Malaysian Journal of Computer Science, Vol. 11 No. 2, December 1998, pp. 52-60

AN OVERVIEW OF THE DEVELOPMENT OF A COMPETENCY MEASUREMENT SUPPORT TOOL – *Qual**Pro**-CM*

Siew Hock Ow and Mashkuri Hj. Yaacob
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
Malaysia
Tel.: 603-7696366
Fax: 603-7579249
email: show@fsktm.um.edu.my
mashkuri@fsktm.um.edu.my

ABSTRACT

Software quality is an important issue in the software development industry. There are three elements that determine the quality of a software system: people, management and technology. Of these three elements, people constitute the most crucial element that determines a successful software outcome. The competency of the people involved in software quality assurance is important. However, there is a lack of appropriate competency measurement models and tools that can be applied for such purpose. This paper highlights the establishment of an indigenous model, the Knowledge, Awareness and Practice (KAP) Model, and the development of an automated competency measurement support tool, QualPro-CM, focusing on the functions and modules as well as its design philosophy. This tool is developed based on the client/server architecture incorporating the use of Web-based Microsoft development tools.

Keywords: ***Software Development; Knowledge, Awareness and Practice Model; KAP Model; Competency Measurement Support Tool; QualPro-CM; Client/Server Architecture; Software Quality Assurance; SQA.***

1.0 INTRODUCTION

Software quality assurance (SQA) is an important issue in software development. The focus of quality assurance is to achieve a ‘right first time’ approach, thus reducing the rate and associated costs of failure through poor products and services [1]. It is determined by the three elements of people, management and technology [2]. Of these, people have been recognised as the most important since they manage and control the software development process [3]. Today’s software systems are as complicated as ever, and large software development projects become generally a team-based effort. Under such circumstances, the success of a software project is highly dependent on the development team. However, the completion of a software system in itself does not necessarily imply the production of a *quality* software system. The developed system risks becoming a shelfware if it is unable to fulfill the user’s requirements completely. It becomes logical that quality be designed into the software development process. It is imperative that the SQA team performs quality assurance (QA) activities such as simulation, review, walkthrough and audit, at every phase of software development to enhance the prospect of a successful software outcome [4, 5]. The quality activities have to be conducted by the right people who are highly conversant in the various aspects of SQA. One way to evaluate a person’s ability is to measure his competency level based on some established methods or a competency model. A scan of the literature on competency measurement reveals that various types of approaches and tests have been established for this purpose. For example, IBM’s Programming Aptitude Test (PAT) is an eligibility test to determine the right candidate for the post of programmer [6]. However, there is a general lack of tests or measurement models that can be used for evaluating the competency level of an SQA team.

A research project was initiated in the Faculty of Computer Science and Information Technology, University of Malaya, to develop a competency model and an associated automated tool to support the measurement process. Resulting from the research, an indigenous competency model known as Knowledge, Awareness and Practice (KAP) Model, can now be adopted to measure the competency level of an SQA team prior to it being assigned any SQA jobs. In this model, only three attributes namely, knowledge, awareness and practice, are taken into consideration. Other attributes such as experience, the condition of the working place, and sociological factors are not taken into account, as their inclusion would make the study very complicated. Furthermore, the outcome of the measurement of these attributes would not be able to reflect a solid and significant interpretation for the intended measurement. However, the impact of these attributes will be analysed using the KAP Model in future studies. Since every employee needs to possess sufficient knowledge in order to perform a job, knowledge is thus selected for measurement. Measuring the attributes relating to awareness and practice is intended not only to study employees' awareness of new developments and technologies that are impacting on the software industry, but also whether they possess good/positive attitude towards embracing new changes and technologies in their jobs.

2.0 THE KAP MODEL

The measurement of employee competency using the KAP Model is conducted using a set of questionnaire. The questions are formulated and categorised into three main sections: knowledge, awareness and attitude. Since this model aims to measure the competency level of an SQA team, the questions formulated are based on the SQA perspectives. The competency score of each team member is calculated based on the answers given in the three sections. The calculation is performed using the following steps:

a. Calculate the score of each question, S_Q

$$\boxed{\mathbf{S_Q = P_Q \times W_Q}} \qquad \text{for } Q = 1,2, \ldots.., M$$

Where, Q: Question number
M: Total number of questions in a section
P_Q: Points for a correct answer for a question
W_Q: Weightage of a question
(The default values for point, P_Q and weightage, W_Q is 1, specify otherwise)

b. Calculate the score of each section, S_S

$$\boxed{\mathbf{S_S = \left(\sum_{N=1}^{N=M} S_Q\right) \times W_S}} \qquad \text{for } N = 1, 2, .., M$$

Where, M: Total number of questions in a section
S: Section number
W_S: Weightage of section S
(The default value for weightage, W_S is 1, specify otherwise)

c. Calculate the total score of all sections, S_T

$$\boxed{\mathbf{S_T = \sum_{N=1}^{N=S} S_S}} \qquad \text{for } N = 1, 2, \ldots.., S$$

Where, S: Total number of sections
(In this case, S = 3)

d. Calculate the competency score, CS

$$\boxed{\mathbf{CS = (S_E / S_T) \times 100\%}}$$

Where, S_E: Total score obtained by employee (using steps a-c above)
S_T: Total score of all sections

Each question is assigned one point for correct answer and a weightage of one. This manner of points allocation, assigns equal importance to each question as well as the three attributes. However, the model allows some degree of flexibility to suit various organisational needs. For example, to study whether knowledge is more important than the other two attributes, the weightage for knowledge can be assigned 2 or a higher value to indicate the degree of importance. Similarly, if an evaluator finds that a question is more important than any other questions, he can assign a higher point for that question.

From the competency score obtained, the grade and competency level are then determined based on a grading scheme established by the evaluator. The grading scheme can be changed or adjusted, as the evaluator deems fit. Depending on the competency level attained, varying levels of training are recommended. Table 1 and Fig. 1 show examples of the grading scheme and training needs for specific competency levels involving eligibility to conduct a software requirements review process. The identification of training needs is to institute appropriate training on the job. According to W. Edwards Deming, there must be a commitment to continuing training and education by the software managers and professional staff, as knowledge is important for improvement [7, 8]. This would enable the employees to be always at the forefront of technology and hence to attain quality output.

Without an automated tool, the competency scores as well as the analysis of the results obtained as outlined above, would have to be done manually. This would be extremely tedious, time consuming and prone to error if it involves a large number of SQA team members. Hence, an automated tool, *Q*ual*P*ro-CM, was developed to support the measurement and the analysis process.

3.0 THE FUNCTIONS AND FEATURES OF *Q*ual*P*ro-CM

*Q*ual*P*ro-CM is custom-built to measure competency level of an SQA team. The measurement process begins with the design of a questionnaire, and followed by answering of the questionnaire by the SQA team members, calculation of competency scores and analysis of results as shown in Fig. 2. Its development methodology is both method-oriented and process-oriented. The main functions of *Q*ual*P*ro-CM are defined according to the process flow as explained and illustrated below.

a. Input Questions

A set of questions is created and stored in a database to be selected for use in the design of the questionnaire. Basically, there are three types of question. The first type comprises objective questions requiring an either a *yes*, *no* or *not applicable* answer. The second type comprises questions with various options as the answers with a check box against each option. Choice of one or more answers is allowed. The third type requires text input as answers to elicit opinions or comments. In addition, three other types of questions are included. These consist of questions which are combinations of one or more of the three basic types. Of the three basic types, only the objective type is used in the calculation of the competency score. The other two types provide additional information for the final analysis of results. Indeed, if the input questions are related to software development perspectives, the measurement of employee competency for the software development team can also be conducted using the KAP Model.

b. Design Questionnaire

In designing a questionnaire, relevant questions are selected from the pools of questions in the database. The questionnaire is divided into different sections. The questions for each section are selected according to their category. When designing the questionnaire, the point and weightage are also assigned for each question and section.

c. Define Grading Scheme

The evaluator or a group of evaluators decide on the competency levels that determine an employee eligibility to participate in an SQA activity such as a review process. These levels are input into the system and can be changed or reviewed from time to time. The training needs for each competency level are also entered into the system.

d. Answering the Questionnaire

Each SQA team member is notified through the *send mail* function of *Q*ual*P*ro-CM of the directory in the network server where the questionnaire is located, and the date and time to answer the questionnaire. A time limit of between 20-30 minutes is given to answer all the questions in a section. If an employee has not answered the section on the specific date and time, he will be denied access to the questions, and a new date and time will have to be set for him to complete that specific section. Also, once the employee has submitted his answers to the system for calculation, he would be denied further access to the questions.

e. Calculation of Competency Scores

After the employee has answered all the questions in a section and submitted it to the system, *Q*ual*P*ro-CM will calculate the scores and store the results in a database. The scores in the database include the section scores and the competency score.

f. Analysis of Results

*Q*ual*P*ro-CM performs analysis of results by comparing the competency score obtained with the competency level defined by the evaluators. The training needs as suggested by the employee in the feedback section of the questionnaire as well as from the evaluators, are also generated. The evaluators can view these results, but not the employee. If there is a need for an employee to know his section scores or competency score, the evaluators can inform him using the *send mail* function of *Q*ual*P*ro-CM.

Table 1: Grading Scheme and Training Needs

Grade	Competency Level	Competency Score (%)	Training Needs	Eligibility to conduct review
A	High	91-100	1. New/updated standards/guidelines to conduct a requirements review process (RRP) 2. New methodologies and tools used in an RRP 3. Improved/revised review process	Yes
B	Average	71-90		
C	Fairly Low	51-70	1. Quality factors of SRS 2. Criteria of quality factors 3. Role/responsibilities of review team members 4. Standards/guidelines to conduct an RRP	No
D	Low	0-50		

D (0-50)	C (51-70)	B (71-90)	A (91-100)
Not eligible to conduct requirements review process (RRP)		Eligible to conduct RRP	

Fig. 1: Grading scheme and eligibility to conduct an RRP

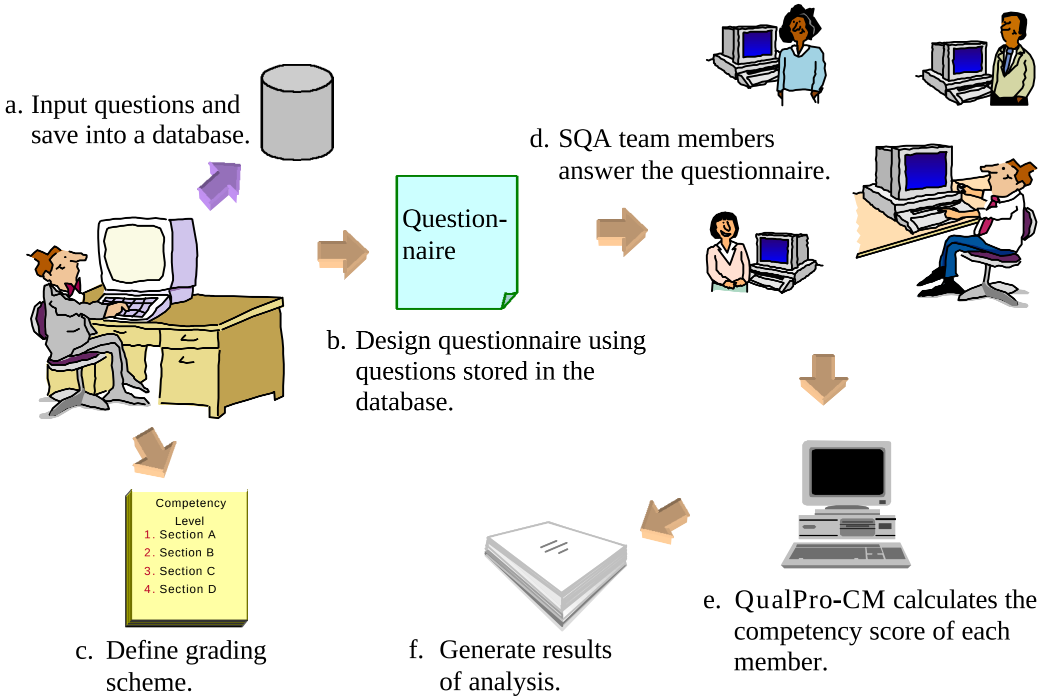
a. Input questions and save into a database.
Question-naire
b. Design questionnaire using questions stored in the database.
d. SQA team members answer the questionnaire.
Competency Level
1. Section A
2. Section B
3. Section C
4. Section D
c. Define grading scheme.
f. Generate results of analysis.
e. QualPro-CM calculates the competency score of each member.

Fig. 2: The process flow of Competency Measurement

4.0 THE DEVELOPMENT OF *Qual*Pro-CM

The tremendous progress in Information Technology (IT) has made it possible for much of the work in software development to be distributed to desktops, laptops, network servers, wide area networks, and global web servers. More importantly, information driven through the IT systems has gained recognition as a primary corporate asset, that is crucial to building and maintaining a competitive advantage. Although the client/server environment is still relatively new, many IT professionals, especially the software project developers, are getting aware of the advantages of the transition to the Graphical User Interface (GUI), iterative development, relational database and open system environment. Indeed, it is faster, easier, and more flexible to develop and deliver applications in the client/server environment [9]. QualPro-CM was thus developed based on the client/server architecture. The following sections highlight the development process of QualPro-CM focusing on the identification of modules, use of appropriate development tools and technologies, its high level design and concept of implementation.

4.1 The Modules of *Qual*Pro-CM

QualPro-CM consists of four main modules namely, *User Administration*, *Competency Measurement*, *Change Password* and *Logout*. The *User Administration* module provides a set of functions that allow a system administrator to manage new and existing users (i.e. employees). The functions and operations performed by this module are listed in Table 2. The *Competency Measurement* module enables an evaluator to maintain the competency measurement for a specific SQA activity, input questions, design questionnaire, perform competency measurement, define the grading scheme and competency levels, analyse the results and to generate various types of reports for management use as shown in Fig. 3. The functions and operations performed by this module are similar to the *SQA Team Administration* module and these include: to add new measurement, edit, delete and view measurement for the *maintain competency measurement* function, to add new questions, edit, delete, and view questions for the *input questions* function, etc. The *Change Password* module enables the user to change the password. The module *Logout* allows the user to exit from QualPro-CM.

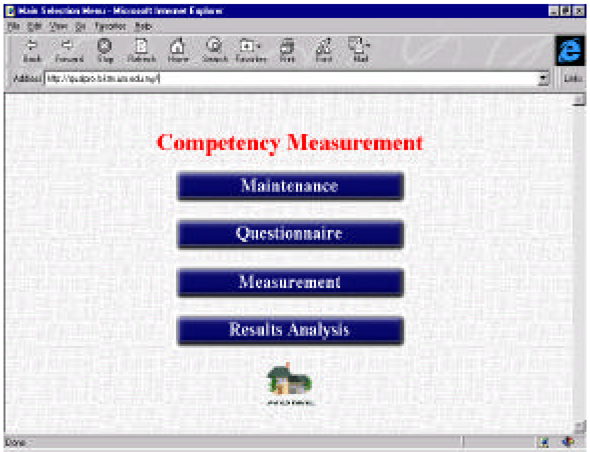
Main Selection Menu - Microsoft Internet Explorer
File Edit View Go Favorites Help
Back Forward Stop Refresh Home Search Favorites Print Font Mail
Address
Links
Competency Measurement
Maintenance
Questionnaire
Measurement
Results Analysis
Done

Fig. 3: The main functions of the Competency Measurement Module

4.2 The Tools and Technologies Used

The development of *Q*ual*P*ro-CM involved the use of a set of Web-based development tools. These tools comprise a set of Microsoft products such as Internet Information Server 3.0 (IIS 3.0), Active Server Pages (ASP), Microsoft Transaction Server 1.0 (MTS 1.0), and SQL Server 6.5, Visual InterDev 1.0, Visual Basic Enterprise 5.0, Visual Basic Scripting Edition (VBScript), and ActiveX [10, 11, 12, 13].

Needless to say, there are various Internet development technologies to choose from to develop *Q*ual*P*ro-CM. Among the main contenders are IBM's offering of its Lotus Domino, Netscape's Enterprise Server, and others. However, there are definite advantages for adopting the Microsoft technologies. Firstly, *Q*ual*P*ro-CM was developed for application in both the Internet and Intranet environments and appropriately selected Microsoft Windows NT as its operating system. The latter has proven stability and provides good security features which are very essential for the type of network environment in which *Q*ual*P*ro-CM is applied. The choice of Windows NT itself brings some benefits in terms of reduced development costs, as the essential tools used, IIS and MTS, both come packaged with the system software itself and need not be separately purchased. Secondly, the use of a suite of other Microsoft software development products such as ASP, Visual Basic 5.0, SQL statements, etc., dispenses with the need to learn a variety of new programming languages in order to integrate all the tiers in the client/server architecture on which *Q*ual*P*ro-CM is based. For instance, the SQL statements, HTML, Visual Basic 5.0 and VBScript used in the programming of *Q*ual*P*ro-CM, all use a similar syntax. This helps in the writing of more efficient and clearly understandable codes, and thus, facilitates easier code maintenance. As such, all these translate into a shorter development time and invariably contribute also to cost reduction in development. The use of Microsoft development tools solely in the present implementation of *Q*ual*P*ro-CM, is not without its drawback. It restricts access to it only to the use of Microsoft Internet Explorer 3.02 or higher version of the browser, in order to appreciate its full functionality. However, this problem can easily be surmounted by replacing all the VBScript (client scripting), currently in ASP pages, with JavaScript. This will render it a more open system to be able to run on different browsers and platforms.

Table 2: The functions and operations of the *SQA Team Administration* Module

Function Name	**Operation**
Add New	• register a new user • set user accessible modules • create password for user
Edit/Delete/ View	• edit user's information and module access rights • delete a registered user • view all registered user
Send Mail	• send mail to a user • send mail to a group of users • configure POP server of the system

4.3 The Design of *Q*ual*P*ro-CM

*Q*ual*P*ro-CM shows a holistic Web development design comprising applications at both the server and the client ends. Web applications tend to extend beyond pure HyperText Markup Language (HTML), Web client, and Web server. There are many issues that need to be considered, such as component design, development languages, database access, personalisation, and security.

*Q*ual*P*ro-CM was developed based on a three-tiered architecture. Tier 1 is the presentation layer, Tier 2 is the business rules/logic layer, and Tier 3 is the data management layer. The presentation layer furnishes the user interface for the overall system. It permits the end users to enter and manipulate data, analyse information, and navigate throughout the system. Validation and verification (V&V) logic is performed in this layer to edit input values and selections. The business rules/logic layer processes decisions, policy enforcement and resource management decisions that ascribe to the data, state and event-processor models. The data management layer maintains consistent and secures data through the enforcement of referential integrity and data security. Transactions (i.e. operations) and queries against the database such as accessing, updating, inserting, locking, retrieving, security, and backup are managed in this layer [14, 15].

All data entered into *Q*ual*P*ro-CM is stored in a relational database in a collection of tables. Each transaction performed on the table is displayed in the browser as a Web page or a form. For example, if a user selects the *add new* operation of the *input questions* function of the *Competency Measurement* module, the client's browser would first request a page from the Internet Information Server (IIS) through the virtual directory (i.e. the directory where the Internet surfer uses to access those files). The IIS gets an ASP from the local directory (i.e. the directory where files are stored in the server's hard disk), processes those server code (i.e. those necessary server process) at IIS, and then sends the results to the browser in HTML format to insert/fill all the input boxes of the form, and then submit to the IIS. Just before submitting the user input form to the server, a validation process (if any) will verify all the valid users' input by using client-site scripting. After that, the form will be sent to the server. The server will invoke server-site scripting (i.e. ASP code) to collect those inputs and do some server-site validation as well (if any).

After the client-site and server-site validations, all the inputs will be sent to the business object (BO) which is created using Visual Basic 5.0 (VB5) and compiled as an ActiveX object. At BO, it will get the context from Microsoft Transaction Server (MTS) in order to invoke the transaction object (all business objects created by VB5 are installed in MTS). All the inputs from ASP will bind (i.e. the mechanism by which a client associates itself with a server) into a collection, followed by some processing (if any) before the transaction. Then, the BO will invoke transaction object by supplying an operation code of the transaction and the input collection. The operation code is a predefined Structured Query Language (SQL) statement template for a particular transaction.

When the transaction object is being invoked, it will be based on the operation code and get the SQL statement or command template from SQL server. It will then pass to the message binding (Msg Binding) object to bind the command template with the input collection. Message binder object will bind the command template with those inputs in the collection based on their data types. The output of the message binder is an executable SQL command for the transaction. The transmission object is responsible to the 'traffic' between MTS and SQL server. It will send the executable SQL command from the message binder to the database (SQL server) and receive any data from the database. The protocol between these processes is Open Database Connectivity (ODBC) [16]. Fig. 4 illustrates the process flow of a client's request to the server in the three-tiered architecture. Fig. 5 and Fig. 6 depict the Active Server Pages (ASP) flow of the *Add New*/*Edit*/*Delete* and *View* operations. The explanations of these operations are given in pseudo-code.

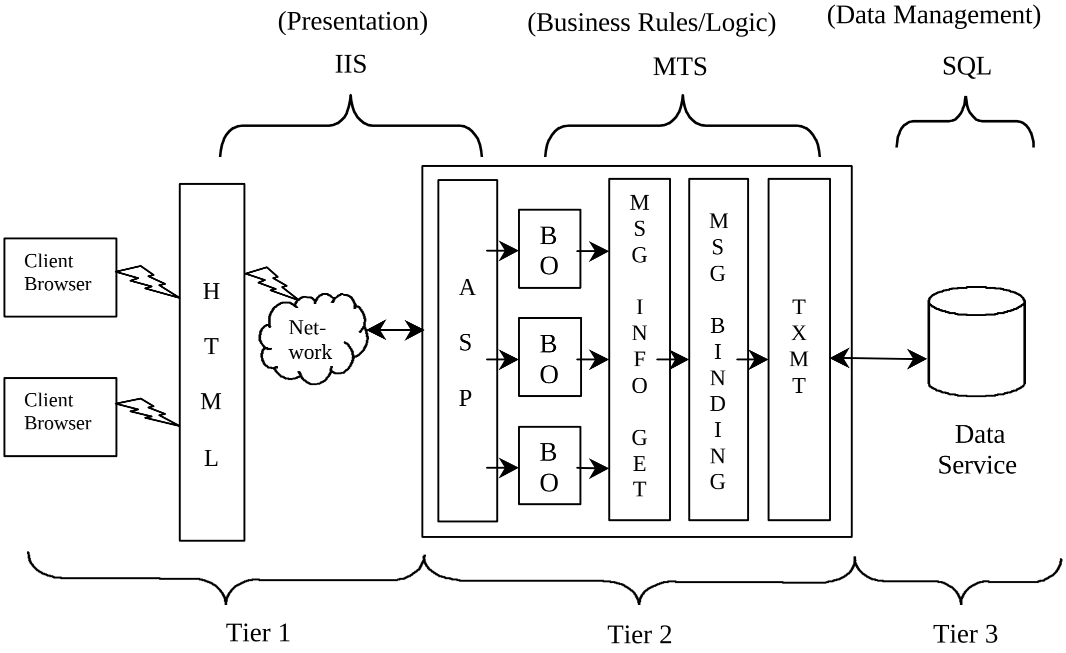

Fig. 4: The three-tiered architecture of *Q*ual*P*ro-CM

a) Add New/Edit/Delete Pages

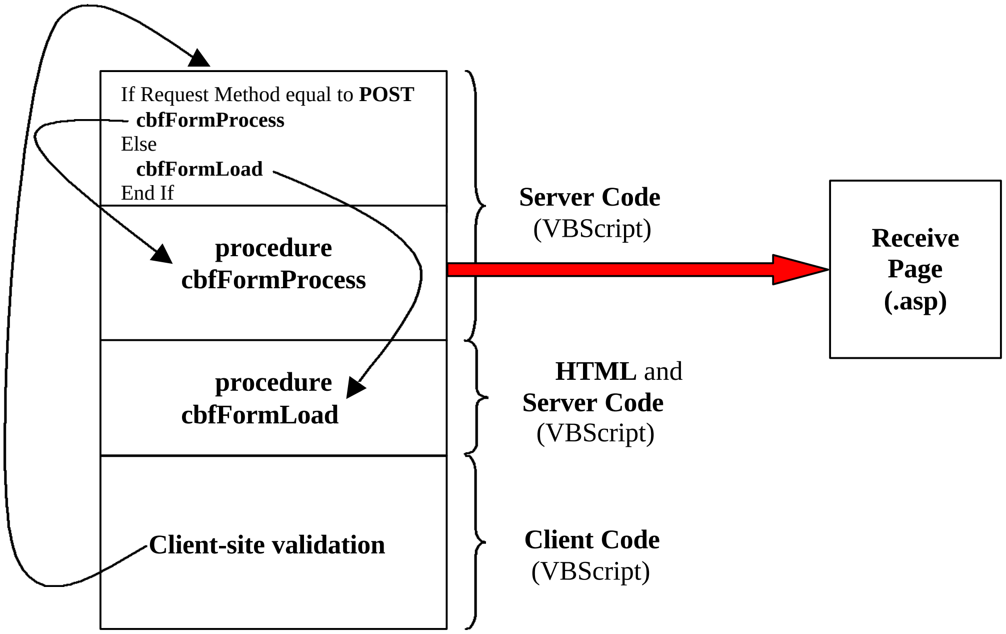

Fig. 5: Active Server Pages (ASP) flow for the *Add New*/*Edit*/*Delete* Operations

Pseudo-code for the *Add New*/*Edit*/*Delete* Operations:

```
1.      Check request method
        1.1     If Request Method equal to POST
                1.1.1   Form processing/process business transaction
                        (invoke procedure cbfFormProcess)
        1.2     Else (If Request Method equal to GET)
                1.2.1   Format/display a form (invoke procedure cbfFormLoad)

1.1.1       Form processing/process business transaction: cbfFormProcess
1.1.1.1     Perform server-site validation (e.g. date validation or additional server site processing)
1.1.1.2     Input collection and then send to Business Object
1.1.1.3     Perform error checking
1.1.1.3.1   If success, redirect the page to a receive page
1.1.1.3.2   Else, display error message and invoke procedure cbfFormLoad

1.2.1       Format/display a form: cbfFormLoad
1.2.1.1     Display all detailed data and/or user's input into a form
1.2.1.2     Perform client-site validation
1.2.1.3     Send the form to itself (reload its own page) for further processing
```

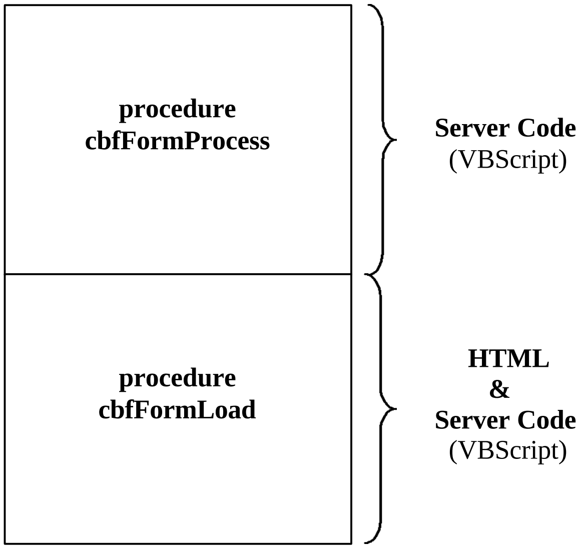

Fig. 6: Active Server Pages (ASP) flow for the *View* Operation

Pseudo-code for the *View* Operation:

```
1.    Send a transaction/request to SQL server through Business Object
2.    Receive results from Business Object
3.    Form/Display result in a table format
```

All the functions performed by *Q*ual*P*ro-CM are executed in this format iteratively until the user logout from the system. This design methodology has contributed to a structural, flexible and easy to maintain coding process.

5.0 CONCLUSION

Today's software systems are large and sophisticated and generally require team-based effort. Under this circumstance, it is inevitable that *people* who remain largely in control of the system design and its outcome, are the main determinant of quality software products. Involving people with the right skills and injecting quality assurance into every phase of software development become important issues. While various development tools are available to aid in the development process, there has been a lack in the availability of tools to aid in the selection of the most competent personnel for the same development purpose. *Q*ual*P*ro-CM addresses this problem, and as an automated measurement tool, it will certainly help in the selection of competent personnel based on the KAP Model to ensure a quality software outcome.

The trend and techniques in software development have undergone tremendous change. This is most notable in software reusability and Internet computing [17]. The latter, in particular, has made it possible for software to be distributed or dynamically shared as Web-based applications over the Internet. Based on this philosophy, *Q*ual*P*ro-CM was developed for the client/server environment and exploited the use of Web-based development tools to enhance its features and functionalities. As a Web-based application, there is unrestricted time and access to the *Q*ual*P*ro-CM from within and outside an organisation. Hence, the competency of an employee can be evaluated without the constraint of time and location.

The present method-oriented development methodology commonly applied in the development of CASE tools [18], is no longer adequate to meet today's development requirements. Rather, a combination of method and process oriented techniques would be more suitable to ensure the production of quality software systems. *Q*ual*P*ro-CM, developed based on the KAP Model (method-oriented), and incorporates the measurement process flow (process-oriented), would certainly make it a fully automated, flexible and powerful tool for the measurement of employee competency.

REFERENCES

[1] S. Fletcher, *Quality and Competence*. London, Kogan Page Limited, 1993.

[2] R. H. Dunn, *Software Quality: Concepts and Plans*. Englewood Cliffs, N.J. Prentice-Hall, Inc., 1990.

[3] B. Curtis, et al. A Field Study of the Software Design Process for Large Systems, *IEEE Transactions on Software Engineering,* Vol. 31 (11), 1998, pp. 1268-1287.

[4] A. Jarvis, and V. Crandall, *Inroads to Software Quality: "How To" Guide and Toolkit*. Upper Saddle River, N.J. Prentice-Hall, Inc., 1997.

[5] C. P. Hollocker, *Software Reviews and Audit Handbook*. New York, John Wiley & Sons, Inc., 1990.

[6] T. J. Swanson, and J. J. Devore, A Structured Approach To Hiring, in D. J. Reifer, Tutorial: Software Management. 3rd Edition. Washington, D.C. IEEE Computer Society Press, 1986, pp. 217-220.

[7] L. J. Arthur, *Improving Software Quality: An Insider's Guide to TQM*. New York, John Wiley & Sons, Inc., 1993.

[8] G. G. Schulmeyer, and J. I. McManus, (eds.), *Handbook of Software Quality Assurance*. 2nd Edition. New York, Van Nostrand Reinhold, 1992.

[9] L. L. Barnes, and D. J. Shimberg, *Client/Server & Beyond: Strategies for the 21st Century*. Upper Saddle River, N.J. Prentice-Hall, Inc. 1990.

[10] *http://www.microsoft.com/products/internet.html*. Microsoft Corporation, 1998.

[11] K. Brophy, and T. Koets, *Teach Yourself VBScript in 21 Days*. Indianapolis, Sams Net Publishing, 1996.

[12] M. Corning, et al. *Working with Active Server Pages*. Indianapolis, QUE Corporation, 1997.

[13] K. Reichard, *The Windows NT 4.0 Internet Server*. New York, M&T Books, 1997.

[14] N. Jenkins, et al. *Client/Server Unleashed*. Indianapolis, Sams Publishing, 1996.

[15] D. Vaskevitch, *Client/Server Strategies: A Survival Guide for Corporate Reengineers*. 2nd Edition. Foster City, CA, IDG Books Worldwide, Inc., 1995.

[16] P. E. Renaud, *Introduction to Client/Server Systems: A Practical Guide for Systems Professionals*. 2nd Edition. New York, John Wiley & Sons, Inc., 1996.

[17] W. T. Tsai, Trends in Software Technology. *Proceedings of the Twenty-First Annual International Computer Software & Applications Conference (COMPSAC'97), August 11-15, 1997, Washington, D.C.* Los Alamitos, IEEE Computer Society, 1997, pp. 173-174.

[18] S. Jarzabek, and R. Huang, *The Case for User-Centered CASE Tools*. Technical Report TRB9/96. Singapore, Department of Information Systems and Computer Science, The National University of Singapore, 1996.

BIOGRAPHY

Siew Hock Ow received her M. Comp. Sc from University of Malaya in 1989. She has been a lecturer at the Faculty of Computer Science and Information Technology, University of Malaya since 1992. Currently, she is pursuing her Ph.D in software engineering. She has published a number of research papers in computer science journals and has also presented papers at both local and international conferences. Her research interests include software quality assurance, CASE tools, computer-aided learning and human-computer interaction. She is a member of the New York Academy of Sciences, affiliate member of IEEE Computer Society and Malaysian National Computer Confederation (MNCC).

Mashkuri Yaacob is Professor of Computer Science and Dean of the Faculty of Computer Science and Information Technology, University of Malaya. He joined the University of Malaya in 1976. He has published over 130 research papers and presented papers at both local and international conferences. He is a member of the IASTED Conference Organising Committee and IEEE Computer Society. His research interests are software engineering, and computer architecture which includes ATM network architecture, virtual circuit management performance, and others.