Journal of Korea Multimedia Society Vol. 11, No. 6, JUNE 2008(pp. 891-902)

Software Performance Assessment Using Goal-Question-Metric Approach

Tariq Mahmood[†], Man-Gon Park[††]

ABSTRACT

The software once developed enters in to the 2nd phase of its useful operational life, where it performs its functions for which it has been developed. This is an important phase of software life, wherein the developers and users have to assess the performance of the software to prolong its operational life with efficiency, effectiveness and economy. Performance of the software is based on the measurement of the quality characteristics to ascertain that it is meeting or exceeding the needs of customers in terms of its functionality, usability, efficiency, reliability, availability, maintainability, portability etc. These software quality attributes defined under ISO 9126 provide an overall good framework for assessing the performance of the software. This paper describes a Software Performance Assessment (SPA) model using the Goal-Question-Metric (GQM) paradigm which has been applied to the operational software in real context with successful results.

Key words: Software Performance Assessment, Goal-Question-Metrics Approach

1. INTRODUCTION

The Software Performance Assessment (SPA) is one of the major areas of research in the Software Performance Engineering field. The performance of the software products, processes and resources is measured in all phases of software system life cycle through several methodologies and approaches using appropriate metrics.

The software development organizations or those organizations which develop in-house software for their internal or external customers especially need to deploy comprehensive software measurement system after once the software system becomes operational to ensure it fulfills its goals and objectives.

There exists extensive research base with several procedures, methodologies and approaches for software performance measurement and improvement of the processes and products during the software development life cycle phases [1-9]. The quality is ensured through use of different models and techniques under different combinations and with different software metrics. A cyclic process starting from the identification of requirements, setting performance objectives, developing, designing, coding, testing, maintenance etc. under the different categories of software modeling is used to develop the software and every stage and its related process is measured and evaluated of its performance and output. Though these approaches are also applied for performance assessment of software for the post release period, the comparative research on the Software Performance Assessment (SPA) specifically during operational life cycle of the software system is less and is mostly linked with software development life cycle processes and activities.

※ Corresponding Author : Man-Gon Park, Address : 599-1 Daeyeon-Dong, Nam-Gu, Busan 608-737, Rep. of Korea, TEL : +82-51-629-6240, FAX : +82-51-628-6155, E-mail : mpark@pknu.ac.kr

Receipt date : Dec. 30, 2007, Approval date : June 1, 2008

[†] Director, National Institute of Science & Technical Education, Islamabad, Pakistan
(E-mail : tariq_moe@yahoo.com)

[††] Dept. of Electronic, Computer and Telecommunication Engineering,Pukyong National University

The commercial and non commercial software development organizations and medium and small size organizations which develop software for specific purposes spend sizeable amount of resources to develop their software. They always need to use some assessment strategies after once the software system is deployed in the field to improve its performance and maintenance [9,16,18]. The large companies like Hewlett-Packard [19], Motorola [4], British Telecom [11], Schlumberger [17], Siemens [15], and so on [7,8,12,13] have devised their software performance measurement programs based on customer experiences and feed back but literature and related researches suggest that small and medium size organizations which develop their own software systems or groupware desperately need to devise and implement the software performance assessment program to assess the quality of their processes and resulting products [3,5,6,9,18]. Hence models and approaches based on metrics are needed and coming up to address SPA with its own specific requirements [1,2,10,14,18].

Over the past three decades, several software measurement methodologies and frameworks to assess and improve the overall organizational software development capabilities, software development processes, products and resources have been developed and practiced. Prominent among them are Quality Improvement Paradigm, Capability Maturity Model, Balanced Score Card, Goal-Question-Metric, Experience Factory, Quality Function Deployment and Total Quality Management [10]. Most of them are focused on process assessment and the pre-release period of the software system. There is thus a need to develop a framework for performance assessment of the software system that addresses assessing the performance of the software during it operational life.

The Goal-Question-Metric (GQM) methodology for software performance measurement is among the widely used practices to monitor and evaluate the performance of the software products, processes and resources [2,12,16]. It links the goals of the organization with the software measurement goals in three levels namely; defining the goals, posing questions to address the goals and identifying the metrics that answers those questions. Due to its generic nature, this methodology has been applied to all software life cycle products, processes and resources. It thus had the potential to undertake software performance assessment during the operational software system life using the selected quality characteristics and defining the appropriate metrics.

This paper presents a model for Software Performance Assessment of an operational software system based on the Goal-Question-Metric approach. This model is especially useful for undertaking performance assessment of small and medium-sized software systems developed in-house whose performance need to be assessed for quality characteristics namely; functionality, usability, efficiency and effectiveness, reliability, availability and maintainability.

The paper has been divided in to 5 sections. In section 2, the Goal-Question-Metric approach to evaluate the performance of the software products, processes and resources has been described. In section 3 the Software Performance Assessment model has been introduced with its parts and main features. Section 4 describes the application of the SPA model to assess the performance of an operational software system using the ISO 9126 software quality characteristics. Section 5 presents the conclusion.

2. THE GOAL-QUESTION-METRIC APPROACH

The Goal-Question-Metric (GQM) approach was developed in early 1980s by Dr. Victor R. Basili and his colleagues during their work at NASA Goddard Space Flight Centre for evaluating defects for a set of projects [2].

The goal question metric model views the measurement process holistically by identifying the measures on the basis of measurement goals and interpreting them upward to assess the level of achievement of the identified goals. This approach has helped many software organizations and projects to develop their software measurement plans, the eminent of which include Hewlett-Packard (HP) mentioned by [19], Motorola introduction of metrics by [4], and in Schlumberger by [17].

GQM has been defined in terms of three main levels namely conceptual, operational and quantitative in top-down fashion. The Fig.1 defines the GQM approach.

The three levels of GQM paradigms are explained below.

1) The conceptual level is the first level wherein the GQM proposes to define the goals of software measurement so that further action is taken in the development of questions and related metrics.

2) The operational level is the most important of all levels as it serves as a link in between the conceptual level and the quantitative level. It requires developing the questions that clarify and elaborate the goals and provide base for identification of metrics.

3) The quantitative level is the metric base which helps in the identification of those measures that can be analyzed and provide sufficient data whose interpretation help the organization and project authorities to observe that to what extent the software measurement goals stated in the conceptual level have been achieved.

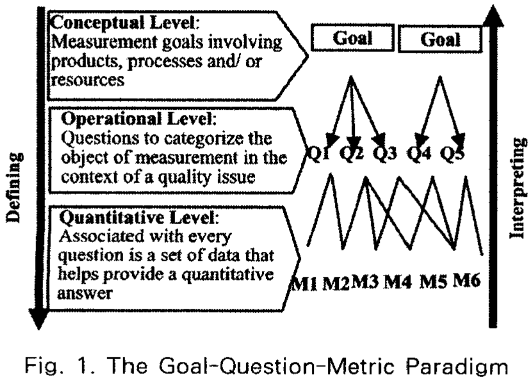

Conceptual Level:
Measurement goals involving products, processes and/ or resources
Goal
Goal
Operational Level:
Questions to categorize the object of measurement in the context of a quality issue
Q1 Q2 Q3 Q4 Q5
Quantitative Level:
Associated with every question is a set of data that helps provide a quantitative answer
M1 M2M3 M4 M5 M6
Defining
Interpreting

Fig. 1. The Goal-Question-Metric Paradigm

Table 1. Steps of GQM Implementation

Step 1: Developing the Goals	Developing goals keeping in view the quality parameters and its linkage with the organization's goal.
Step 2: Questions that define the goals	Raising questions to assess and improve the software performance with focus on the related quality characteristics.
Step 3: Identifying the metrics	Identifying metrics that answers the related question in a quantifiable way.
Step 4: Collection, and analysis of data	Devising Data Collection Forms, Obtaining metrics data, storing and analysis
Step 5: Interpretation & Documentation	Interpretation of data, documenting and reporting the results.

GQM has been implemented by the software industry in different ways depending upon the software product, process or resources to be evaluated. Comprehensive guidelines of implementing software evaluation program under GQM have been given in [16] with four main phases of planning, defining, collecting and interpreting. However, in general, the activities of GQM comprises of five steps of Table 1.

Due to its generic nature, this methodology has been applied to all software life cycle products, processes and resources. It thus had the potential to undertake software performance assessment during the operational software system life using the selected quality characteristics and defining the appropriate metrics.

3. SOFTWARE PERFORMANCE ASSESSMENT MODEL USING GQM

The SPA Model with GQM approach has thus been proposed which provides a framework to the organizations to devise the software performance assessment using the GQM methodology.

It includes three separate models each for developing software measurement goals, developing the questions and identifying the related metrics.

3.1 Developing Goals for SPA

In order to develop software measurement goals, the GQM approach provides a template which is composed of five facets of information to define what the measurement should accomplish in precise terms. Each GQM goal statement explicitly contains these facets:

1) Object: The product or process under study; e.g., a software system or end product;

2) Purpose: Motivation behind the goal (why); e.g., better understanding, assessment, control, prediction, improvement;

3) Focus: The quality attribute of the object under study (what); e.g., functionality, usability, efficiency & effectiveness etc.;

4) Viewpoint: Perspective of the goal (who's viewpoint); e.g., management, developers, or customers etc.;

5) Environment: Context or scope of the measurement program; e.g., relationship to whole program/ project of software evaluation.

The three major considerations in developing the measurement goals for software performance assessment are:

1) Development of Software Performance Assessment goal in relationship to the software quality characteristics.

2) Development of Software Performance Assessment goals linked upward to the goals of the software and the goals of the organization.

3) Development of Software Performance Assessment goals linked to the stakeholders' view.

In view of the above explanation, the following model depicted in Fig.2 is suggested for developing the software performance assessment goal.

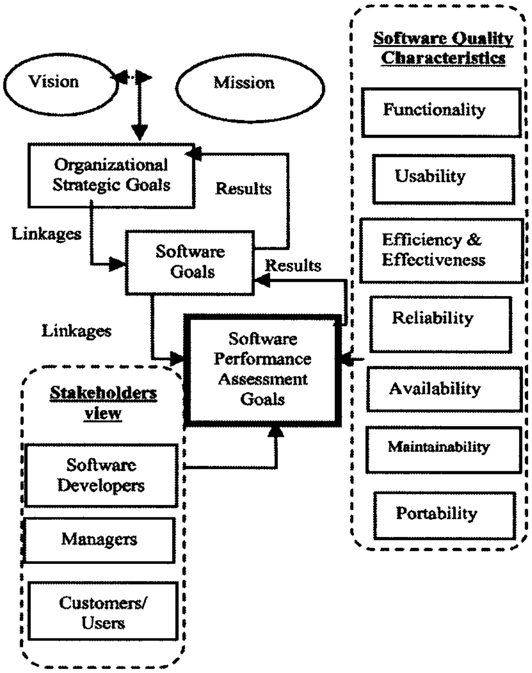


Fig. 2. Developing Goals for SPA

The organization's business is driven by their vision and mission which provides basis for formulation of goals and strategies to discharge their functions and define processes and improve their performance. Their services, products and programs reflect the organizational vision and mission and hence are linked with the organizational strategic goals. The organizational goals are thus the basis for formulation of software goals that contribute to achieve the business goals. The software goals then form the basis for the measurement goals, which are also associated with the areas of performance assessment and the stake holder's view.

3.2 Identifying Questions for SPA

Formulating questions characterize the object of measurement in the context of the quality issue from a particular view point. The model in Figure above shows the relationship of the software performance assessment goals with different areas of the performance measurement and hence the questions developed must focus on those areas and their

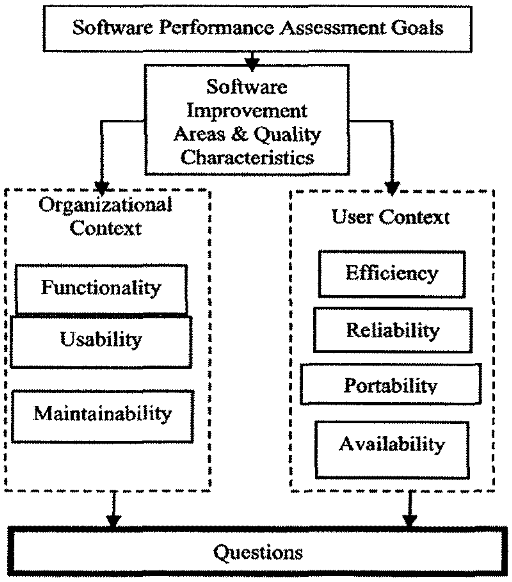


Fig. 3. Developing Questions for SPA

assessment in the organizational and user context. The following model is suggested for identifying the goals for SPA.

The above model in Fig. 3 depicts the relationship of the questions with the measurement goals and the quality aspects that need to be measured in software performance assessment. The quality aspects have been divided in to organizational and user context. The end result of this model is a set of relevant questions for their further relationship in determining the appropriate metrics. The list of quality aspects, improvement areas may vary with the type of product being measured and goals of measurement. The questions are important in software performance assessment to focus on the actual issues for deriving the appropriate measures.

3.3 Defining Software Metrics for SPA

Software metrics form the basis of measurement, as it provides data for analysis, interpretation and reporting. Care is taken for identification of proper metrics, correct procedure of collection of data and its validation, using proper tools for entry and storage and then analyzing, interpreting and reporting back to observe how much they measure the performance assessment goals.

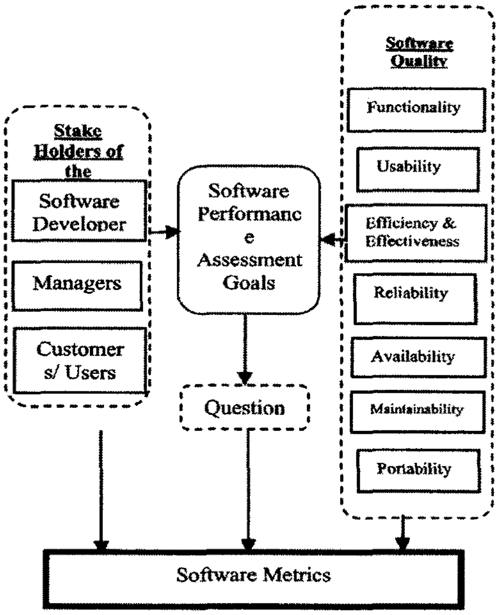


Fig. 4. Developing Software Metrics for SPA

We suggest the above model in Fig. 4 for identification of appropriate metrics base for SPA. The above software metric development model suggests that software metric base has its relationship with the stakeholder, important software performance areas under considerations and at the same its linkages with goals for final interpretation. The software metric base can be grouped in several ways like terms of objective & subjective software metrics or meta metrics, derived metrics and analytical metrics or product, process and management metrics for collection, analysis, automation and measurement.

3.4 The Integrated SPA Model

On the basis of these three constituent models developed, the complete software performance assessment model has been developed. The model connects all the three main parts of goals, questions

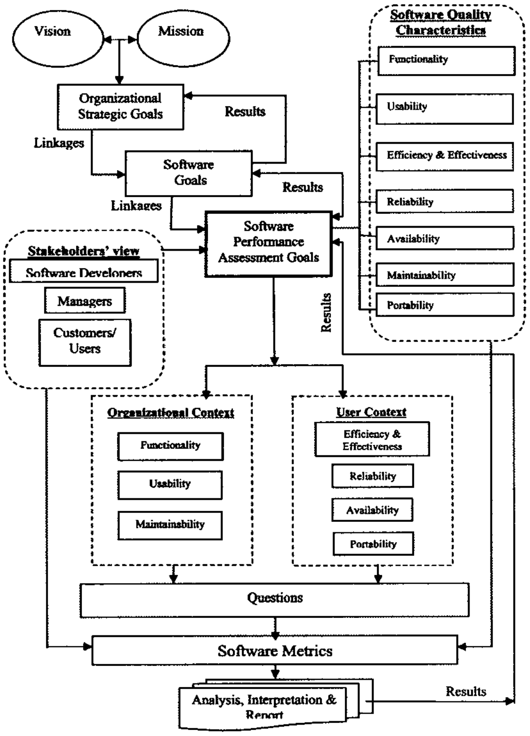


Fig. 5. Software Performance Assessment (SPA) Model

and metrics. The Fig. 5 shows the Software Performance Assessment Model.

The SPA model has the following salient features:

1) It modulates the software performance assessment process in five modules with involvement of all essential components and can be accomplished in parts with the participation of concerned people;

2) Each module has its special requirement with involvement of management and stakeholders;

3) It is simple and flexible for making changes according to the type of software and performance assessment program. The dotted boxes in the model for quality characteristics,, stakeholders' involvement, question building in user context and organizational context show that these are subject to modification by the management, software performance assessment team and addition and deletion can be made accordingly;

4) It follows the spirit of GQM in top-down approach of defining the goals, questions and metrics and bottom-up approach of interpretation of the results;

5) The model can be standardized and customized to suit the particular software systems performance assessment requirements and organizational size;

6) The model is useful in developing the overall software performance assessment program for small, medium and large sized software systems;

7) Though all the parts are interlinked, organizational/ software goals, Quality Characteristics and Stake holder's views supplements each other and jointly contribute to the development of measurement goals involving the main parties.

4. IMPLEMENTING THE SPA MODEL

In order to implement the software performance assessment model stated above, we designed the input-process-output (IPO) models for planning for implementation of SPA, for defining goals, questions and metrics, for collection and analysis of data on the identified software metrics and for interpretation of the results.

4.1 Planning for the Implementation of SPA

The main steps for planning the software performance assessment are:

1) Appointment of a project team involving the management personals, software developers, internal users and employees and the external customers.

2) Preparation of a project document incorporating the detail like organizational goals, software goals, software assessment goals, resources required, time period and tentative cost.

3) Submission and approval of the SPA project by the competent authority.

The Input-Output-Process model of the Planning process is given below in Fig. 6.

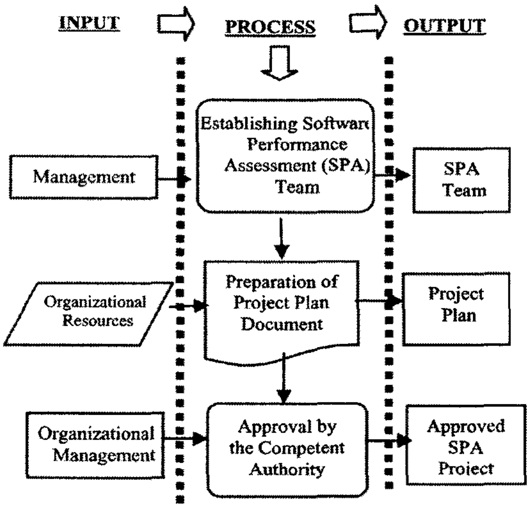

INPUT
PROCESS
OUTPUT
Establishing Softwar Performance Assessment (SPA) Team
Management
SPA Team
Preparation of Project Plan Document
Organizational Resources
Project Plan
Approval by the Competent Authority
Organizational Management
Approved SPA Project

Fig. 6. Planning for SPA

4.2 Defining Goals, Questions and Metrics for the Implementation of SPA Model

This Phase involves defining the goals of software measurement, developing the questions which clarify the goals with respect to their measurement targets and also help in the identification of the appropriate metrics and finally developing the appropriate metrics and interpreting them towards the goals of measurement. The main steps for SPA in this phase are given below.

1) Identification and grouping the software improvement areas (quality characteristics) in the context of user and organization.

2) Defining software measurement goals and its relationship to software goals and organizational goals.

3) Defining questions for software performance assessment on the basis of identified software improvement areas in user context and in organizational context.

4) Reviewing questions.

5) Defining metrics.

6) Checking and refining metrics.

7) Produce SPA measurement plan

The IPO model of this Phase is given below in Fig. 7.

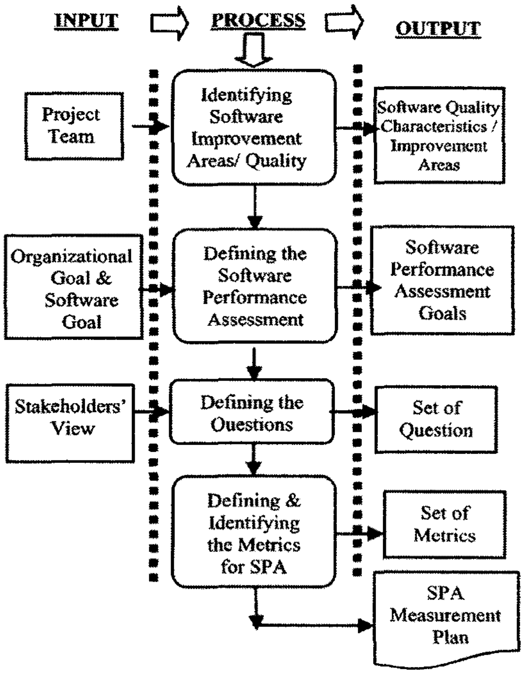

INPUT
PROCESS
OUTPUT
Project Team
Identifying Software Improvement Areas/ Quality
Software Quality Characteristics / Improvement Areas
Organizational Goal & Software Goal
Defining the Software Performance Assessment
Software Performance Assessment Goals
Stakeholders' View
Defining the Questions
Set of Question
Defining & Identifying the Metrics for SPA
Set of Metrics
SPA Measurement Plan

Fig. 7. Defining for SPA

4.3 Data Collection

There are several ways for data collection such as manual forms for data collection, electronic forms and automated data collection tools. The tools for storage and analysis also range from use of simple spread sheet to customized computerized web based measurement systems. The main steps for data collection phase in SPA are given below.

1) Designing the data collection form as per requirements of the metrics and measurement data collected on various aspects of the software improvement areas and measurement goals.

2) Collection of data manually from the users and customers who have been using the software.

3) Storing the data in prescribed software tool like spread sheet, SPSS or any other database package.

4) Analysis of the data from its raw form to graphic and tabulated forms.

The IPO model for this Phase is given below in Fig. 8.

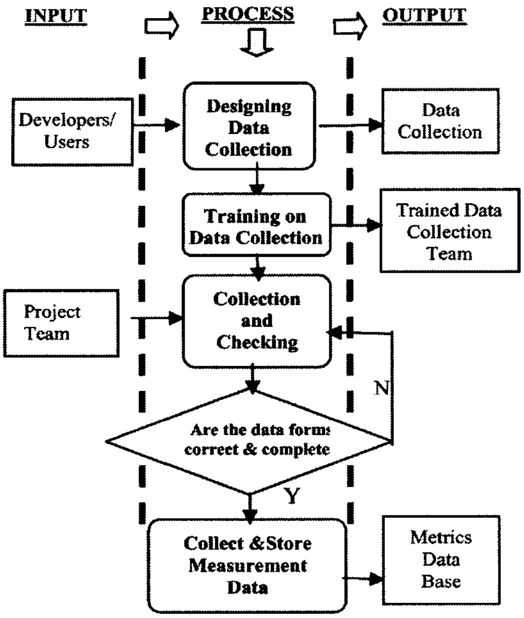

INPUT
PROCESS
OUTPUT
Developers/ Users
Designing Data Collection
Data Collection
Training on Data Collection
Trained Data Collection Team
Project Team
Collection and Checking
N
Are the data forms correct & complete
Y
Collect &Store Measurement Data
Metrics Data Base

Fig. 8. Data Collection for SPA

4.4 Interpretation of Data

The main steps for interpretation of data collected through the metrics defined in the previous phase of Software Performance Assessment are given below.

1) Interpretation of data using software application tool like spreadsheet.

2) Organizing and holding feedback sessions with all related material.

3) Finalization of software performance assessment report with all necessary attachments, results, lessons learned and recommendation for the improvement or upgrading of the software and a plan of action.

4) Formal submission of the report to the higher authorities (which approved the project) for necessary action on the implementation of the recommendations.

5) Future plan of action and allocation of necessary resources.

The IPO model for the interpretation phase is given below in Fig. 9.

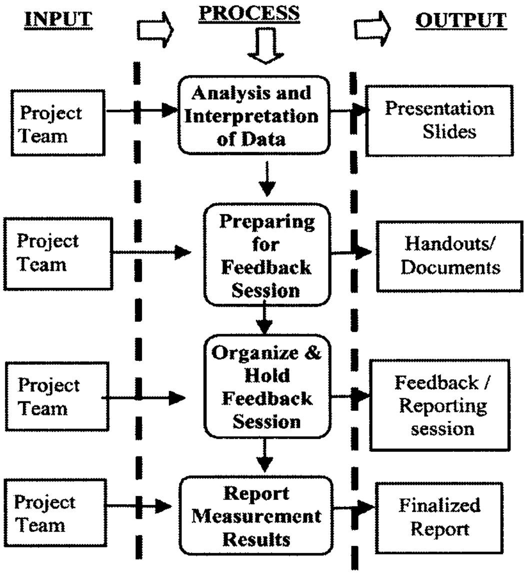

INPUT
PROCESS
OUTPUT
Project Team
Analysis and Interpretation of Data
Presentation Slides
Project Team
Preparing for Feedback Session
Handouts/ Documents
Project Team
Organize & Hold Feedback Session
Feedback / Reporting session
Project Team
Report Measurement Results
Finalized Report

Fig. 9. Interpretation of Data in SPA

The above software performance assessment models have been implemented to assess the performance of a Web-based Teaching and Learning System (WB-TLS) software. The Web-based Teaching and Learning System has been designed by an in-house team comprising of developers and the instructors by the Colombo Plan Staff College (CPSC) for technician education to use the web technology to enhance teaching and learning with cost efficiency and provide the participants a life long learning environments. The CPSC is an inter-governmental regional organization having 29 member countries in the Asia Pacific region. It conducts around 25 programs on various topics focused to country specific needs and the regional trends on Technical and Vocational Education. The training program material prepared is uploaded along with text, images and instructor's videos on the WB-TLS website. The WB-TLS performs several tasks including the interaction with the cyber instructors using e-board, submission of assignments, taking e-test, evaluation of the program, learning the program, registering online etc. The instructors have the options to manage the participants' record and test score and authorize for registration in the program and perform other administrative functions.

The WB-TLS has so far been used for around 40 training programs on different topics related to technical and vocational education. Though its evaluation is conducted after every program, but has been found insufficient to assess the performance of the software and know it weaknesses and strengths in many areas.

The software performance assessment model proposed above has been applied for the performance assessment of the WB-TLS with promising results.

Using the SPA model, the software performance assessment goals, questions and metrics were developed. The figures below show the four WB-TLS performance assessment goals, the related questions and metrics for each goal that were developed using the SPA Model.

The goal#1 was related to assessment of WB-TLS in terms of functionality with a view point of developers.

Goal #1 of WB-TLS Measurement
Assess the Performance of WB- TLS for the purpose of improvement with respect to Functionality with a view point of Developers
Questions
Will the software have the right set of functions for the tasks it must perform? Will the software produces the right agreed results or effects? Will the software adhere to domain-related standards or regulations in Law? Will the software be able to prevent un-authorized access, whether deliberate or accidental, to programs and users?
Functionality Metrics
Suitability, Accurateness, Compliance , Security

The goal, questions and metrics of goal#2 related to assessing the usability of the software in terms of users and instructors are given below.

Goal #2 of WB-TLS Measurement
Assess the Performance of WB-TLS for the purpose of improvement with respect to Usability with a view point of Users & Instructors
Questions
How much easy to understand WB-TLS? How much easy to Learn WB-TLS? How much was easy to operate WB TLS?
Usability Metrics
Understandability, Learn-ability , Operability

The goal, questions and metrics related to goal#3 for assessing the efficiency & effectiveness of the software with view point of Instructors and developers are given below.

Goal #3 of WB-TLS Measurement
Assess the Performance of WB- TLS for the purpose of improvement with respect to Efficiency & Effectiveness with a view point of Instructors & Developers
Questions
Was speed of VoDs download Satisfactory? Were resources available in successfully operating WB-TLS? How much was effective WB-TLS in Teaching and Learning?
Efficiency & Effectiveness Metrics
Time-Behavior, Resource Behavior, Effectiveness

The goal, questions and metrics related to goal#4 for assessing the reliability, availability and maintainability of the software with view point of developers are given below.

Goal #4 of WB-TLS Measurement
Assess the Performance of WB- TLS for the purpose of improvement with respect to Reliability, Availability & Maintainability with a view point of Developers
Questions
How much fault free WB-TLS operation is during the program? Is WB_TLS available anywhere and time with network connection? How much WB-TLS is maintainable?
RAM Metrics
Reliability (Maturity, Fault Tolerance, Recoverability, Availability (During & After the Program) Maintainability (Analyzability, Changeability, Stability)

The metrics data collected through the data collection forms designed for each of the quality characteristics namely functionality, usability, Efficiency and Effectiveness, Reliability, Availability and Maintainability (RAM) has been obtained and analyzed for the past three years (2004-07) programs conducted using WB-TLS. It has been found that metrics data using SPA model provided very useful information about many aspects of the WB-TLS which couldn't be realized earlier through the formal evaluation of the program being conducted using the normal summative evaluation forms. Following information on various aspects of the WB-TLS was found.

1) Every training program using WB-TLS has its unique feature due to the different title of training programs, country environments and culture and hence assessment of functionality, usability, efficiency and effectiveness provided multi dimensional information to further improve WB-TLS for subsequent programs in the same environments.

2) The quality characteristics and sub-characteristics wise information such as functionality, usability, reliability, effectiveness and their sub-components helps in understanding the strengths and weaknesses of the WB-TLS and taking actions for its further improvement in the weaker areas of performance.

3) The country-wise information over the past three years showed the behavior of the WB-TLS in the particular country in terms of the specific culture of the country, educational standards, infrastructure maintained, facilities available and language barriers. This information is very useful for further improvement of WB-TLS in the given local environments.

4) The specific information related to WB-TLS with a view point of developers such as performance of its teaching and learning, instructor-participant interaction, administration functions, and reliability, maintainability and availability functions and tasks provide insight to the developers to improve and upgrade the system in those particular areas.

5) The specific information related to the instructors such as participant's response to their textual material and videos through the e-board provide very useful information to improve their teaching, instruction and assessment methods.

6) The overall assessments through the proposed SPA model provide the management to asses the level of achievement of organizational strategic goals, the software goals and act accordingly for the improvement and upgrading of the WB-TLS for future programs and maintain the high level of quality and competitiveness.

5. CONCLUSION

The integrated SPA model presented depicts several innovative features like modularity in nature, flexibility to adjust to undertake performance assessment of any kind of software, and the stages and steps wherein the stake holder and management involvement is necessary.

The applicability of the proposed model to the Web-based Teaching and Learning Software system shows the great potential of this model. It provided us to look in to the performance of the software in different aspects of managers, developers and users. The proposed SPA model is specifically useful for undertaking performance assessment of small and medium sized software systems developed in-house whose performance need to be assessed for functionality, usability, efficiency and effectiveness, reliability, availability and maintainability. The performance assessment is a dynamic and continuous process and hence SPA model could be used continuously for the assessment of the operational software systems.

REFERENCES

[1] Baker, M.D., "Implementing an Initial Soft-

ware Metrics Program," *Proceedings of the Aerospace and Electronic Conference*, pp. 1289-1294, 1991.

[2] Basili, V., Caldiera, G. and Rombach, H. *The Goal Question Metric Approach*, Wiley & Sons Inc, 1994.

[3] Bucci, G., Campanai, M. and Cignoni, G.A., "Rapid Assessment to solicit Process Improvement in SMEs," *Proceedings of the EuroSPI' 2000 Conference*, Copenhagen, Nov. pp. 7-9, 2000.

[4] Daskalantonakis, M.K., "A Practical View of Software Measurement and Implementation Experiences within Motorola," *IEEE Transaction on Software Engineering*, Vol.18, No.11, Nov. 1992.

[5] Denvir, T., Maclennan, F. and Cooke, M., "An Approach to Software Assessment," *Proceedings of Software Engineering Standards Symposium*, pp. 89-93, 1993.

[6] Fenton, N. and Hall, T., "Implementing Effective Software Metrics Programs," *IEEE Software*, 14(2), pp. 55-64, 1997.

[7] Fuggetta, A. et al., "Applying GQM in an Industrial Software Factory," *ACM Transactions on Software Engineering and Methodology (TOSEM)*, Vol.7. No.4, pp. 411-448, 1998.

[8] Groves, L. et al., "A Survey of Software Development Practices in the New Zealand Software Industry," *Proceedings on Australian Software Engineering Conference*, p. 189, 2000.

[9] Grünbacher, P. "A Software Assessment Process for Small Software Enterprises," *Proceedings of the 23rd EUROMICRO Conference*, Budapest, Hungary, pp. 123-128., 1997.

[10] Huang, J. and Far, B.H., "Intelligent Software Measurement System for Automating the Goal-Question-Metrics Process," *Proceedings of the 18^{th} IEEE International Conference on Tools with Artificial Intelligence (ICTAI'06)*, pp. 737-742, Washington D.C., November 2006.

[11] Karkaria, D.J, Norris, M.T. and Pengelly, A.D., "Software Product Assessment," *IEEE Journal on Selected Areas in Communications*, Vol.12, No.2, pp. 271-278, Feb. 1994.

[12] Latum, F.V. et al., "Adopting GQM-Based Measurement in an Industrial Environment," *IEEE Software*, Vol.15, No.1, pp. 78-86, 1998.

[13] Mäkinen, T., Varkoi, T. and Lepasaar, M. "A Detailed Process Assessment Method for Software SMEs," *Proceedings of 7th European Software Process Improvement Conference (EuroSPI)*, pp. 1-26, Copenhagen, Denmark, Nov. 2000.

[14] Ould, M. A., "Software Quality Improvement through Process Assessment-A view from the UK", *IEEE Colloquium on Software Quality*, 1992.

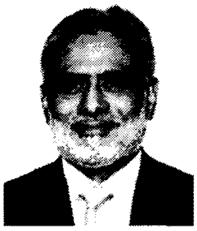
[15] Paulish, D.J. and Carleton, A.D. "Case Studies of Software-Process-Improvement Measurement," *Computer, IEEE Computer Society*, Vol.27, No.9, pp. 50-57, 1994.

[16] Solingen, R.V. and Gerghout, E., *The Goal/Question/Metric Method: A Practical Guide for Quality Improvement of Software Development*, McGraw-Hill Companies, London, 1999.

[17] Solingen, R., and Berghout, E., "Integrating Goal-Oriented Measurement in Industrial Software Engineering: Industrial Experiences with and Additions to the Goal/Question/Metrics (GQM) Method," *Proceeding of the seventh International Software Metrics Symposium*, pp. 246-258, London, UK, Apr. 2001.

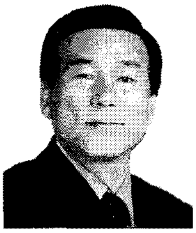
[18] Wangenheim, C.G., Anacleto, A. and Salviano, C.F., "Helping Small Companies Assess Software Processes," *IEEE Software*, 23(1), pp. 91-98, 2006.

[19] Wang, Y., and He, Q., "A Practical Methodology for Measurement Deployment in GQM," *Proceedings of the Canadian Conference on Electrical and Computer Engineering (IEEE CCECE 2003)*, Montreal, Canada, IEEE Computer Society, Vol.2, pp. 1329-1332, May 4-7, 2003.



Dr. Tariq Mahmood

Dr. Tariq Mahmood is currently working as Director Research at National Institute of Science & Technical Education, Ministry of Education Islamabad, Pakistan. Previously he worked as the Faculty Consultant at Colombo Plan Staff College for Technician Education Manila Philippines from July 2002 to October 2005. As an Educational Adviser, He has been dealing with Science, Technical and Computer Education in the Ministry of Education, Islamabad at the national level for the past 23 years. He acquired M.Sc in Mathematics from Punjab University, Lahore Pakistan, M.A. in Computers in Education from King's College, University of London, UK and Ph.D Information Systems from Pukyong National University, Busan, Republic of Korea. His main areas of expertise are Software Performance Assessment, Project Management, and Technology Management. He developed, managed and implemented several projects in science, computer and technical and vocational education in collaboration with international donor agencies like Asian Development Bank, World Bank and UNESCO in Pakistan.



Dr. Man-Gon Park

Dr. Man-Gon Park is serving as a Professor of the Division of Computer and Multimedia Engineering, College of Engineering, at the Pukyong National University (PKNU), Republic of Korea since 1981. Also he is the president of the Korea Multimedia Society (KMMS) from 2008. He supervises Graduate Students in Software Engineering and Multimedia Information Systems. Dr. Park served as the Director General and CEO of the Colombo Plan Staff College for Technician Education (CPSC) from 2002-2007, which is an intergovernmental international organization of 29 member countries for Human Resources Development in Asia and the Pacific Region. He served there also as a Faculty Consultant seconded by the Government of the Rep. of Korea as an expert in information systems development and ICT-based TEVT systems from 1997-2001. He has been the visiting professor at the Department of Computer Science, University of Liverpool, UK; exchange professor at the Department of Electrical and Computer Engineering, University of Kansas, USA; and visiting scholar at the School of Computers and information science, University of South Australia. He was dispatched to Mongolia and People's Rep. of China by KOICA on various projects as information systems consultant. He has also embarked on consulting works and conducted training programs in ICT on individual capacity for Korean groups of companies, governmental and non-governmental agencies and other institutions in Korea. His main areas of research are software reliability engineering, business process reengineering, Internet and web technology, multimedia information processing technology, quality management systems, and ICT-based human resources development.