducted by Software Quality Assurance team of C-DAC, external audit is conducted by third party. During the audit, auditors will recode their observation, errors and deviations. This is termed as "Non Conformity- (NC)" in "Auditor Note Sheet" statement. We have collected all the NC's reported, same is categorized as per section V and grouped as serious, moderate and minor. From the analysis recorded at Table 1 & Table 2 the total errors and defects are presented in Table 3. The total serious, moderate and minor errors of both the years are represented in Table 4. Figure3 and Figure4 projects the cumulative errors for two years.

Table 3 – Cumulative errors for 2 years

SI No	Year	Total errors
1	2015	13
2	2016	17

Table 4 – Severity of Cumulative Errors

Type of errors	Year 2015	Year 2016
Serious	4	3
Moderate	3	3
Minor	6	11

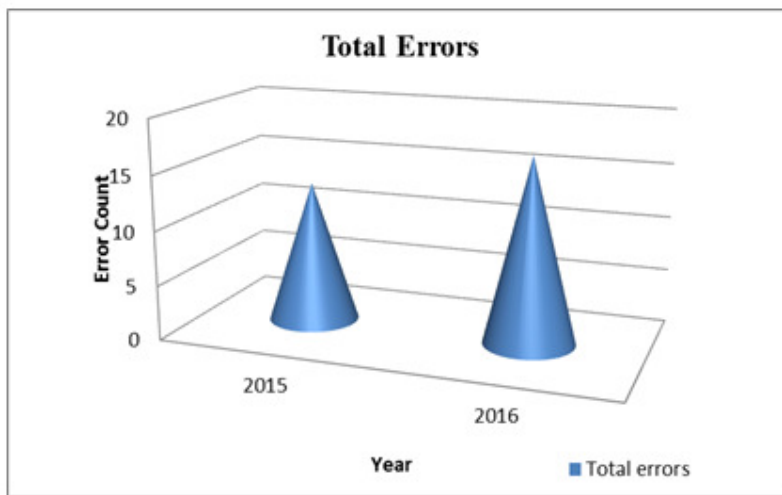


Figure 3 – Projection of errors

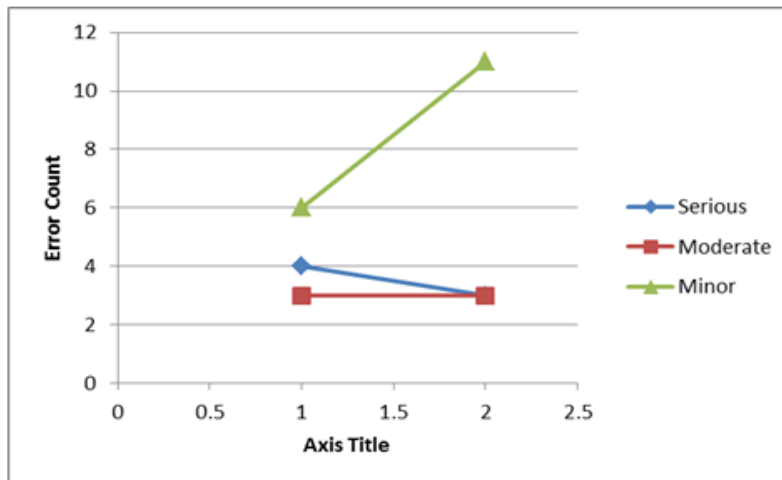


Figure 4 – Cumulative Projection of Severity errors

All the errors and defects are categorized and grouped mainly to know the statistics of software quality of projects. The data represented in Table 3 for the year 2015 is collected from 9 projects. The projects are either in design, implementation or maintenance states. The projects belong to various domains such as distributed computing, cryptography, high performance computing, Internet of things, mobile applications etc. These projects are implemented in programming languages java, c, python and other scripting languages. Some of these are using databases.

In Table 3, it is recorded that in year 2015 total error reported was 13. Out of which 4 are serious, 3 are moderate and 6 are minor type. The one serious error was due to Incomplete or erroneous specifications- effective 'peer review process' was recommended. Remaining 3 serious errors was due to Intentional deviation from specification – reason was WBS not updated, approval was not taken in time etc. All the causes of error was analyzed and training provided on quality process.

Also, there were 2 moderate and 6 minor errors due to 'Incomplete or erroneous specifications' and one more was due to 'Intentional deviation from specification'. In both the case effective peer review process and training on quality process was recommended. Similar analysis was carried for the year 2016.

The objective of the paper is to measure the errors and defects (non conformity) of all the projects, review it and recommend the appropriate corrective action. So that the project development cost will not over shoot, it can be delivered in time hence the quality of the project will improve. Hence software quality of products delivered by organization improves.

9. CONCLUSION

To improve the software quality, we collected software Process metrics. Our focus was mainly towards collecting metrics obtained through the quality control process. The errors and defects found through the software quality audits was our prime focus. These defects were subsequently categorized into 11 types. An analysis of such defects was conducted and recommendation for improving such defect and process are suggested. It was found that after implementing the recommendation the defects captured for the next subsequent year was reduced.

ACKNOWLEDGEMENT

We thank Centre for Development of Advanced Computing (C-DAC), the premier R&D organization of the Ministry of Electronics and Information Technology (MeitY) for supporting us to carry this work. We thank the Ms Veena KS from Software Quality Assurance team,C-DAC, Bangalore for sharing data.

REFERENCES

- [1] Ashwin Tomar and V. M. Thakare, "The Survey of Metrics on Software Quality Assurance and Reuse", National Conference on Innovative Paradigms in Engineering & Technology (NCIPET-2013)
- [2] Roger S. Pressman, Bruce R. Maxim, "Software Engineering: A Practitioner S Approach, Eighth Edition", 2015, McGraw-Hill Education.
- [3] Yasuo Kadono, Hiroe Tsubaki, Seishiro Tsuruho. 2008. "A Study on Management of Software Engineering in Japanese Enterprise IT Industry", 978-1-4244-3397-1/08/\$25.00 ©2008 IEEE

- [4] Neera Bansal Talbert, Paramax Space System, Texas. “Representative Sampling within Software Quality Assurance”, 1063-677393 \$3.00 0 1993 IEEE
- [5] Valentino Sartori, Birhanu Mekuria Eshete, Adolfo Villafiorita. 2011. “Measuring the Impact of Different Metrics on Software Quality: a Case Study in the Open Source Domain”, ICDS 2011 : The Fifth International Conference on Digital Society, ISBN: 978-1-61208-116-8
- [6] Peter Hegedus, University of Szeged, Hungary. 2015. “Advances in Software Product Quality Measurement and Its Applications in Software Evolution”, ICSME 2015, Bremen, German.
- [7] www.iso.org
- [8] www.cdac.in
- [9] Chandramouli Subramanian, Chandramouli Seetharaman, B. G Geetha Saikat Dutt, “Software Engineering”, 2015, Pearson India Education Service
- [10] Rajib Mall, “Fundamentals of Software Engineering”, 4th Edition, PHI Learning Private Limited