Advances in Computer Science and Information Technology (ACSIT)
Print ISSN: 2393-9907; Online ISSN: 2393-9915; Volume 2, Number 2; January-March, 2015 pp. 151-155
© Krishi Sanskriti Publications
http://www.krishisanskriti.org/acsit.html

Aspects of Effective Software Metrics

Gaurangi Saxena[1], Asmita Gupta[2] and K. Chandrasekaran[3]

[1,2,3]Department of Computer Science and Engineering, National Institute of Technology Karnataka, Surathkal, Mangalore 575025
E-mail: [1]saxenagaurangi94@gmail.com, [2]guptaasmita.ag@gmail.com, [3]kchnitk@ieee.org

Abstract—*Software metrics are an important entity in the software development life cycle as they provide an effective way of measurement and evaluation of different processes involved in the software development such as documentation of software requirements, tests, designs and programs. .Measurement proves to be helpful in answering various questions associated with the successful implementation of any software process. Software metrics facilitate software management and prove to be vital in the successful development of software. There are many approaches available to deploy software metrics in the software development process. The Goal Question Metric (GQM) is one such approach that focuses on effective implementation of the software programs by stressing that the first step for any organization implementing a software program, should be to specify the goals for itself and its projects. After the specification of goals, the goals must be traced to the data that is to be used to define these goals operationally, and finally a framework to make sure that the interpreted data is incongruous with the stated goals. Information is quantified whenever possible, and this information is further analyzed so as to predict whether the goals can be achieved. In this paper, we focus on the application of Goal/Question/Metric approach in different practical scenarios and compare the results of application to these scenarios in order to understand the vitality of software metrics. A case-study conducted by us further illustrates the implementation of GQM to different teams following different strategies in developing a software product. In particular, we give an extensive comparative study illustrating how GQM affected the software development process followed by these teams.*

1. INTRODUCTION

Software metrics are an important entity in the software development life cycle and provide an effective way of measurement for the software development [1]. In development of large scaled software, complexity arises and hence makes it difficult to control quality. Software metrics help in establishing a control over software quality. The concepts of software metrics are coherent, understandable and well established, and many metrics related to the product quality have been developed and used. Software development requires an evaluation and measurement technique for the feedback and evaluation of the various sub processes involved. Measurement proves to be helpful in answering various questions associated to the successful implementation of any software process. It facilitates project planning and enables us to determine the weaknesses and strengths of the current products and processes. It also provides a rationale for refining or adopting the appropriate technique [2]. Overall, measurement aids us throughout the course of the project, helping us asses its progress and take an appropriate action based on this assessment, or to evaluate the impact of such an action. The rapid development in the software industries has led to the software metrics being developed at the same rate. Software metrics are becoming increasingly crucial in a software management and also in the successful accomplishment of the development process. The metrics enhance the rate of progress in productivity of a software product and also critically evaluates the quality. The metrics field is being developed to help the organizations in quantifying their success in a better way [3].

Section 2 of this paper discusses the related work in this area, Section 3 discusses the studies carried out, followed by discussions in Section 4 and results in Section 5.

2. RELATED WORK

Literature Survey

Least square regression analysis is one of the most common techniques used. The technique is simple to use and is also easily accessible from many of the popular statistical packages. Robust Regression analysis has been deployed in [4] and [5], to screen for outliers in the models of metrics. It claims that by altering the error measure, the model under consideration can become more resistant to outline data points. Neural networks is one of the most common model building techniques, that has been used extensively as a substitute to least mean squares regression. Fuzzy systems have only been used for software development models [6]. A fuzzy system is basically expressing variables in linguistic terms, such as small [7]. Various systems can be easily simulated using fuzzy systems as it consists of crisp rule systems [8]. Rule based systems has also been used for modelling development of a software [9].

Case based reasoning is a technique to records observations, such as data about a project specification as well as the effort required to implement it. Regression and Classification trees based on the same principles each have a different aim. Regression Trees are useful in the cases when the resulting value lies in the specified interval domain [10]. Decision trees are used to guess the most probable output class for the given observation. Table 1 summarizes these techniques and provides a comparative study.

Table 1: Comparative Study

Technique	**Model free**	**Can resist outliers**	**Explains output**	**Suits small datasets**	**Can be adjusted for new data**	**Reasoning processes visible**	**Suits complex models**	**Include known facts**
Least Square Regression	✗	✗	Partially	✗	✗	✓	✗	Partially
Robust regression	✗	✓	Partially	Partially	✗	✓	✗	Partially
Neural networks	✓	✗	✗	✗	Partially	✗	✓	Partially
Fuzzy systems	✓	Partially	✓	✓	Partially	✓	✓	✓
Rule based systems	✗	N/A	✓	N/A	N/A	✓	✓	✓
Case based reasoning	✓	Partially	✓	Partially	✓	Partially	✓	✗
Regression trees	✓	✓	✓	Partially	✓	Partially	✓	Partially
Classification or decision trees	✓	✓	✓	Partially	✓	Partially	✓	Partially

Drawbacks

Disadvantage of neural network is that they operate as black box. The user is not aware of any information about how outputs are derived [8]. This nature of neural network makes it difficult to evaluate the network output gradient vectors. In fuzzy system it's difficult to specifying the system with a very high and acceptable accuracy by also preserving the degree of meaningfulness. Rule-based systems, unlike fuzzy systems do not have a degree of true or false and hence all the antecedents and consequents can only be true or false. Case-based reasoning systems cannot tolerate noise or inappropriate features. Robust regression can be used only to indicate the probable data points [11].

3. GOAL QUESTION METRIC APPROACH

The Goal Question Metric (GQM) [12] approach basically focuses on the first step for any organization should be able to specify goals for itself and for its projects. After specifying the goals, the goals must be traced to the data that is to be used to define these goals, and finally a framework to make sure that the data which is interpreted is incongruous with the mentioned goals. Information needs to be quantified whenever possible and required, and this quantified information can be analyzed as to whether the goals are achieved or not.

The result of this Goal Question Metric approach application [13] is the specification of a measurement system which targets a particular set of problems and rules which are being used for interpretation of measurement of data. The resulting measurement model has three levels:

1. GOAL: "A goal is defined particularly for an object, for a variety of reasons, with respect to various models of quality, from various points of view, relative to a particular environment." [14].The various objects of measurement can be products such as Deliverables and documents that are produced in the development life cycle for eg. designs, programs etc., processes such as, testing, designing, specifying and interviewing or Resources used in order to produce outputs; E.g. personnel, software, hardware, office space.
2. QUESTION: This refers to a set of questions which is used to depict how the assessment of a specific goal is going to be carried out. These questions assess the quality from a **particular** viewpoint [15].
3. METRIC: A set of data which is related with every question in order to find the answer to it in a quantitative manner. This data can be Objective, if it depends only on the object being measured or subjective if it depend on the object being measured and also the viewpoint under consideration [16].

3.1. Implementing GQM in our Case Study

Our case study revolves around how GQM was applied to different teams whose ultimate goal was to develop a software product. This study adopts the process used in [17] and attempts to find the effectiveness of applying software metrics in the development process. We considered three teams consisting of 15-20 members each:

Team A: Had a centralized group of critics (3-5) within the team, which periodically reviewed the software product being developed by the developers and gave their instructions to the remaining members on how to improve the product.

Team B: A decentralized team in which all the members periodically asses and review the software product. Based on their assessment they brainstorm their ideas on how to improve the product being developed.

Team C: Did not use any software metrics throughout the development process. They depended on the end result to ensure they achieved their primary goal.

The goals broadly identified were:

- *Goal 1:* Efficient project planning.
- *Goal 2:* Increase software reliability.
- *Goal 3:* Enhanced user satisfaction
- *Goal 4:* Improving speed
- *Goal 5:* Increase software productivity.

We divided our study into two phases, namely Fact finding and Data collection for Teams A and B, which have been discussed below:

Phase one: Fact Finding: This phase was designed to understand in what state each team believed its metrics program to be in. This was done in order to find about the nature of two teams, metrics being used and more importantly how they were used and also the management of metrics. This phase helped in forming the baseline for phase two. This phase was carried out by conducting interviews of the people responsible in both the groups.

Phase two: data collection. In this phase we handed out detailed questionnaires to members of both the teams to find out the details about the functionality of the metrics programs used by them. This phase helped us to understand how each team was affected by the metrics.

The questionnaire developed by us is given in Table2 .

The aims of the questionnaire were to

- Recognize which metrics were being used.
- How satisfied were the members with the development process and their product.
- Identify the way metrics program was being implemented and consequently managed,
- Identify the contributions that each member in a team made to metrics. Also to implore the understandings and experiences of the success of metrics.

Table 2: Questionnaire

QUESTIONS

Which team are you a part of? (Team A (Developer/Critics) , Team B, Team C)
What is the current request processing speed? To what extent are you satisfied with it?
How fast are the changes performed after the request is completed?
How much time is spent in improving the speed of the product from the time put in in developing the product?
Do you think that the cost incurred is worth the results?
Do you think that the product developed is efficient in fulfilling user needs?
How much time was spent in overcoming the problems faced by the users?
What was the accuracy of estimating the actual value of project schedule?
Do you think that the privacy of the user is maintained?
How does the product performed under heavy user load?

3.2 GQM Developed by Team A and B

In Team A, only a certain number of individuals used the GQM proposed. They were not concerned with the development process. Based on their results after using the metrics for each goal, they instructed the developer team on how to improve the product. The developer team just incorporated their changes.

In Team B, all the members used the GQM. Each member analyzed the metrics related to each goal and suggested their independent views on how improvements can be made. These ideas were brainstormed and a unanimously agreed decision was taken and a change was employed. All the members of this team were aware of the development process and the appropriate usage of GQM.

In Team C, none of the members used GQM. They had just a set of goals which ultimately led them to the finally goal. However, they never critically evaluated their sub-goals and the accomplishment of one goal was sufficient to lead them to start working towards achieving the next goal.

Table 3: GQM Developed

GOALS	QUESTIONS	METRICS
SPEED	What is the current request processing speed? To what extent are you satisfied with it? How fast are the changes performed after the request is completed? How much time is spent in improving the speed of the product from the time put in in developing the product?	(Time taken to complete execution /Expected time)*100 Time at update – Time of request (Time spent in improving speed)/(total working hours) * 100
LOW COSTS	Do you think that the cost incurred is worth the results?	(Revenue collected / Total Costs) * 100

USER SATISFACTION	Do you think that the product developed is efficient in fulfilling user needs? How much time was spent in overcoming the problems faced by the users?	This depends upon the percentage of problems rectified.
RELIABILITY	Do you think that the privacy of the user is maintained? How does the product performed under heavy user load?	Yes/no Time taken to complete request under heavy load / Expected time taken
PROJECT PLANNING	What was the accuracy of estimating theactual value of project schedule? What was the accuracy of estimating theactual value of project effort?	Actual Project Duration / Estimated project duration Actual project effort / Estimated project effort

4. DISCUSSIONS

After the development of a software product, members of each team was asked to score their satisfaction with different goals they had attempted to achieve from 1-5, 1 being the most satisfied and 5 being the least satisfied. We found out a mean score to analyze how satisfied the members of each team were, with their software product and development process.

Table 4: Mean Scores

Parameters	Team A	Team B	Team C
Low costs	3.4	2.5	3.6
Conformance	3.1	2.9	3.2
Speed	2.8	2.7	3.3
Reliability	3.0	3.1	3.5
User Satisfaction	3.6	3.2	3.9

5. RESULTS

We analyzed how useful the software metrics proved to be to both Team A and B. Team A was centralized in which only a few members followed the GQM approach. The developer group in this team did not find GQM as useful as the members of the critics' team. Due to this only a few members of this team found GQM to be helpful for them. In team B, all members used GQM and hence found it to be more useful than team A.

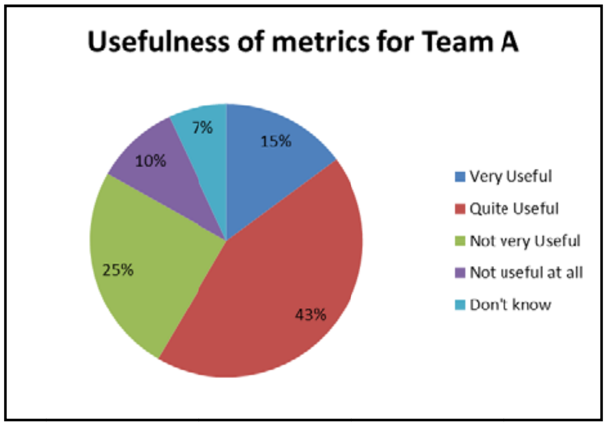

Fig. 1: Usefulness for Team A

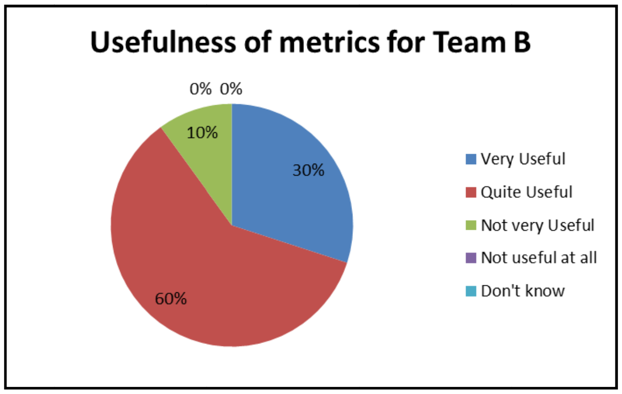

Fig. 2: Usefulness for Team B

The approach which is being adopted by the three teams gave us a particular insight about the usefulness of software metrics. Team A was centralized and hence only a few members were aware about the metrics being used in evaluating the goals. Hence, they lacked transparency among themselves. Team B was decentralized and followed incremental approach. Each member of this team gave an independent opinion and reviews about how the product can be improved. GQM helped Team B in avoiding problems, such as lack of transparency, since all the members were aware about the usage of GQM. Team C did not use GQM or any other software metrics. They were not able to critically evaluate their sub goals and hence in the end were not completely satisfied with their product or the development process. We observed that there should be a definite link between the metrics and improved goals. If this relationship is not clear, the practitioners are not motivated to put effort into the metrics collection. Practitioners at the Team B were reasonably satisfied and happy with metrics feedback whereas practitioners at Team A were less satisfied.

6. CONCLUSION

Our study revealed that that the carefully devised implementation strategy largely affects the success of a metrics program. We noticed that implementation of GQM can help ameliorate many problems arising in an organization. Several factors such as distribution of power, size play a crucial role implementation of GQM. However, metrics just indicate the problems and gives possible solutions to overcome these problems. It is the action taken by the user which solves these problems and brings in the desirable results. The measurements using these metrics are not the final goal. The goal is to gradually improve the software product through thorough analysis and feedback from the application of software metrics.

REFERENCES

[1] Fenton, N. E., & Neil, M. (1999). Software metrics: successes, failures and new directions. *Journal of Systems and Software, 47*(2), 149-157.

[2] Rawat, M. S., MGM's, C. O. E. T., Mittal, A., & Dubey, S. K. (2012). Survey on Impact of Software Metrics on Software Quality. *IJACSA) International Journal of Advanced Computer Science and Applications*, *3*(1).

[3] Gray, A. R., & MacDonell, S. G. (1997). A comparison of techniques for developing predictive models of software metrics. *Information and software technology*, *39*(6), 425-437.

[4] MacDonell, S. G. (1992). *Quantitative functional complexity analysis of commercial software systems* (Doctoral dissertation, University of Cambridge).

[5] Miyazaki, Y., Terakado, M., Ozaki, K., & Nozaki, H. (1994). Robust regression for developing software estimation models. *Journal of Systems and Software,27*(1), 3-16.

[6] Kumar, S., Krishna, B. A., & Satsangi, P. S. (1994). Fuzzy systems and neural networks in software engineering project management. *Applied Intelligence,4*(1), 31-52.

[7] Bastani, F. B., DiMarco, G., & Pasquini, A. (1993, May). Experimental evaluation of a fuzzy-set based measure of software correctness using program mutation. In *Proceedings of the 15th international conference on Software Engineering* (pp. 45-54). IEEE Computer Society Press.

[8] Ramsey, C. L., & Basili, V. R. (1989). An evaluation of expert systems for software engineering management. *Software Engineering, IEEE Transactions on*, *15*(6), 747-759.

[9] Griech, B., & Pomerol, J. C. (1994). Design and implementation of a knowledge-based decision support system for estimating software development work-effort. *Journal of Systems Integration*, *4*(2), 171-184.

[10] Srinivasan, K., & Fisher, D. (1995). Machine learning approaches to estimating software development effort. *Software Engineering, IEEE Transactions on,21*(2), 126-137.

[11] Frye, D., Zelazo, P. D., & Palfai, T. (1995). Theory of mind and rule-based reasoning. *Cognitive Development*, *10*(4), 483-527.

[12] Diederich, J. (1990, August). An Explanation Component for a Connectionist Inference System. In *ECAI* (pp. 222-227).

[13] Koziolek, H. (2008). Goal, question, metric. In *Dependability metrics* (pp. 39-42). Springer Berlin Heidelberg.

[14] GQM, G. Q. M. Goal Question Metric. *Softwaremetriken-Schätzverfahren*, 27.

[15] Caldiera, V. R. B. G., & Rombach, H. D. (1994). The goal question metric approach. *Encyclopedia of software engineering*, *2*(1994), 528-532.

[16] Daskalantonakis, M. K. (1992). A practical view of software measurement and implementation experiences within Motorola. *Software Engineering, IEEE Transactions on*, *18*(11), 998-1010.

[17] Hall, T., & Fenton, N. (1997). Implementing effective software metrics programs.*IEEE software*, *14*(2), 55-65.