Measurement of Software Maintenance from User Satisfaction Perspective: A Case Study

Nurdhiya Hazwani Helmee, Rozilah Rosli, *Koh Tieng Wei, Khaironi Yatim Sharif

Abstract - **The importance of software maintenance in a software development environment is a long term and continuous activity for improvement. The service quality of the software maintenance provided in which involving helpdesk, programmer and software analyst should be measured in order to satisfy users who use the software. In this paper, we propose a software measurement program for product improvement in order to maximize the user satisfaction where one government agency is selected as our case study. Our measurement plan is based on Goal Question Metric (GQM) approach. Results from this measurement program with some meaningful quantitative analysis has given an achievable roadmap for the organization to plan for next improvement activities toward their goals.**

Keywords—User satisaction; Software maintenance; Goal-Question-Metric, Software metric

I.   INTRODUCTION

As an overview, software maintenance is defined as "The process of modifying a software system or component after delivery to correct faults, improve performance or other attributes, or adapt to a changed environment" [1]. In addition, it is also defined as "The totality of activities required to provide cost-effective support to software system. Activities are performed during the pre-delivery stage as well as the post-delivery stage."[2]. Software maintenance processes and activities have been largely standardized. Standard organizations such as ISO, IEEE, and CMMI have detailed the activities to be carried-out by software maintainers [3].

The importance of software maintenance in a software environment is not only one-time activity for continuous improvement but it is a long term and improvement activity. During this stage, the adaptability of the software to the evolution of technology needs to be consistent and here the maintenance stage plays the important roles. High quality of customer service is extremely important for an organization to handle the maintenance stage. The customer service may consist of helpdesk or call centre which handle the process between end users and the application itself. Furthermore, measuring the software maintenance requires helpdesk ticketing information. This information gathers from end users which faced obstacles in using the system. The issue which consider as ticket log will be raise to helpdesk team and stored in helpdesk system.

*Corresponding Author: Koh Tieng Wei

Faculty of Computer Science and Information Technology
Universiti Putra Malaysia
43400 UPM Serdang, Selangor
MALAYSIA

To assist in routing of requests to appropriate support staff, each request would be categorized [4]. Based on the request, we can categorize this log as incident, problem or service request. Users may sometimes raise ticket issue of incident in term of configuration problem or interrupt to the system. Besides, problem can be considered as one or more incident occurs in the system. On the service request, user request based on information or seek advice. Thus, user request to reset password and provide services to user are the example of service request [5]. The requests also prioritize into low, medium or high priority. In this paper, the focus will be more on maintenance log category which is fall under problem type of ticket log.

In this paper, we propose a software measurement program for product improvement in order to maximize the user satisfaction where one government agency is selected as our case study. This paper is structured as follow. In Section II, we present the industrial problem background and scope which relate to our case study. In Section III, we focus more on related work regarding the measurement of software maintenance as the important issue in software development process. The methodology of study will be discussed in Section IV. Finally, recommendation and conclusion remarks will be in Section V.

II.   INDUSTRIAL PROBLEM

A. *Background*

ITD is IT division under of Agency J (both IT division name and the agency name has been changed to protect the identity of the organization). ITD manages a big system, System S (name has been changed to protect the identity of the organization) which the development began about 13 years ago and has been implemented by phases started five years after that to others agencies across the country who are supposes to use System S. It consists of estimated 3 thousands of tables and about 10 thousands application files where the organization needs to take care. Initially, the system was developed by vendor using Active Server Pages (ASP) language with SQL Server as database. Currently it is maintained by the internal staffs.

In this organization, Helpdesk (HD) department is organized by the internal staff. Users will issue any problems through the HD team while using System S and launch complaints through email, phone call, website or social media of the organization had. Some previous complaints they received expressed that the users are not satisfied with the system that has been maintained by the organization. This kind of bad scenario will give negative perception to the maintenance team as they should provide the best services in terms of correcting the system. All system issues facing by the

users will be recorded as logs in helpdesk system and be categorized as shown in Table I.

TABLE I. CATEGORY OF HELPDESK LOG

No.	Helpdesk Log Category	Description
1	Application -Process (AP)	System error
2	Application- Lookup (AL)	Imperfection in Business Rules
3	Network Access (NA)	Failure of network access
4	Data Verification (DV)	Defect in data
5	Performance (PF)	System slow response
6	Implementation (IMP)	Issues during module implementation
7	Enhancement (ENC)	Based on new policy
8	User Interface (UI)	Based on new interface
9	Explanation (EXP)	Advice from the team to users
10	Others (O)	Others

Based on Table 1, AP relates to system error faced by users. This category error will affect the users work process where the issue will affect user to stop their work task. In this paper, we will only focus on this category which gives major impact to the system functionality. The AL is where the business rules stored as a reference for users. Another log category is the NA and PF of the system which contributes the network and server team as well. Only several modules have been impact by IMP category. For ENC category, the development team need to take responsibility to meet the new requirement based on new circular or change in business process. In this case the internal customers or users will involve more. Error in UI often realize in the early stage of implementation where the module is not firm yet. EXP category shows the issue raise by the users is not an error. Therefore, helpdesk agent responsible to explain well to customers regarding the matter and advise them to follow the user manual or guideline in order to assist their understanding of system flow. Finally, other error found that do not relate in any of the category provide can be tag in O category.

TABLE II. STATUS OF HELPDESK LOG

No.	Helpdesk Log Status	Description
1	New (N)	New issue raised by users
2	Diagnosed (D)	New task for helpdesk/technical team to diagnose problem
3	Pending (P)	No response from helpdesk/technical team
4	Clarification (CL)	Not enough information from users
5	Explanation (E)	Explanation by helpdesk/technical team to user due to lack of users understanding
6	Closed (C)	Problem solve and patch to production server

On the other hand, helpdesk logs are managed based on the status. The status mentioned as below Table II above. Helpdesk team need to address quick response for current issue has been raised by the users and will set the status 'New'. This status alerts the team to escalate the issue to the right technical team. 'Diagnosed' status shows that the issue is

in action of diagnosis by the maintenance team either helpdesk, programmer or software analyst based on ticket numbers provided. 'Pending' status represent the log is still floating where no action taken by the technical team to solve the issue. In addition, 'Clarification' is for the specific user to provide complete requirement of issue because the helpdesk and technical team of the organization unable to process it without the complete one. 'Explanation' status is for users to take action on understanding the use of this system. There is no bug in this log. For the log with status 'Closed' it indicates that the helpdesk, programmers and/or software analysts have taken action on it. Issue that has been solved and is ready for patching stage where it involves the fixed application file is given with latest version in the production server. In our study, we used the 'Closed' status for our analysis section.

B. *Scope of Problem*

There are four types of maintenance, [2, 6] which Corrective Maintenance is to correct discovered program faults, Adaptive Maintenance to cater for changes in some way of business requirements and external environment, Perfective Maintenance to redo and clarify system structure, improve performance and maintainability and Preventive Maintenance to find fault before they become failures. In a study [7] concluded that the corrective maintenance nowadays may consumes up to 50%-68% of all maintenance.

In this paper, we focus on corrective maintenance which discusses on how to fix bug or correcting fault in term of application error log from complaint by end user. System S consists of several modules or sub systems related to all levels of staffs, either they are in support group or management level. The system helps them by automate their manual operation to system wise. The application error log as mentioned previously will based on one of the active module that have been implemented, Module L (name has been changed to protect the identity of the organization). This is an operational module which is the use of it on daily basis and it has been used by all their users across the country.
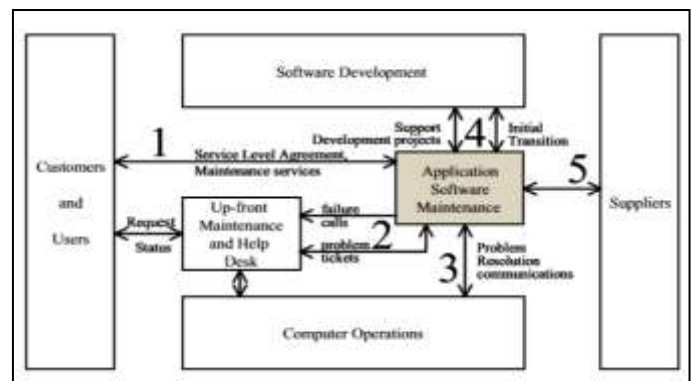

Figure 1. Software Maintenance Context Diagram

Figure 1 shows the context diagram of software maintenance process. This diagram adopted from [3] which presents several interfaces on software maintenance in an organization. There is a relationship between customers and users interface with application software maintenance through Service Level Agreement (SLA) in maintenance services. Once this agreement taken, users can receive services support on daily basis. In this application interface, users might face with problems and tend to raise issue to upfront maintenance

48

and helpdesk. In application software maintenance, there is another interaction with software development and computer system where the development team support the project and initial transition. The author also mentioned on the relationship between application software maintenance and suppliers which do not relate with our case study since the supplier site do not included where the organization maintains the system internally. In the author case, the suppliers are outsourcing. There may have more than one supplier such as subcontractor provide specific expertise during peak time, support service of yearly licensed software and partial or complete subcontractor on the same IT portfolio.
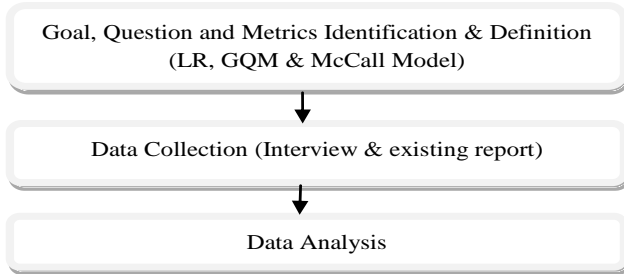
III. STUDY METHODOLOGY

Figure 2. Study Methodology

This study has been conducted base on case study approach. Since the objective of this paper is to propose a measurement program, therefore we adopted GQM as baseline of the proposal. In addition, we have adapted the relevant theory and ideas from the literature review (LR) to permit a meaningful measurement program for product improvement in software maintenance domain. We simplify our methodology as shown in Figure 2.

A. *Goal and Questions Identification*

From the problem statement we stated in Section II, we identified our goal to aligns to a propose solution. Hence, the goal of this study is to improve or maximize user satisfaction specifically of Module L and System S as a whole. Before defining the metrics, we identified three questions (Q1-question 1, Q2-question2 and Q3-question 3) that their answers will support on goal achievement based on adaption from the LR [9, 10] but tailored to the need of organization in our case study.

B. *Metric Set Definition*

Based on [11], Helpdesk services (HD) are based on all customer calls while corrective maintenace services metrics based on failure reports. Thus, in our case there is close relation between HD services and maintainenance service because HD is part of corrective maintenace team other than programmers and software analysts. As previously mentioned, HD responsible to record all customers problems either via phone calls or e-mails and they also analyze the records then come out with failure report (log report). All HD, programmers and software analysts perform their job using computer dedicated to each of them. They utilize that resource depending on their respective task. This is how we derived the metrics to answer Q1, Q2 and Q3 in order to achieve the goal.

Although our GQM model is adaption from [8] and [9] where in those papers they used Software Reliability as one of the metrics to answer Q1, but here in our case study we do not

consider that metric. We take software quality factors from McCall's model as part of sources for metrics formulation. Through Figure 2, McCall's factor model categorizes their 11 software quality factors into three – product operation, product revision and product transition [10]. Since our focus is on corrective maintenance where product revision factor dealt with so, we agree to ignore Reliability as it falls under product operations. To explain further on the metrics used in our GQM Model as depicted by Table III, we provide them in Table IV.

TABLE III   GQM MODEL

Goal	Questions	Metrics
Maximize user satisfaction	Q1-How many problems are affecting the user?	Backlog Total Number
	Q2-How long does it take to fix a problem?	Time taken (duration) from log opened to log closed
	Q3-Where are the bottlenecks?	• Staff utilization • Computer resources utilization

TABLE IV   DESCRIPTION OF METRICS

Metrics	Descriptions
Backlog Total Number	Backlog is a problem recorded as a log by different ID (ticket number) which is unique. It is a number of unresolved log over all log recorded.
Time taken (duration) from log opened to log closed	Duration is counted from the date log is recorded (log status equals to 'New') until the date it has been solved (status equals to 'Closed').
Staff utilization	Staff is service maintenance team consists of HD, programmers and software analysts who responsible to solve backlog.
Computer resources utilization	Each staff has his/her own complete set of computer which they use to solve backlogs.

C. *Data Collection*

Data were collected based on metrics set using two methods; interview the staffs and from existing reports started from October 2013 to December 2013. A total of eight interviews were informally conducted within that period by face to face and over phone. The interviewees were involved HD, programmers and software analysts. This is to make it simple and can minimize disruption of normal working pattern [11]. Most of existing reports are kept online and in the database, this makes process of data collection easier.

Depending on metrics, data related to log are historical data from record of 1st May 2013 to 30th June 2013 and for staff and computer utilization, they are some current data up to December 2013 and some is within May to June 2013. We used spreadsheet as a tool to keep and analyze the data. The important thing to highlight in this process is we put trust on data supplied to us without checking on their validity since the agency has already taken care of those records and they need to produce monthly report. However, we did some filtering on the data to discard unnecessary for our study.

D. *Data Analysis & Result*

We break down the analysis and result according the questions from our GQM Model (Table III).

Q1-How many problems are affecting the user?

Within the scope of work we defined earlier, Bar graph at Figure 5 shows total backlog for both months, May 2013 and June 2013. We can track log that still unresolved in May 2013. Out of total logs 56, 6 logs remain not solved. The logs were tagged with one specific indicator to show that they were under software analyst action. While for Jun 2013, all log
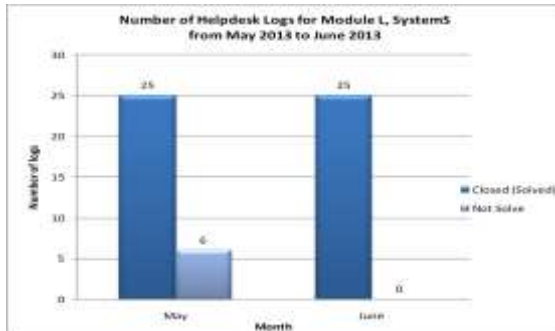

Figure 5. Bar graph for number of logs (solved and not solved) in May and June 2013

solved.

From other perspective, we use the Backlog Management Index (BMI) as a metric to manage the backlog unsolved within a month [11].

$$BMI = \frac{Number\ Of\ Log\ Closed\ During\ The\ Month}{Number\ Of\ Log\ Arrival\ During\ The\ Month} \times 100\%$$

The backlog has reduced about 19.25% as 100% has been solved in June 2013 compared to 80.65% log in May 2013. By not looking to complexity of problem, how severe the problem affecting user and other aspects, the management may assume that their software maintenance service is still under control because of small percentage of unresolved log. On the other hand, the management may assume the division has enough staff because the staffs afford to solve all logs for example in this for case study, log on June 2013 were all solved.

Q2-How long does it take to fix a problem?

In order to meet customer satisfaction, the duration taken by the maintenance team must meet the Key Performance Indicator (KPI) of the organization. In our case study, the organization does not set proper KPI or benchmark. They just set a guideline that every log must be solved as soon as possible. The user should be informed the status of what action is being taken to that log within seven working days' time. If the log is still unresolved, they should update the status to the user within another seven days and the process is applicable and repeatable until it solved. If the problem managed to be solved less or within the first seven days it is to be considered as good as possible.

To gather the information of this study, we analyze the transaction records of helpdesk log based on the category and status as previously mentioned. Table V shows number of helpdesk log based on AP category and Closed status of Module L for System S. From data retrieved, we considered that only 27 logs out of total log only solved within 7 days,

where another 11 is completed within 8 to 14 days and 12 exceeds 14 days.

However, these logs have been solved based on the log priority and level of the log complexity. Indeed, the level of skill and knowledge of maintenance team especially for programmers and software analysts play major roles to solve the issue in short time. Apart from that, the time taken to solve the problem may be affected by human mistake or due to careless. For example, there were cases where logs have been escalated to the wrong programmer or software analyst and the logs were there for such time without their knowing and not any action have been taken.

TABLE V. HELPDESK LOG BASED ON APPLICATION -PROCESS CATEGORY AND CLOSED (SOLVED) STATUS OF MODULE L FROM MAY 2013 TO JUNE 2013

Category : Application (Process)				
Log Status : Closed (Solve)				
Month\Day	<=7 days	8 days - 14 days	> 14 days	Total
May	16	5	4	25
June	11	6	8	25
Total	**27**	**11**	**12**	**50**

Q3-Where are the bottlenecks?

In this metric, we divide the bottleneck issue into two parts which are staff utilization and computer resource utilization.

Staff Utilization

Table VI shows the total number of staff in ITD department which segregate based on the workload. The staff has been allocating with specific modules to ensure the system able to work in the best condition. They are responsible to handle the issue that has been raised by the users particularly in the AP category log.

TABLE VI NUMBERS OF STAFF INVOLVE FOR MODUL L PER TOTAL NUMBER STAFFS IN MAINTENANCE TEAM BASED ON STAFF CATEGORY FOR MAY 2013 – JUNE 2013

Staff Category	Number of Staffs	Number of Staff Involved Log Module L (May 2013-June 2013)
Helpdesk	10	4
Programmer	43	1
Software Analyst	34	2
Total	**87**	**7**

We are focusing on our study to Module L which has the issue for other than 'Closed' status of log which regardless their status (N, D, P, CL or E) we consider it as not solve. Thus, Table VII represent the lack of problem that become backlog and unable to close.

TABLE VII NUMBERS OF LOG NOT SOLVED BASED ON ACTION BY MAINTENANCE SERVICE TEAM FOR MAY 2013-JUNE 2013

Month / Year 2013	Number of Not Solve Log/Action By			Total
	Helpdesk	Programmer	Software Analyst	
May	3	0	3	6
June	0	0	0	0

From the record given, we consider that the staff allocation for this module is not enough since this module is an active module and involves many users. The helpdesk team members supposed to serve all the issues on time to the particular programmer or software analyst but there is the

impediment of escalation process and caused of the late action taken. On the technical team site, they also receive status of unsolved log if the helpdesk teams forward it in fast response time. In this case, there are two software analysts that able to handle the module. In spite of that, one of them is the main analyst that understands most on the business process and technical view of this module. The second software analyst is sometimes unable to back up the module since he/she has been allocated with the main module that needs more attention with. Though, this section needs a specialized team that consist of helpdesk and technical team members. One of the objectives of this proposed point is to assist staff to manage and solve the problems in very fast respond.

Computer Resource Utilization

Computer resource utilization metric represents the actual use of system's resources. The resources for this metric are based on the computer and notebook which provided to staff that involve in Module L of System S.

Based on Table VIII, helpdesk team members, programmer and software analysts utilize the peripherals to run work in maintenance area. The computers are distributed evenly to each of the maintenance team member with a good condition and specification. The staffs are always aware on computer and notebooks maintenance as it is under ITD management procedure to ensure either the computers or notebook provided are in a good condition or needed for replacement with a new one. This maintenance procedure normally involves service from vendor.

TABLE VIII NUMBERS OF COMPUTERS AND NOTEBOOK ALLOCATION FOR STAFF

Staff Category	Number of Staffs	Number of Resources by Category	
		Personal Computer	Notebook
Helpdesk	4	4	0
Programmer	1	1	1
Software Analyst	2	2	2
Total	7	7	3

Besides, there are internal teams of technician that manage to repair or solve the technical problem of staff computer or notebook at first attempt. Additionally, there is no sharing of computers or notebook among the staff. There is less possibility that staff in maintenance team be short of resources to complete daily tasks. Thus, there is no issue or technical constraint on computer resources utilization which can affect the maintenance team from giving good services to the user.

IV. RECOMMENDATION AND CONCLUSION

As the organization never implements the proper measurement program previously, now they should have one. They might plan a measurement program that tailored to their organization behavior and needs using GQM framework. To permit the plan goes well, they must always keep data in proper way, for instance using database. At the same time they supposed to develop meaningful KPI or benchmark either related to product, process or resource to ensure they might be able to check on performance for every metrics they use.

If they would like to measure on user satisfaction, it is suggested for them to use best practice by doing survey before and after implementing measurement plan. Furthermore, in regards HD team, they should improve on the procedures because the process is interrelated with product and resource. Perhaps, they might have a better helpdesk system that can prevent some human mistake and can avoid cases that lead on work delay.

Maintenance team is such an asset to the organization, hence they should always upgrade their level of skill and knowledge to accelerate on their daily task. Organization should send them for training to ensure they are kept up to date with current technology. Finally, in respect of critical task and staff utilization, the management can restructure the maintenance team and provide them with better task specification in order to optimize resource utilization. As a concluding remark, software measurement is a kind of management tool that may support management in decision making specifically to improve the quality of software or system they develop and maintain. Measurement program should cover along software development process as the best quality software is developed so the less effort and cost at software maintenance is needed. Hence, the organization should plan the measurement well and choose right metrics to answer manager question consequently bring them toward reaching the goals.

References

[1] M.Riaz, E.Mende and E Tempero, "A Systematic Review of Software Maintainability Prediction and Metrics", Third International Symposium on Empirical Software Engineering and Measurement, 2009.

[2] M.Z. Md Nor and R.Abdullah, "A Technical Perspective of Knowledge Management in Collaborative Software Maintenance Environment", SpaceOps 2010 Conference.

[3] A.April, J.H.Hayes, A.Abran and R.Dumke, "Software Maintenance Maturity Model (SMmm): the software maintenance process model", Journal Of Software Maintenance And Evolution: Research And Practice J. Softw. Maint. Evol.: Res. Pract. 2005; 17:197–223

[4] S.Chan and P.Puxley, " Distributed User Support and the Gemini Observatory HelpDesk", Proceedings of SPIE – The International Society for Optical Engineering 4010, pp. 72-80

[5] B.C.Potgieter, J. H. Botha, and C. Lew. "Evidence that use of the ITIL framework is effective." 18th Annual conference of the national advisory committee on computing qualifications, Tauranga, NZ.2005.

[6] B.P. Lientz and E.Swanson, "Problems in application software maintenance". Communications of the ACM, Vol. 24 No.11, 1981.

[7] S.R.Schach, B.O. Jin., L.Yu, G.Z.Heller and J.Offutt, " Determining the Distribution of Maintenance Categories: Survey versus Measurement", Journal of Empirical Software Engineering, Vol 8, 2003.

[8] G.E.Stark, L.C. Kern, and C. W. Vowell. "A software metric set for program maintenance management." Journal of Systems and Software vol.24, no. 3, pp. 239-249,1994.

[9] G.E.Stark. "Measurements to Manage Software Maintenance." serena.com.[Online].Available:

http://www.serena.com/docs/agile/papers/Measurements-to-Manage-Software-Maintenance.pdf [Accessed: Oct. 30,2013]

[10] D.Galin, Software Quality Assurance: From Theory to Implementation. Pearson education, 2004.