Malaysian Journal of Computer Science, Vol. 15 No. 1, June 2002, pp. 68-76

QUALPRO-RP – AN ERRORS MANAGEMENT AND REWORK EFFORT MEASUREMENT TOOL

Yun Kiew Lee
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
Malaysia
email: yklee@vads.com

Mashkuri Hj. Yaacob
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
Malaysia
email: mashkuri@fsktm.um.edu.my

Siew Hock Ow
Faculty of Computer Science & Information Technology
University of Malaya
50603 Kuala Lumpur
Malaysia
email: show@fsktm.um.edu.my

ABSTRACT

Periodic reviews of different events and milestones during the software development phases are vital, both to management visibility and progress of the development. There are both formal and informal reviews. Various errors will surface during the review process. A tool is needed to manage the errors and rework systematically. This paper highlights on the development of a Computer-Aided Software Engineering (CASE) tool for errors management and rework effort measurement in Software Quality Assurance (SQA).

Keywords: ***QualPro-RP; Errors Management; Rework Effort Measurement; Software Quality Assurance; SQA; Computer-Aided Software Engineering; CASE***

1.0 INTRODUCTION

The process of developing a software product involves a set of activities. The fundamental activities involve software specification, development, validation and evolution.

A number of model for software development has been introduced. The most widely adopted model is the software lifecycle model. This model divides the software development process into the following phases [1, 2]:

1) Software Requirements Analysis
2) Design
3) Build/Code
4) Test
5) Documentation

At the end of each development phase, a software review team will conduct a review process. There are different types of review processes that aim to achieve various purposes. Some reviews are held to achieve written agreement with the customer and others are to provide interface insight into design and development efforts [3]. Reviews are among the methods of validating the quality of a process or software product. Potential outputs of a review include problem reports, corrective actions and budget or schedule status information [4].

The implementation of a review process at the end of each development phase helps to produce reliable systems. It ensures the systems developed not only fulfilled the users' requirements but also achieved other issues such as security, portability and reusability.

Most of the time, software errors are caused by human errors during the software development process. Quality control plays the role of comparing the observed software quality with the defined quality level in earlier stage, correcting the source of problems and brings the quality back within the acceptable range. Quality control minimises effort expected in rework by controlling the sources of errors [5].

Fig. 1 illustrates the relationship between quality assurance and quality control [6]. The product assurance organisation performs a quality observation (inspection) and reporting function. Developers fix the errors detected. Quality control involves the entire software development organisation, while the quality assurance function belongs to the product assurance office [6].

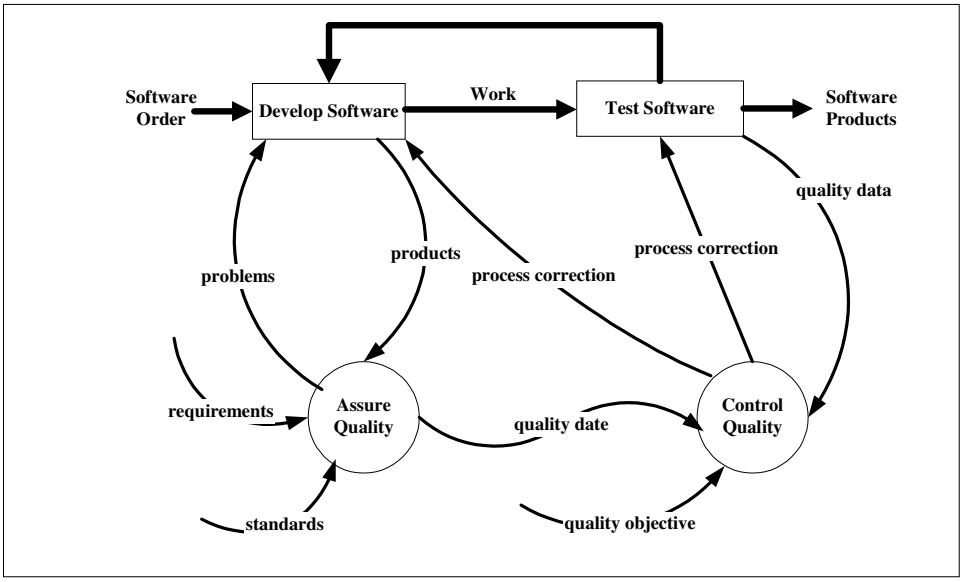


Fig. 1: Relationship between Quality Assurance and Quality Control

In order to produce software products with quality, software quality assurance (SQA) is needed. The software review process is one of the activities of SQA. The errors found during the review process need to be managed and stored in a database for future references [4]. This would help to prevent the development staff from making similar errors during the development phases. Furthermore, control such as measurement of rework effort is also needed to find out the time spent to resolve the errors. Hence, a tool is needed for such purposes.

SQA activities help in developing quality software systems. The SQA activities principally consist of auditing the software activities, software products, conducting technical reviews and checking software records throughout the software development phases. This is to ensure that the software is conformance with the software process and software product standards [7]. To achieve this, it requires the application of relevant software metrics concept throughout the software development lifecycle. This paper focuses on the development of a Computer-Aided Software Engineering (CASE) tool namely, QualPro-RP, that helps to control and manage errors, and to perform rework effort measurement. In this paper, the term "error" is used to denote both error and defect.

1.1 Error Management

The software review process is conducted at the end of every software development phase. Errors are reported during the review process. The type of errors varies depending on different environments. To a software development team, unmet users requirements are also considered as a type of error.

In order to manage the errors detected during a software review process, the errors are first classified and then grouped according to the development phases. QualPro-RP allows users to input the errors found for the entire software development lifecycle, beginning from the user requirements phase until the delivery of the software system.

QualPro-RP provides a systematic way of grouping the errors according to different development phases and sub-phases. This allows the users to record and search for the errors quickly and in a structured manner. Users who are not assigned to resolve the errors are only allowed to view the list of errors with the suggested solutions. In addition, QualPro-RP allows the administrator to create and modify the list of errors and the detailed error descriptions only.

1.2 Monitoring of Rework Effort

Effort is required to resolve the errors found during the review process. The number of team members involved to remove the errors within a specific time is the rework effort. The rework effort is recorded and used for the measurement of competency level of the rework team members.

For every solution or update of the detected errors, the rework team members must record it as a new case. The review team will conduct another review on the errors that have just been resolved. If the review result is still unsatisfactory, the errors will be recorded, and the rework team members will be requested to perform another rework. For each review, the rework team members must record the solutions together with the time spent using QualPro-RP. This process is repeated until they resolved the errors successfully and the review result is satisfactory.

The respective team member who is involved in resolving the errors are allowed to access, update and edit the errors information using QualPro-RP. Other users will only be allowed to view the listing of cases together with the solutions. This is to prevent unauthorised personnel from modifying the cases. The time spent by a team member in resolving the errors is the main data component for the measurement of rework effort. The total time spent by a team member on different rework tasks is then calculated.

The accuracy in the calculation of rework effort depends greatly on the honesty of the team member in logging the time spent to resolve the errors. Inaccurate data collection will certainly affect the results generated and further disturb the decision making of the project supervisor in the selection of good rework team members. Besides honesty, there are other issues that should be considered when comparing the individual performance or productivity among the rework team members. These issues include the type of programming language used in software development, experience of the personnel, complexity of the errors, the job functions and responsibilities of each rework team member.

The competency Level of a rework team member is calculated using the formula:
Total Time Spent / Total Number of Errors Resolved

The formula to calculate the rework team effort is:
Total number of team members (pax) x rework time spent

2.0 THE TOOLS AND TECHNOLOGY USED

QualPro-RP was developed in 2000 using Windows NT version 4.0, which was the latest technology available during that time. Today, although the latest technology is Windows 2000 and Windows XP, the functionalities and features of QualPro-RP are not affected by this advancement of technology. The tool still can be used and accessed by the users at anytime and anywhere.

QualPro-RP adopted the design of client/server architecture and thus, it is web-enabled. QualPro-RP can be accessed through the Internet or via the Intranet of the Local Area Network (LAN) if it is implemented as an internal application for an organisation.

QualPro-RP is running on Windows NT version 4.0 platform together with Microsoft Internet Information Server version 1.0 (IIS v 1.0) as the web server. The web server is the core engine to enable the system to offer web-based services. A request or transaction comes from the client site will go through the transaction server before it reaches the web server. The transaction server is a queuing engine to queue up the requests one by one for the web server to process and response to them. The database server sits on the same machine as the web server. Microsoft SQL server version 6.5 is used for the database.

The development of QualPro-RP used Microsoft Visual Studio version 6.0 as the software development tool for the client and server scriptings. Hence, VBScript, JScript and HyperText Markup Language (HTML) were written using the Microsoft Visual InterDev 6.0 [8]. The Microsoft Visual Basic 6.0 was used in the development of Component Object Model (COM) object. QualPro-RP provides applications for both server and client ends. It was developed based on a three-tier architecture, comprising the front-end, the business object and the back-end as shown in Fig. 2.

a) The Front-End Layout

The front-end furnished with the user interface for the system. The end user can access the system through the Internet or Intranet via the web browser such as Microsoft Internet Explorer. Validation and verification are performed at this layer for the general text input and editing. JScript, VBScript and HTML were used to construct this layout.

The HTML pages display the layout when the web pages are launched [9, 10]. The HTML handles the presentation such as reports and result analysis listings in tabular and graphical formats. It also manages the input of data from the users in a form format. The JScript and VBScript handle the validations of data input by the user before the result is passed to the server for further processing.

b) The Business Object

The business object consists of logic processing modules, data input/output validation and verification modules, policy enforcement and resource management. It is the "Middle Man" of the front-end and the back-end system. This component comprises the COM, which was built using Visual Basic 6.0.

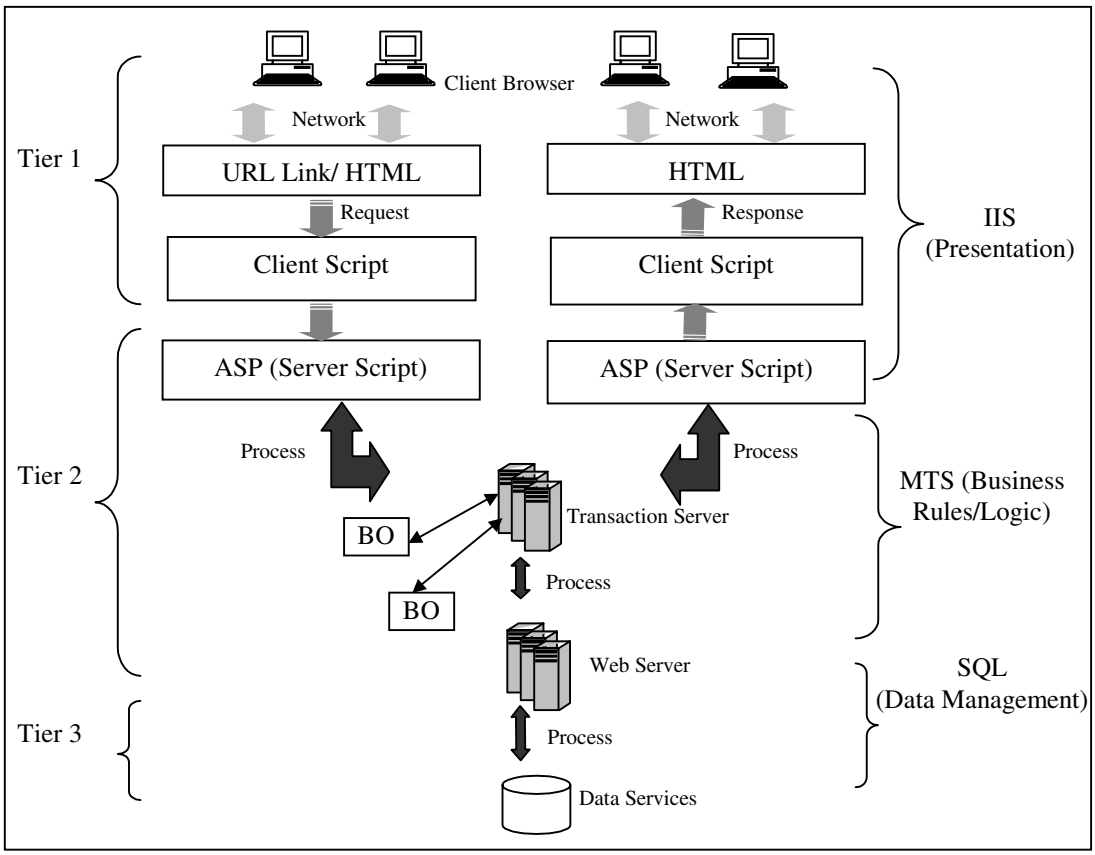


Fig. 2: System Architecture and Process

The file that is being processed by the web server is the Active Server Pages (ASP), which uses some of the built-in objects of the IIS web server [11]. These ASP files can be written in either VBScript or JScript. The codes embed the COM objects and pass parameters into and get the values out of the business objects.

The business object enables faster processing time when compared with the two-tier architecture. With the implementation of the business object, the maintenance and modification of the current system could be made easier as it is object-oriented. The objects are independent from each other and the modification of one object usually would not have much impact on another object. Data stored in the database server is extracted and input by the COM object.

c) The Back-End System
This layer contains the database management system (DBMS), which maintains consistency and secures the data through the enforcement of referential integrity and data security. Transactions such as query, update, modify, security setting and add user permission are executed in this layer.

3.0 MODULES OF THE SYSTEM

As shown in Fig. 3, QualPro-RP consists of the following three main modules:

1) The Maintenance Module
This allows a user to create, view, edit and delete errors for the different review in the software lifecycle. It also allows new members to be added to a project for the rework task.

2) The Rework Effort Module
This module allows the rework team members to access and record errors and cases, input the solutions and time spent to resolve the errors. It also allows the user to check the rework task assigned to the team members.

3) The Result Analysis Module
This module enables the user to check the individual effort as well as rework team effort.

It also allows an administrator to create and register new members into the system and assign tasks to them. In order to achieve better management of the project progress, only one administrator at the supervisory level is authorised to create, edit and delete the review and rework team members.

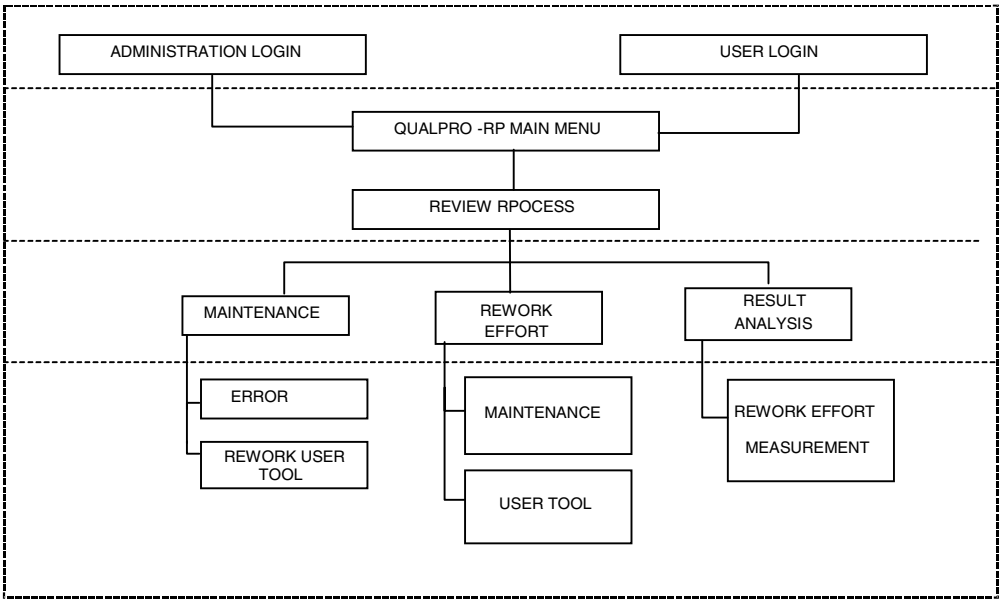


Fig. 3: Modules of QualPro-RP

4.0 THE REVIEW PROCESS

In order to conduct a formal review and rework process, the review team and rework team must first be established as shown in Fig. 4. After the review team had been formed, review tasks will be conducted on the documents/deliverables assigned pertaining to the system to be developed. Basically, the team will compare the users' requirements with the output. If the output does not meet the requirements, then it is considered as a type of errors at the requirements analysis phase. Besides reviewing the user requirements, design and program codes are

also reviewed. The errors found will be recorded into QualPro-RP. A solution would be suggested for the detected errors. The rework team members will be appointed to resolve the errors based on the suggested solutions.

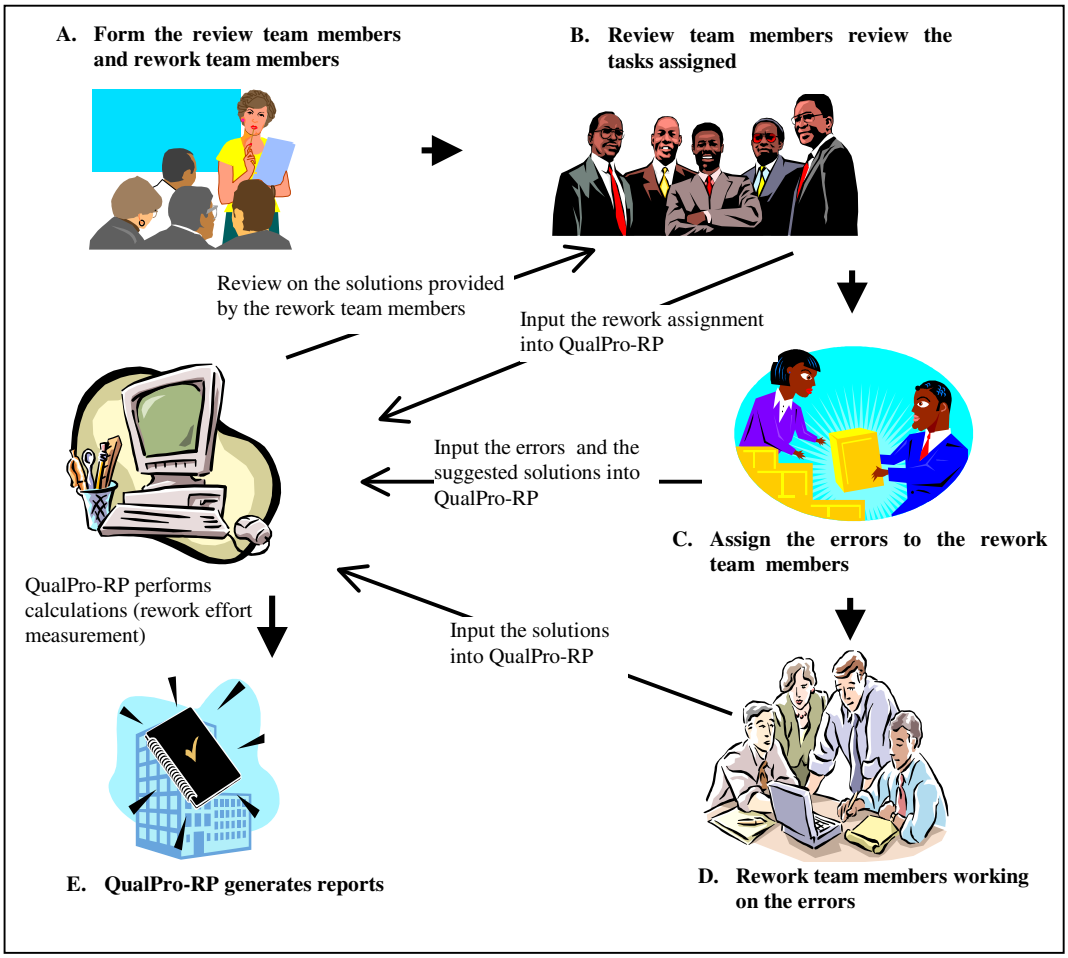


Fig. 4: The Review Process Flow

Before the review or rework team members begin to use QualPro-RP, a few preliminary steps must be performed in order to allow them to access the system. The system supervisor (who has the full access rights) must first create the project. All the people involved in the development, review and rework processes are then registered into QualPro-RP by the supervisor. Usually, they are assigned the access rights to specific functions that are relevant to their job responsibilities. For example, a rework team member is allowed to add a case but not allowed to add new errors into the error database.

The review team member who has been assigned the administrator access rights can record the errors encountered during the review process through the maintenance module. In most of the cases, the authority to record new errors (i.e. errors which is not found in the existing database of the error lists) is assigned to one of the team members only. The supervisor who possesses an in-depth knowledge of error attributes and groupings would be the most ideal person to be assigned this responsibility.

After the errors have been verified and recorded, one or more people from the rework team would be assigned to resolve the errors. The administrator will register and create accounts for these members to perform errors removal. They can record the comments and resolution for the errors into QualPro-RP.

Each rework team member must record the following data for the calculation of rework effort and recording of the rework process and outcomes:
1) The login date
2) Time spent to resolve the errors
3) Corrective actions performed

Once a rework team member has completed the rework process, he must close the task. Then, the review team will be informed to conduct another review on the errors together with the rework solutions provided. If the solutions are satisfactory, no further rework is required. If the solutions are still unsatisfactory, the review team would record the same error again and assign it to the same person or another rework team member. This process is repeated until the solutions provided are satisfactory.

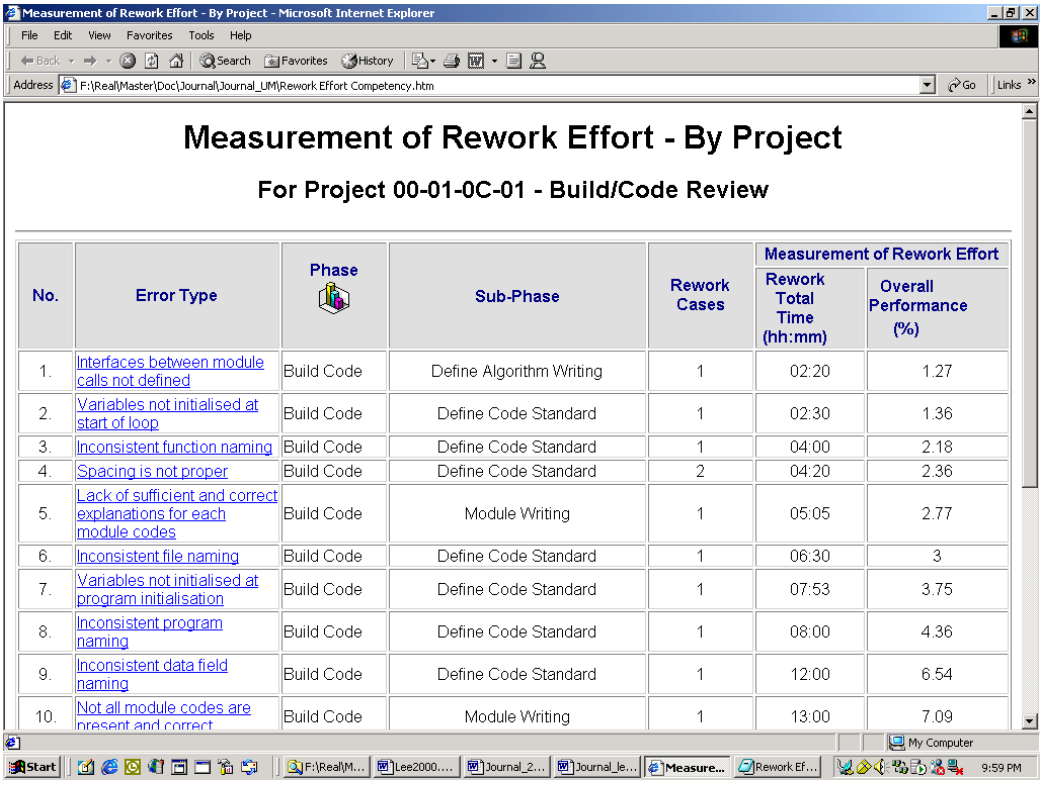
These review and rework processes are applied throughout the five major software development phases mentioned above. QualPro-RP also provides a function that generates various reports based on the data collected during the rework process. These reports are presented in both tabular and graphical formats for management purposes.

5.0 CASE STUDY RESULTS

A CASE study was conducted to investigate the usefulness of QualPro-RP in error management and rework effort measurement. This case study involved five rework team members from a software company (Company A).

A software project was selected for the case study. The objective of this project is to develop a software system for a web-based portal. This project contains ten modules. The following three modules were selected for the case study:
1) Online payment - real time online credit card payment
2) Affiliation - with new web pages and reports introduced to an existing portal
3) Budget reporting - reports which show the budget, income and analysis of year end income

The case study was conducted from December 2000 to March 2001 (four months). Every rework team member was assigned to resolve certain number of errors that was found during the review process. The competency level of each rework team member and the overall rework team effort were calculated using QualPro-RP.



No.	Error Type	Phase	Sub-Phase	Rework Cases	Measurement of Rework Effort	
					Rework Total Time (hh:mm)	Overall Performance (%)
1.	Interfaces between module calls not defined	Build Code	Define Algorithm Writing	1	02:20	1.27
2.	Variables not initialised at start of loop	Build Code	Define Code Standard	1	02:30	1.36
3.	Inconsistent function naming	Build Code	Define Code Standard	1	04:00	2.18
4.	Spacing is not proper	Build Code	Define Code Standard	2	04:20	2.36
5.	Lack of sufficient and correct explanations for each module codes	Build Code	Module Writing	1	05:05	2.77
6.	Inconsistent file naming	Build Code	Define Code Standard	1	06:30	3
7.	Variables not initialised at program initialisation	Build Code	Define Code Standard	1	07:53	3.75
8.	Inconsistent program naming	Build Code	Define Code Standard	1	08:00	4.36
9.	Inconsistent data field naming	Build Code	Define Code Standard	1	12:00	6.54
10.	Not all module codes are present and correct	Build Code	Module Writing	1	13:00	7.09

Fig. 5: Measurement of Rework Effort – By Project

The following three types of reports were generated:
1) by project – analysis of errors detected and resolved for a specific project
2) by error – analysis of rework team members' effort (time) for a specific group of errors
3) by software development phase – analysis of rework team members' effort (time) for a specific software development phase

Fig. 5 shows a report of the measurement of rework effort for the project "00-01-0C-01 – Build/Code Review" of the online payment module. A bar chart would be displayed when the user clicks at the chart icon indicated in the "phase" column. The total rework time spent on each of the development phase was calculated using QualPro-RP.

Table 1 shows the detailed rework time spent by the rework team members involved (denoted by the userid: C0001 - C0005) in the case study.

Table 1: Total Rework Time for three modules in case study

Userid	**Requirement**	**Design**	**Build/Code**	**Test**	**Documentation**	**Total**
Module: Online Payment						
C0001	7:10	46:16	53:36	169:00	52:00	328:02
C0002	2:30	29:40	32:05	75:30	43:15	183:00
C0003	0:00	68:10	67:45	87:30	24:00	247:25
Sub-Total	9:40	144:06	153:26	332:00	119:15	758:27
Module: Affiliation						
C0004	16:00	50:00	94:30	44:00	10:00	214:30
Sub-Total	16:00	50:00	94:30	44:00	10:00	214:30
Module: Budget Reporting						
C0005	12:00	16:10	18:30	19:00	2:00	67:40
Sub-Total	12:00	16:10	18:30	19:00	2:00	67:40
Total	**37:40**	**210:10**	**266:26**	**395:00**	**131:15**	**1040:37**

*Unit of measurement in hours:minutes (hh:mm)

6.0 USER FEEDBACK

A questionnaire was designed to collect feedback on the usefulness of QualPro-RP. This questionnaire was distributed to each of the rework team members involved in the case study. The questionnaire consists of two sections:
1) The Pre-system Walkthrough
2) The QualPro-RP System

Besides distributed to the five rework team members involved in the case study, the questionnaires also distributed to ten other members who were not involved in the case study. Below are the summaries of the user feedback based on fifteen sets of questionnaire collected:
1) Two hours of training session is recommended for the users of QualPro-RP (to get familiar with the tool)
2) QualPro-RP is easy to use and user-friendly
3) QualPro-RP is a good tool as it provides analysis of the competency level of the rework team members in both listing and graphical formats. This helps the management to evaluate the performance of the rework team members.

7.0 CONCLUSION

QualPro-RP was developed to help support software review processes perform throughout the development of a software product. It enables a better management of the errors detected. The results of the review processes are recorded in the database. These results can be used for future reference by the software development team to prevent making similar mistakes. This would certainly contribute to the reduction of the time spent in a review process. In addition, based on the individual rework effort, a project manager would be able to identify and select the employees who are competent in software debugging. The measurement of rework team effort helps to improve the accuracy in the estimation of time and costs incurred in a software review process.

REFERENCES

[1] Budgen, D. 1994. *Software Design.* Wokingham, Addison-Wesley Publishing Company.

[2] Sommerville, I. 1995. *Software Engineering.* New York, Addison-Wesley Publishers, Inc.

[3] Bruce, P. and Pederson, S. M. 1982. *The Software Development Project Planning and Management.* New York, John Wiley & Sons, Inc.

[4] Fisher, A. S., 1988. *CASE Using Software Development Tools.* New York, John Wiley & Sons, Inc.

[5] Malaiya, Y. K. and Srimani, P. K. 1991. *Software Reliability Models: Theoretical Development, Evaluation, and Applications.* Los Alamitos, IEEE Computer Society Press.

[6] Card, D. N. and Glass R. L 1990. *Measuring Software Design Quality.* Englewood Cliffs, Prentice-Hall, Inc.

[7] Software Quality Assurance Plan (based on ANSI/IEEE Std 730.1-1989). 2001. *http://davidfrico.com//ieeesqapdoc.doc.*

[8] Banick, S. and Morrison, M., 1998. *Using Visual InterDev™ 6.* Indianapolis, Que© Corporation.

[9] Refsnes Data (1999-2001) copyright. W3schools Online Web Tutorials. http://www.w3schools.com/

[10] Wang, W. 1996. *Visual Basic® 4 For Windows® For Dummies.* Foster City, IDG Books Worldwide, Inc.

[11] Hettihewa, S. 1999. *Sams Teach Yourself Active Server Pages 2.0 in 21 Days.* Indianapolis, SAMS.

BIOGRAPHY

Yun Kiew Lee obtained her Bachelor of Computer Science from University Putra Malaysia. Currently, she is pursuing her Master of Computer Science at the Faculty of Computer Science & Information Technology, University of Malaya. Her research areas include computer software engineering and CASE tools.

Mashkuri Hj. Yaacob is the Deputy Vice Chancellor (Academic) of the University of Malaya. His research areas include design for testability, tools for teaching - computer architecture, software reliability engineering, instruction set architecture, VLSI technology, ATM network architecture and virtual circuit management performance.

Siew Hock Ow obtained her PhD in Computer Science from the University of Malaya in 2000. Currently, she is an Associate Professor and Head of the Department of Software Engineering at the Faculty of Computer Science & Information Technology, University of Malaya. Her research areas include software quality assurance (SQA), software metrics, Computer-Aided Software Engineering (CASE) technology, project management and E-Learning.