A SURVEY ON THE CURRENT PRACTICES OF SOFTWARE DEVELOPMENT PROCESS IN MALAYSIA

*Fauziah Baharom, **Aziz Deraman and **Abdul Razak Hamdan

**Faculty of Information Technology, Universiti Utara Malaysia*
06010 Sintok, Kedah, Malaysia

***Faculty of Technology and Information Science,*
University Kebangsaan Malaysia
43600 Bangi, Selangor, Malaysia

Email: fauziah@uum.edu.my
Email: a.d@pkrisc.cc.ukm.my
Email: arh@ftsm.ukm.my

ABSTRACT

Software quality is becoming increasingly important as our society becomes more dependent on computers. The required levels of quality are not likely to be achieved without giving proper attention to the software development process. Many studies regarding the practices of software development process have been carried out. The focus of these studies is mainly to examine the effectiveness of the practices. Most of studies were conducted in the United States, European countries, India, Japan and are very limited in Asian countries especially in Malaysia. In the view of that, we have carried out a study on the current practices of software development process in Malaysia. In this paper, we have analyzed and discussed the results found from the study. The structured questionnaire approach was used for data collection. The data was analyzed by using basic statistical techniques such as frequency and cross-tabulation. The results of this study have laid out some important points regarding the practices of the software development process in Malaysia. In addition, most of the practitioners were pursuing practices that have been traditionally used by the organizations. Lack of awareness in using good software development practices could be a key factor that contributes to the occurrence of quality

problems in the organizations. Consequently, the results of this study have given us a good incentive to prolong our research in the area of software process certification.

Keywords: Software development practice, software quality, CASE tools and software testing.

1.0 INTRODUCTION

Nowadays, computers have been recognized as useful tools to support the daily works by almost all businesses in the world. These tools are commonly used to perform various crucial functions such as billing, payment, inventory control, purchasing and process control (Ward & Venkataraman, 1998). As computers are operated by software therefore producing high quality software is becoming a major challenge for the software industry. Quality was seen as a means to increase productivity, lower costs and providing the customer with desired product (Wheeler & Duggins 1998) and lack of quality would give a significant impact on society and economy such as increasing costs and property damage.

Issues related to software quality problems have been discussed since the early days of computing. Reports about unfinished development projects, project overruns and system failures are still the rule (Kautz & Nielsen, 2000). As we know, development of software is known to be a complex task that covers various activities until the final product can be delivered to the user or customer. The required levels of quality are unlikely to be achieved without giving proper attention to the software development process (Bekbay & Liu, 2003). Consequently, several innovations for improving software development process have been developed and promoted in the software industry.

Many studies regarding the practices of the software development process have been carried out. The focus of these studies is mainly to examine on the effectiveness of certain practices. Most of the studies were conducted in the United States (US), Europe, India and Japan but very few were carried out in Asian countries especially in Malaysia. Therefore, this study has been carried out to focus on the current practices of software development process in Malaysia. In addition, the aim of this study is to investigate the deployment of software development activities, the use of tools, programming languages and

techniques, and also to examine the issues and problems that might occur during the development process. This paper will highlight and discuss the findings found from the study.

2.0 RELATED WORKS

Most of the studies regarding software development practices have been carried out in the United Kingdom (UK), US, Europe, Japan, India (Wheeler & Duggins, 1998; Cusumano et al, 2003) and New Zealand (Groves et al, 2000). However, very limited numbers of similar research studies have been carried out in Asia and particularly in Malaysia (Yazrina et al, 2002).

Ward and Venkataraman (1998) had observed on various approaches for improving software quality such as better evaluation techniques, better quality measurement, better development processes and better development tools.

Blackburn, Scudder and Van Wassenhove (1996) conducted a survey, which aimed to identify the factors that contributed to the timeliness of software development. Several factors were identified which are the use of prototyping approach, CASE tools, parallel development, communication between team members, reuse of code and implementation of testing strategies. The results also indicated that spending more time and effort on customer specifications will improve the development productivity and speed.

In 1990, Cusumano and Kemerer (1990) published a survey of 40 software development projects in the US and Japan. The results showed that both countries were similar in several areas such as types and size of software products developed, development tools utilized, programming languages and hardware platforms used for development. However, they were differ in term of time spent during software development projects. In Japan, the study showed that their software people spent more time on product designing phase while in the US, more time was spent on actual coding. As a result, Japanese projects were completed with fewer numbers of defects and higher line-of-code productivity compared to the American projects.

Starting from 2001, the study on software development practices was prolonged to be more global. Cusumano, MacCromack, Kemerer and Crandall (2003) had conducted a global survey of software development practices. Their study

took about 2 years to complete. A total number of 104 software development projects responded to the survey. The projects were primarily centered in India, Japan, Europe and the US. Most of the software projects in India, Japan and Europe used architectural, functional and design specifications rather than just writing code with minimal planning and documentation. This is different compared to software projects in the US, where the sample indicated that specifications are less frequently used. The sample also reported that 100% of Indian projects used detailed designs and implemented design reviews. These responses suggest that there is some truth to the recent speculation that Indian software companies are increasingly adopting more formal project management techniques. Almost all of the samples made use of early beta releases, which is becoming a useful tool for testing and obtaining user feedback. In Japan, the waterfall process is still a popular choice, while most of the Indian, European and US projects are broken into multiple milestones known as the incremental process.

Yazrina et al (2002) have done a study on the use of system development methodology in Malaysia. The method used to carry out the study is by using questionnaire. The results of the survey have been analyzed using basic statistical measures such as mean comparison, frequency, cross tabulations, correlation and variance analyses. Findings showed that there were some respondents who did not use any methodology while developing an information system due to lack of expertise and too many deliverables that need to be delivered from the methodology. However, 43.3% of the respondents agreed that the use of methodologies are helpful in the development of an information system because it involved certain type of tools such as CASE tools, word processing, spreadsheets, graphic and presentation software. Tools ease the information system development and the development could be completed on schedule or time.

Groves et al (2000) had conducted "*A Survey Of Software Development Practices in the New Zealand Software Industry*". The study focused on determining the techniques and tools used for expressing software requirements. The group of researchers reported that there is a wide range of kinds and sizes of software organizations as well as type of applications developed in New Zealand. This diversity makes it difficult to identify meaningful patterns in the software development practices employed. The larger software development groups tend to have more well-defined software development processes, spending more time on capturing requirements and to have more rigorous testing regimes. It

also appears that the well-defined software development processes are used to produce software for customers rather than in-house providers. Most of the respondents used text documents and general-purpose tools such as word processors and spreadsheets to express requirements and specifications. Most companies stated that they used standard change request forms to capture requirements changes and they hold review meetings after each early lifecycle phase except for code reviews.

3.0 RESEARCH METHOD

The general objective of this survey is to examine the current practices of software development processes and to identify the related issues and problems which commonly occur in producing software products in Malaysian organizations. It is supported by the following six specific objectives:

i. To determine the extent of the use of standards and procedures, tools, methods and software development life cycle models
ii. To determine the deployment of software development activities
iii. To determine the extent of software reuse
iv. To determine the awareness of the user or/and customer involvement during software development
v. To identify the quality problems whichcommonly occur in producing software product
vi. To determine the awareness of improving the developer's skills

A structured questionnaire with 69 questions was used in this study. The questionnaire was divided into four sections, which include:

i. *Respondent background.* This section covers the general information of the respondent such as the current position, years of experience, his/her involvement in information technology and activities in which he/she is currently involved.

ii. *Organizational profile.* It solicits information regarding the type of organization, organization's function or type of industry, whether the organization has an IT department and number of IT professionals in the organization.

iii. *Process quality control.* This section seeks information related to the use of life cycle models, the deployment of standards and procedures,

the use of methodologies, tools and techniques and the implemented activities during development.

iv. *Human resource development.* This last section looks for training programs provided by the organization for improving the skill of the staff.

4.0 DATA ANALYSIS

About 220 sets of questionnaires were distributed to private and public sectors. Out of this amount, only 45 were returned, 4 sets without answers and only 41 sets were answered completely and were used for analysis. The results of the survey have been analyzed by using basic statistical methods, which are frequency and cross tabulations. These will be discussed in the next section.

4.1 Demographics and general information

The set of questions asked in the first two sections of the questionnaire are used to assess the qualifications of the respondents and the organization profile.

- Respondents' Qualifications
  In this section, the respondents were asked about their current positions and years of experience. The data was then analyzed using cross tabulations analysis. Table 1 shows the results obtained from the analysis.

Table 1: Respondent's Qualification

Current Position	Less than 3 years	3 to 10 years	11 to 20 years	More than 20	Total
Software Development Manager	1	2	2	1	6
Software Support Manager	1	2	1	0	4
Systems Analyst	8	3	0	0	11
Project Leader	0	3	3	0	6
Software Engineer	3	0	0	0	3
Programmer Analyst	2	4	1	0	7
Other IT Professional	0	1	2	1	4
Total	15	15	9	2	41

Out of the 41 respondents, only 26.83% had working experiences of more than 10 years and most of them were Systems Analysts.

- **Organizational Profile**

The second section of the questionnaire is needed in order to study the organizational background. Table 2 depicts the number of organizations that have an IT unit or department.

Table 2: Organization with/without IT Department

IT department/unit	Government	Private Sector	Total
Yes	100% (21)	80% (16)	37
No	0% (0)	20% (4)	4
Total	**21**	**20**	**41**

From the results acquired, it can be concluded that most organizations have IT departments especially the government agencies (100%). On the other hand, only 4 (20%) out of 20 private agencies do not have IT departments. These organizations are involved in different types of businesses such as Education and Training, Construction, Software development and so on. Table 3 shows the distribution of the organizations based on their type of business.

Table 3: Business Sectors

Type of Business	Number of Organizations	Percentage (%)
Service & Public Administration	8	19.51
Education/Training	4	9.76
Mining & Quarrying	1	2.44
Manufacturing	2	4.88
Electricity, Gas, Water Supply	1	2.44
Construction	1	2.44
Transport & Storage	1	2.44
Telecommunication	2	4.88
Banking, Financial & Insurance	1	2.44
Software Development	11	26.83
Health & Social Work	2	4.88
Consultation	2	4.88
Others	5	12.19
Total	**41**	**100**

A high percentage of respondents came from small IT-based organizations and the number of IT professionals employed is between three to five people (see Table 4).

Table 4: Number of IT professional

Number of IT Professional	Number of Organization	Percentage (%)
1 - 2 people	3	7.32
3 - 5 people	11	26.8
6 - 10 people	2	4.88
11 - 20 people	6	14.63
21 - 30 people	3	7.32
31 - 40 people	1	2.44
41 - 50 people	6	14.63
51 - 100 people	3	7.32
More than 100 people	6	14.63
Total	**41**	**100.00**

4.2 Software Development Practices

In this section, the results of the findings are presented based on the stated objectives.

Objective 1: To determine the extent of the use of standards and procedures, tools, methods and software development life cycle models.

- **The use of standards and procedures**

The use of standards and procedures is required to bring uniformity and control to the process of developing a software product (Wheeler & Duggins, 1998). In this survey, our respondents were asked about the standard to which they refer such as IEEE, ISO 9000, CMM, other international, local standards or none at all.

From the analysis, the results indicated that 36.59% of the organizations that responded are following their own standards. Surprisingly, 31.71% of organizations are still unaware of the importance of standards in producing high quality products. Only 7.32% are using IEEE standards and none of them are using CMM (0%) to control their software development process. ISO 9000

standards are the most acceptable international standards in Malaysia. The detail results obtained are as shown in Table 5.

Table 5: The Use of Standard

Type of Standard	Number of Organization	Percentage (100%)
IEEE	3	7.32
ISO 9000	10	24.39
Local	15	36.59
Not at all	13	31.71
Total	**41**	**100.00**

The results also convey that formal procedures are used to:

- manage the software development project such as cost estimation and project scheduling (80.5%).
- ensure a deliverable is properly understood whenever it is passed from one discrete group to another (e.g. user to analyst to designer) (80.5%).
- control changes to software requirements, designs and other documentation during development (75.6%).
- check whether common code standards were applied to each software project (63.4%).
- ensure that formal reviews of project status were periodically conducted (63.4%).
- make sure that independent audits (such as inspections or walkthroughs) were conducted for each major stage in the software development process (48.8%).

- **The Use of CASE Tools**

Nowadays, many CASE tools are available that can be used to support the software development process such as management tools, editing tools, prototyping, supporting tools, testing and debugging tools. These tools are helpful as they lead to some improvement in software quality and productivity.

From the analysis, it is shown that private sectors are more aware in using CASE tools compared to government agencies. The results are presented in Figure 1.

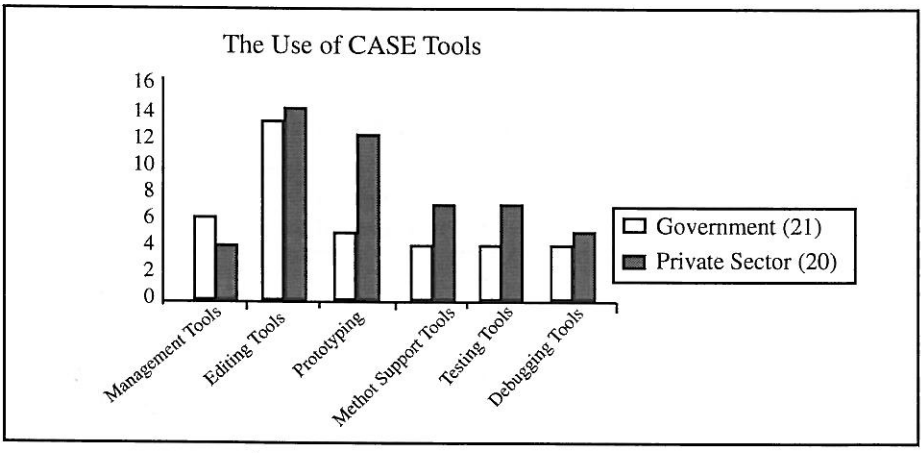

Figure 1: The Use of CASE Tools

The use of specific notations is helpful in presenting software requirement specifications. In addition, it will create better communication among developers or between developers and users/customers. From the results acquired, most of the organizations preferred to use semi-formal notations (e.g. DFD, ERD) compared to formal (e.g. Z, VDM) and informal (Natural Language) notations.

Table 6 lists the various programming languages used by the respondents during development activities. It is obvious from Table 6 that visual language and object oriented are the most popular languages used by the organizations.

Table 6: The Use of Programming Language

Programming Language	Number of Organizations	Percentage (%)
3GL (e.g.: COBOL, Fortran, C)	5	12.20
4GL (e.g.: Informix, DbaseIV, SQL)	12	29.27
Visual Language (e.g.: Visual Basic, Delphi)	18	43.90
Object Oriented (e.g.: Java, C++)	18	43.90
Others	15	36.59

- **The Use of Software Process Model**

Normally, specific software process models such as Waterfall, spiral, rapid prototyping will be used to develop software. In this study, the respondents

were also required to specify the level of software process model that they used. From the results attained, 39.02% of the organizations had never used spiral and object-oriented models. The most acceptable model that was regularly and always used is the waterfall model. Table 7 below presents the detailed results:

Table 7: The Level of Software Process Model Used

Level of	Waterfall	Rapid Prototyping	Evolutionary	Formal Method	Incremental	Spiral	Object Oriented	Reuse Oriented
Never	29.27	9.76	29.27	29.27	29.27	39.02	39.02	31.71
Rarely	2.4	12.20	14.63	12.20	12.20	17.07	17.07	14.63
Sometimes	4.88	24.39	12.20	7.32	14.63	2.44	4.88	7.32
Regularly	29.27	17.07	9.76	19.51	9.76	7.32	7.32	21.95
Always	21.95	19.51	2.44	4.88	2.44	2.44	0.00	2.44
No Answer	12.20	17.07	31.71	26.83	31.71	31.71	31.71	21.95

- **The Use of Requirement Collection Techniques**

The most critical activity that always arises during software development is the collecting requirement of the proposed system. This activity is vital to have a clear understanding on the needs of user or/customer. Understanding the nature of the problems can be very difficult especially if the software system is new. Therefore, in order to simplify the requirement collection process, some techniques and methods have been introduced.

Table 8 specifies the various techniques that can be used for requirement collection. In this study, interview seems to be the most popular technique that has always been used by the organizations.

Table 8: The Use of Requirement Collection Techniques

Level of Use	Questionnaire	Interview	Observation	Use case scenario	Document review
Never	5 (12.2%)	0 (0%)	1 (2.44%)	6 (14.63%)	0 (0%)
Rarely	12 (29.27%)	2 (4.88%)	6 (14.63%)	7 (17.07%)	5 (12.2%)
Sometimes	9 (21.95%)	6 (14.63%)	13 (31.71%)	8 (19.51%)	14 (34.15%)
Regularly	11 (26.83%)	13 (31.71%)	14 (34.15%)	13 (31.71%)	9 (21.95%)
Always	1 (2.44%)	20 (48.78%)	6 (14.63%)	4 (9.76%)	12 (29.27%)
No Answer	3 (7.32%)	0 (0%)	1 (2.44%)	3 (7.32%)	1 (2.44%)

The various software requirement analysis methods used by the organizations have been listed in Table 9. The table also reveals that the Jackson System Development method has not been applied by any of the organizations. More than half of the organizations are using SAD, SSADM and OOA methods.

Table 9: The Use Of Software Requirement Analysis Methods

Methods	Number of Organization	Percentage (%)
Structured Analysis and Design (SAD)	6	14.63
Structured System Analysis and Design Method (SSADM)	12	29.27
Structured Requirements Definition	4	9,76
Jackson System Development	0	0.00
Object Oriented Analysis (OOA)	8	19.51
No Specific Method	14	34.15

Objective 2: To determine the deployment of software development activities.

- **The Deployment of Software Development Activities**

This section will discuss the deployment of activities during software development. From the results acquired, half the organizations commonly used informal and ad-hoc methods and procedures with reasonable control of schedules and cost. Software measurement is an activity used to understand, improve and predict the quality of the software process or product. The survey indicates that not more than 24.4% of the organizations have implementing quantitative measurement in order to improve the quality of their product.

Total effort spent can be used to measure the effectiveness of a software development process. By looking at Table 10, it is hard to find an organization that spends more than 20% of their efforts for planning and requirement analysis. Most of the organizations spent from 5% to 20% of their efforts for planning and requirement analysis.

Table 10: Effort of Employing Software Development Activities

Effort	Planning	Analysis	Design	Development	Testing	Maintenance
less than 5%	9.76% (4)	2.44% (1)	2.44% (1)	2.44% (1)	2.44% (1)	9.76% (4)
5 to 10 %	31.7% (13)	24.39% (10)	12.19% (5)	2.44% (1)	14.63% (6)	21.95% (9)

(*continued*)

11 to 20 %	26.83% (11)	34.15% (14)	36.59% (15)	17.07% (7)	24.39% (10)	21.95% (9)
21 to 30 %	14.63% (6)	14.63% (6)	26.83% (11)	24.39% (10)	14.63% (6)	17.07% (7)
31 to 40 %	7.32% (3)	9.76% (4)	7.32% (3)	14.63% (6)	24.39% (10)	9.76% (4)
41 to 50 %	2.44% (1)	4.88% (2)	4.88% (2)	19.51% (8)	4.88% (2)	7.32% (3)
More than 50 %	2.44% (1)	4.88% (2)	4.88% (2)	14.63% (6)	9.76% (4)	7.32% (3)
No answer	4.88% (2)	4.88% (2)	4.88% (2)	4.88% (2)	4.88% (2)	4.88% (2)

Testing is an important process in improving the quality of the software product. The purpose of this process is to find errors, which might occur during specification, design and code generation. Based on the results acquired, formal testing is widely implemented in government and private organizations. The results obtained are as follows:

Table 11: Awareness of Formal Testing

Awareness	Government	Private Sector
Yes	81% (17)	85% (17)
No	19% (4)	15% (3)
Total	**100% (21)**	**100% (20)**

The results depicted in Table 12 indicate that most of the organizations began to apply formal software testing at the end of the coding phase (41.5%). The rest of the results are as follows:

Table 12: The Beginning of Formal Testing

The Beginning of Formal Testing	Percentage (%)
the end of coding phase	41.5
early, as soon as software projects were acquiring documentations or elements that can be tested	19
while integrating major software modules	22
when implementing the final acceptance testing	14.6

Table 13 indicates several available testing techniques commonly practiced during software development. The results show most of the organizations carried out unit testing, integration testing, system testing and acceptance testing.

However, about half of the organizations were practicing alpha and beta testing.

Table 13: Testing Techniques

Testing Technique	Number of Organizations	Percentage
Unit Testing	28	68.29
Integration Testing	32	78.05
System Testing	35	85.37
Regression Testing	5	12.20
Usability Testing	17	41.46
Acceptance Testing	32	78.05
Alpha Testing	10	24.39
Beta Testing	9	21.95
Functional Testing	19	46.34
White-box Testing	6	14.63

The results also indicate that while carrying out a software project, 58.5% of the organizations were producing formal test plans. In additional, 73.2% of the organizations agreed that software testing by itself is not sufficient for guaranteeing the software product meets with its requirements.

Objective 3: To examine the extent of software reuse

Software should be considered as an asset and reuse of these assets are essential to increase the return on their development costs. Demands for lower software production costs, faster delivery and increased quality can only be met through widespread and systematic software reuse (Sommerville, 2001). From the results obtained, most of the organizations are providing good encouragement on the creation and implementation of software reuse.

Table 14: The Encouragement of Software Reuse

Encouragement	Government	Private Sector
Yes	67.7% (14)	85% (17)
No	33.3% (7)	15% (3)
Total	**100% (21)**	**100% (20)**

For the organizations that encourage reuse, the most three ranked components that are frequently being reused were source code (65.85%), design documents (48.78%) and user documentation/specification (48.78%).

Table 15: Component Being Reuse

Component Being Reused	Number of Organization	Percentage
Source code	27	65.85
Data	15	36.59
Design documents	20	48.78
Requirement Specifications	18	43.90
Test data	14	34.15
Feasibility studies	12	29.27
User documentation/specification	20	48.78
Cost benefits calculators and estimates	12	29.27

In the degree of code modification, the results also indicate that only 9.76% of the organizations reuse the source code totally without any changes and 46.34% of the organizations had changed more than 20% of lines code. The detailed results are indicated in Table 16 below:

Table 16: Degree Of Code Modification

Degree Of Code Modification	Percentage (%)
Unchanged	9.76
Less than 20% of lines changed	29.27
More than 20% of lines changed	46.34
No answer	14.63
Total	100

Objective 4: To determine the awareness of the user or/and customer involvement during software development

In the trend of improving software quality, involvement from the user or/and customer is considered as essential. From the results revealed in Table 17, both the government and private sectors are concerned about this issue. More than half of the organizations stated that the user or/and customer is regularly involved during the software development process.

Table 17: Level of User or/and Customer Involvement

Level of Involvement	Government	Private Sector
very rare	19.05% (4)	10% (2)
sometimes	19.05% (4)	20% (4)
regularly	52.38% (11)	55% (11)
always	9.52% (2)	15% (3)
Total	100% (21)	100% (20)

Objective 5: To identify the quality problems which commonly occur in producing software products.

We continued our investigation by looking for the quality problems that commonly occurred in the organizations. In our questionnaires, we had listed four types of quality problems which are: software can not be delivered on time, over budget, customer unhappy with the product quality and software needs further improvement or maintenance after it was released. From the analysis, all of the organizations claimed that they have faced with at least one of the quality problems and the results are classified as follows:

- 75% of the organizations mentioned that their software products need further improvement after they were released
- 55% of the organizations could not deliver their software systems on time
- 20% of the organizations faced with budgetary problems
- 22.5% of the organizations were confronted by their customers who were unhappy with the quality of the software products.

Objective 6: To find out the awareness of improving the developer's skills

Training of employees is an important item. Almost all of the organizations (80.49%) claimed that they provide their developers and project managers training in improving software development or project management skills. More than half of the government agencies and 45% of the private sectors were providing training from 1 to 2 times per year. The table below shows the detailed results about the number of training seasons provided annually by the organizations.

Table 18: Number of Training Sessions Per Year

No. of training session per year	Government (21)	Private Sector (20)
None	9.5% (2)	15% (3)
1-2 times	57.1% (12)	45% (9)
More than 2 times	33.3% (7)	35% (7)
No Answer	0 (0%)	5% (1)
Total	100% (21)	100%

The results also indicate that 97.6% of the organizations agreed that knowledge and skills could be improved through attending appropriate training, courses and/or seminars. Training, experience and formal education are three factors that have been identified which can contribute in improving knowledge and skills in software development. From the analysis, we found majority of the organizations believed that experience is the most important factor. The result is depicted in Figure. 2.

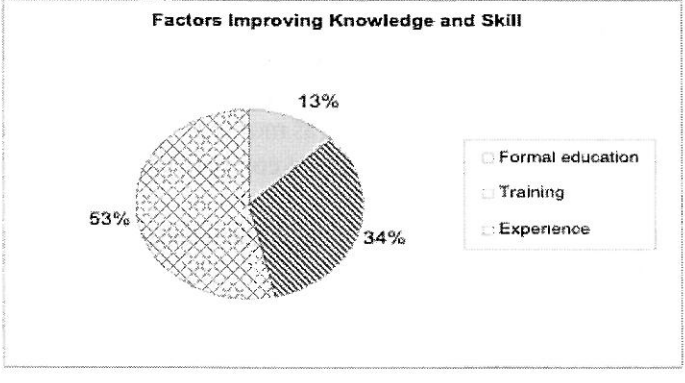

Figure 2 : Factors Improving Knowledge and Skill

5.0 CONCLUSION

It was discovered that most of the respondents work as system analysts and a major percentage of them had less than 3 years of experience. Out of 41 organizations, only 11 organizations were totally involved in software development. Looking at the respondents' years of experience (Table 1) and number of IT professionals (Table 4), to produce high quality software, standards,

CASE tools and methodologies should be used extensively. However, Table 5 indicated that 31.71% of the organizations are still not referring to any standards when developing software. A number of CASE tools were introduced to support the software development process therefore the expected software product could be delivered on time. Unfortunately, Figure 1 depicts that in the government agencies the use of CASE tools appeared to be very small especially for prototyping (23.81%), testing (19.05%) and debugging (19.05%).

According to Pressman (2001) more than 40 percent of all efforts is often recommended for analysis and design for large software development projects. However, from the results obtained in Table 10, only a small percentage of the respondents were following this recommendation.
Producing high quality software, economically and with fast delivery are supposed to be the goals of every software developer. To achieve these goals, use of methods during development process is certainly important. However, the results in Table 9 shows that software requirement analysis methods are not been applied by every organization due to reasons such as lack of expertise and too many deliverables (Yazrina et al, 2002). Therefore, for a method to be utilized, the organizations should allocate sufficient training for their staffs. In addition, adopting object and reuse oriented process models are also necessary. From the results obtained in Table 7, most of these process models have never been used. The private sector however was more encouraging in software reuse development (Table 14) due to intention to compete with their competitors.

Consequently, the results of the study have laid out some important points to be considered such as:

- It was discovered that most of organizations in Malaysia were still lacking in the use of standards.
- The majority of the organizations were facing problems such as quality, late delivery, over budget and software products needing further improvement or modification.
- The use of CASE tools in the government agencies (67.7%) appeared to be not so encouraging compared to the private sector (85%).
- 34.15% of the organizations did not use any specific method for performing requirement analysis activities.
- Source code was the only component that was frequently being reused by the organizations. Therefore, the reuse of other components should be considered in improving software quality.

- For the deployment of process model, the waterfall method was more acceptable compared to other models such as object oriented and reuse oriented.
- There were a small number of organizations that spent more than 20% of their effort for executing planning and analysis activities.
- Most of the organizations were initially executing formal software testing after coding phase were finished. Actually it is also possible to do this formal testing earlier, for instance, as soon as software projects were creating documentations or elements that could be tested.
- Alpha and beta testing was hardly implemented in the organizations.

To conclude, most of the practitioners were pursuing practices that have been traditionally used by the organizations. Lack of awareness in using good software development practices could be a key factor that contributed to the occurrence of quality problems in the organizations. The results of this study have given us a good incentive to prolong our research in the area of software process certification.

6.0 ACKNOWLEDGEMENT

This research is funded by the Intensification of Research in Priority Areas (IRPA) from Malaysia Ministry of Science, Technology and Environment (MOSTE).

7.0 REFERENCES

Bekbay, S. & Liu, S. (2003). A Study of Japanese Software Process Practices and a Potential for Improvement Using SOFL. *IEEE Computer Society*, 100 - 107.

Blackburn, D. J., Scudder, D. G. & van Wassenhove, N. L. (1996). Improving Speed and Productivity of Software Development: A Global Survey of Software Developers. *IEEE Transactions on Software Engineering*, 22(12), 875-885.

Cusumano, M. & Kemerer, F. C. (2003). A Quantitative Analysis of US and Japanese Practice and Performance in Software Development. *Management Science*, 36(11), 1384-1406.

Cusumano, M., MacCormack, A., Kemerer, F. C. & Crandall, W. (2003). A Global Survey of Software Development Practices. *IEEE Software.*

Groves, L., Nickson, R., Reeve, G., Reeves, S. & Utting, M. (2000). A Survey of Software Development Practices in the New Zealand Software Industry. *IEEE Software.*

Kaurtz, K. & Nielsen, A. P. (2000). Implementing Software Process Improvement: Two Cases of Technology Transfer. *Proceedings of the 33rd Hawaii International Conference on System Sciences 2000.*

Pressman, R. S. (2001). *Software Engineering: A practitioner's Approach.* McGraw-Hill.

Sommerville, I. (2001). *Software Engineering* (6th ed.): Addison Wesley.

Wheeler, S. & Duggins, S. (1998). Improving Software Quality. *ACM*, 300-309.

Ward, A. W. & Venkataraman, B. (1999). Some observations on software quality. *ACM Southeast Regional Conference 1999.*

Yazrina, Y. (2002). The Use of Information System Development Methodology in Malaysia. *Jurnal Antarabangsa (Teknologi Maklumat),* 2(2002), 15-34.