International Journal of Science and Research (IJSR)
ISSN (Online): 2319-7064
Index Copernicus Value (2013): 6.14 | Impact Factor (2013): 4.438

Early Prediction Model to Manage Risks in Earliest Stages to Increase Project Success

Norulhaida Abdullah[1], Marzanah A.Jabar[2]

[1, 2] Department of Computer Science and Information System, University Putra Malaysia, 43400 Serdang, Selangor, Malaysia

Abstract: ***Manage a software project is so complex and risky. If fail to organize it systematically, will increase the number of software project failure. Project failure in Software Project Management (SPM) is distinct due to the characteristic of software project: intangible, unpredictable, and often fail and one-off project. The problem in this research is how project failure can be minimize or avoided. We proposed an early prediction model with proactive signal messenger which to detect risk in earliest stages and we validated it using questionnaires and pilot study. The proposed model developed achieved its goal which is to minimize risk in software development process to increase project success. The result of this study has shown that the proactive elements detection in earliest stage can detect risk earliest and increased project success***.

Keywords: Software project management, early prediction model, early warning signs, software project failure.

1. Introduction

Software project have inherent uncertainties and risk [12]. Software project risks can be described as the product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure [1]. Normally, project failure in software project management is distinct due to the normal causes of software project in example is ineffective and not efficiency schedule planning of the project, communication breakdown among project stakeholder, project resources has been assigned to a higher priority project and sometimes subject matter expert are over scheduled. Lots of money and efforts is wasted on not produce software projects. The main causes behind failed software projects are that it is often too late to correct the problems by the time they are detected [9].

Software project risk and software development phases are dependence with each others. Software development process or the Software Development Lifecycle (SDLC) described is a structure used on the development of a software system. Normally, software development process involves five different phases: Requirements Analysis and Definition, Design, Implementation and Unit Testing, Integration and System Testing, and the Operation and Maintenance phase [10]. Each phase of the Software Development Life Cycle (SDLC) is vulnerable to different types of risk factors. It is important to identify and understand these risks because it is a preliminary stage for managing risks successfully. The project risks factors has been defined as a condition that can present a serious threat to the successful completion of software project management [1].

Based on previous research of software risk management, they were explained that by analyzed threats to success in example risks action can be taken to reduce the chance of failure of a project [1]. A list of risk was developed by experts named as early warning sign (EWS) with purposed to detect project failure in the first early project stages.

In the early project stages, the degree of uncertainty will be higher in terms of the deliverables, schedule, budget and other project parameters [3]. The early project stages are critical because of early warning sign (EWS) in these stages provided corrective action, when taken would allow the project to be completed within the original time estimated. This is because the corrective actions in this stage are cheaper than the costly recovery in later stages [3].

The challenge of managing project is to ensure project success. In managing problem if problem detected at later stage of the project will be costlier and risky to fail. In this paper we proposed a model for early detection of problem in project management stages. We analyzed current related work in early detection and we found that the early warning signals as described by Igor Ansoff's theory enable project managers to manage and anticipate otherwise unforeseeable project problems [8]. Its means percentage of project success will increase.

In this paper, we proposed a model which is identified symptoms of project failure in an earliest stage of project management as component to others method. The model has a proactive signal, which will produced result of software success whenever user complete all task or phases and activities during their project development process. The model was proposed based on systematic review of previous research. The remaining of this paper is structured as follows. In section 2, we make a review on related topic of the research and in section 3 we come out with conceptual theoretical model and then we presented the proposed model and gaps. After that, in section 4 we presented the research methodology and pilot study. Lastly, in section 5 we conclude the work and determine the limitation.

2. Literature Review

The first stage in the risk management process is to identify the risks, so that appropriate countermeasure can be taken [1]. An early warning sign (EWS) is defined as "an event or indication that predicts, cautions, or alerts one of possible or impending problems in the early stages of the project" [2]. Early warning sign is an observation, a signal, a message

which can be seen as an expression, an indication, a proof or a sign of future or incipient positive or negative issues [8].

Many studies have proven that proper management of software risks affects the success of software development projects [4]. Controlling risk in software projects is considered to be a major contributor to project success. Software projects are high risk activities, generating variable performance outcomes. Industry surveys suggest that only about a quarter of software projects succeed outright that is, they complete as scheduled, budgeted and specified), and billions of dollars are lost annually through projects failures or projects that do not deliver promised benefits [5].

Researchers have proposed a variety of classification frameworks on software project risk factors. For example, Schmidt et al. [1] launched a cross-cultural research in Hong Kong, Finland, and the U.S. They established a framework that covers 14 dimensions and 33 risks. Wallace et al. [12] classified 27 software risk factors into six dimensions. Kappelman et al. [2] classified 12 software risk factors into two risks group, people-related risks and process-related risks.

Research by Yong Hu et al. (2013) using an Integrative Framework for Intelligent Software Project Risk Planning (IF-ISPRP) to help in minimizing the impact of project risk and achieving a better foreseeable project outcome [13]. They proposed three core components which are Risk Analysis Module, Project Risk Database and Risk Planning Module and used the approach Many-to-many actionable knowledge discovery (MMAKD) in risk planning module. The aim of this approach is to produce cost minimal action set for risk control based. The strength of this study is they proposed IF-ISPRP framework which is established based on real software project data and they proposed method which is prevalent in practical risk planning compared with existing study that use integer-programming technology. But this study however had limitation which is it does not consider of risk-control action. Figure 1 below shows IF-ISPRP framework model.

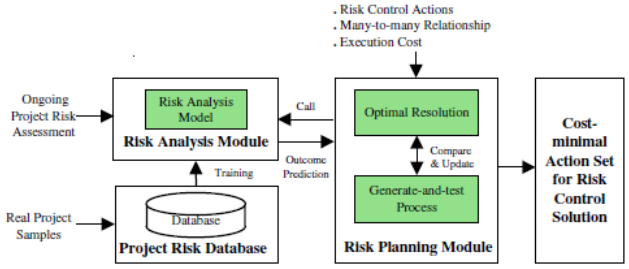

Figure 1: IF-ISPRP Framework (by Yong Hu et al., 2013)

Another review is from Ingrid Spjelkavik et al. proposed a conceptual model of Early Warning to measure factors other than time, cost, quality and measure early in a chain of cause and effect [6]. The approach consists four elements which are Mindset, Data collection, Reflection and Actions. The strengths of this research is they measure factors other than cost and quality and they also measure early in a chain of cause and effect. But in this research, they have not yet empirical data which proves the success with early warning indicators. Figure 2 shows Model of Early Warning.

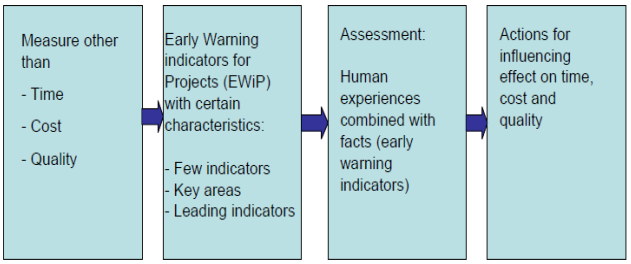

Figure 2: Early Waning Model (by Ingrid Spjelkavik)

In M. Kothari study, he said that early warnings concept is certainly a potential way to make project risk management more proactive [7]. He had proven it with theory from Ansoff theory and Nikander (2002). In fact, Nikander (2002) assert that early warnings can potentially become the cause of a problem if not acted appropriately in the time available range and hence can be seen as a risk. Figure 3 below shows the interconnectedness of the concepts of early warnings and risk by Nikander [8].

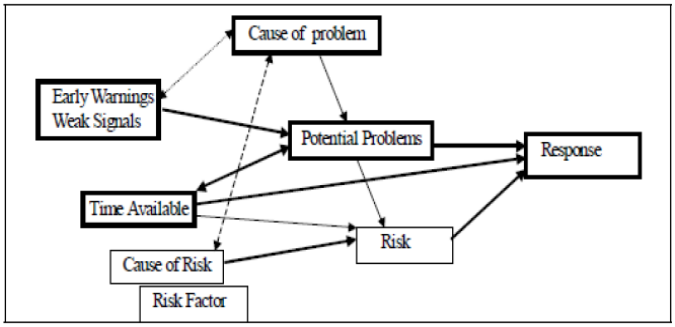

Figure 3: Interconnectedness concepts (by Nikander (2002))

As shown in figure 4, Ansoff‟s theory of weak signals can be placed in a broader sense with theories of project management. Further, Ansoff‟s weak signals phenomenon provides insights of a probable impending issue; which coincides with the understanding of the concept of early warnings.

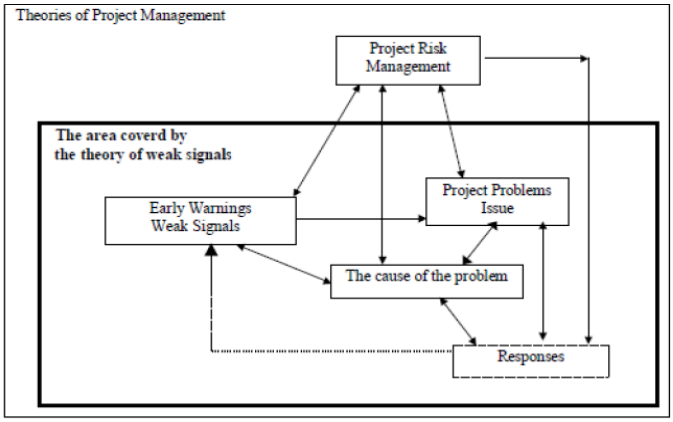

Figure 4: Ansoff ‘s theory of weak signals.

3. Conceptual Model

Based on literature review above, we come out with conceptual model as shown in figure 5 below. In our conceptual theoretical model, to ensure that a software project successful, four main elements which are Risk

Analysis, Risk Management, Indicator and Action are need to considered. It's because this four element is basics element that used by majority researchers in their model. Risk analysis is used to determine project issues and project problem such as time cost and quality in develop a software project. After risks are analyzed, risk management needed to avoid problems and failure occurred again. Every risk management projects are used at least one indicator to give alert message or we called it signal or weak signals or known as Early Warning Signs to predicted symptoms of project failure from earliest stages. Figure 5 below will be present clearly about this conceptual model.

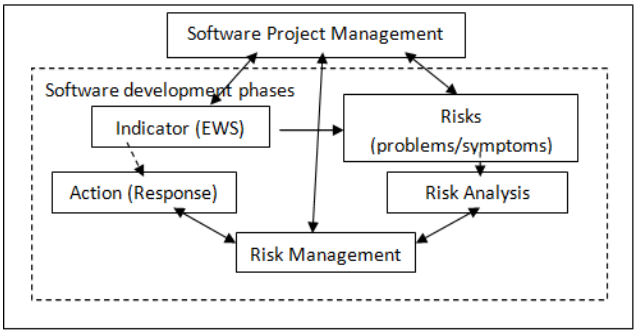

Figure 5: Proposed Conceptual Model

3.1 Proposed Model

Based on conceptual theoretical model above, we motivated to propose an early warning model which added some features as requirement of our study. The proposed framework had signal what we named its as proactive signal which is will increase project success in earliest stages and also measure factors other than time, cost and quality as presented by Ingrid Spjelkavik et al. and it also added with characteristic like cost minimal action as presented by Yong Hu et al. It's synchronized with our objectives to reduce cost and detect symptoms of project failure in early stage. As shown in figure 6, our model has five main components there are Phases, Risks, Attribute, Result and signal.

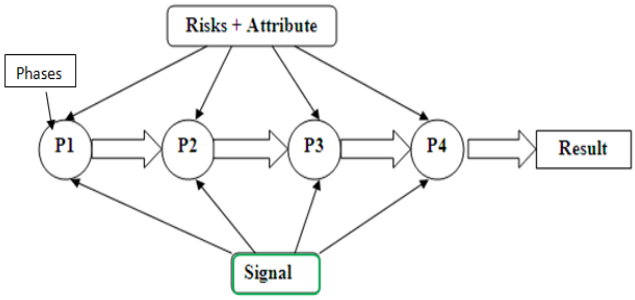

Figure 6: Proposed Model

Description about proposed model:
P1, P2, P3, and P4 -Presented the phases in software development life cycle (SDLC).
Attribute –Presented activities in SDLC and include deliverable for each phases.
Signal-Presented proactive element which is give user alert message when activities incomplete.
Risks- Presented symptoms in development software project or risk factors that causes project failure.
Result-Presented percentages of project success after every phase are complete.

Phases
The Unified Software Development Process is a popular iterative and incremental software development process framework. The best-known and extensively documented refinement of the Unified Process is the Rational Unified Process (RUP). The Unified Process divides the project into four phases **Inception, Elaboration, Construction, and Transition. Identify risk** is one from a few goal in inception phase. So, if we can identify risk in earliest stage and manage the risks, software development process phases will be organized smoothly. Software Development Life Cycle (SDLC) consists of a set of steps or phases in which each phase of SDLC uses results of the previous one. SDLC adheres to important phases that are essential for developers, such as **Planning, Analysis, Design, and Implementation**.

Attribute
A few attributes included in this model there is project size, resources size, duration, action, risk factors, deliverable and proactive signal. The attributes play their own roles in this model.

Activity
In this model we consider and proposed only 14 activities. In planning phase we suggested (identify opportunity, analyze feasibility, develop work plan, staff project, control and direct project), in analysis phase (develop analysis strategy, determine business requirement, create use case, model process and model data), in design phase (design physical system, design architecture, design interface, design program, design database and data) and in implementation phase (construct system, install system, maintain system and post implementation).

Deliverable
Every activity and phase has their own deliverable. User should attach deliverable or output after finished every task as milestone they are success finished one phase before they will ready to follow next task.

Risk
12 risk factors included in this model, there is lack of top management, lack of functional requirement, weak or not skillfully project manager, no change control, no stakeholder involvement, not efficient schedule planning, weak commitment project members, communication breakdown among project stakeholders, project resources have been assigned to a higher priority project, no business case for the project and subject matter experts are over scheduled.

3.2 Gaps and Motivation

Normally, the conceptual framework or model is used as detecting mechanism to detect early warning signs of emerging problems in the project during the execution phase and the tool need to implement as indicator for emerging problems in projects and give project managers the ability to act up on these signals at an early stage to influence the development for achieving project success [6].

The gaps between proposed framework with previous frameworks are our model exist a 'signal' which is we called it proactive signal and map software development phases to risks. The signal actually have alert message that will avoided system proceed to next level if previous task incomplete. So, the signal present unique characteristic compare than previous work and present simple way to manage project in small scale of software project.

4. Research Methodology

This research is both exploratory and constructivist in nature. Based on the research objectives and the identified research questions, the System Development Research Methodology (SDRM) as suggested by Nunamaker and Chen (1990) has been utilized as research methodology [16]. According by Limbu, 2008 SDRM methodology has been used extensively in software engineering and information systems development research domain and it can accommodate dynamic evolution of the research in order to create innovations, define new ideas, and develop new technical capabilities [15]. SDRM has five stages which is shows in figure 7.

Based on the research objectives and the selection of the SDRM research methodology, the design of the research is as follows. Stage 1 comprises of extensive literature review of the problem domain. Stages 2, 3 and 4 comprises of creation of simulation model to address the research objectives. Stage 5 comprises of observation and evaluation of the model using individual early warning signs (EWS) testing and controlled experimental study.

There are four common risk management issues based on previous studies which are about people commitment, skills, communication and users demand. Based on these issues we proposed three metrics as shown in Table 1 while proposed attributes are shown in Table 2.

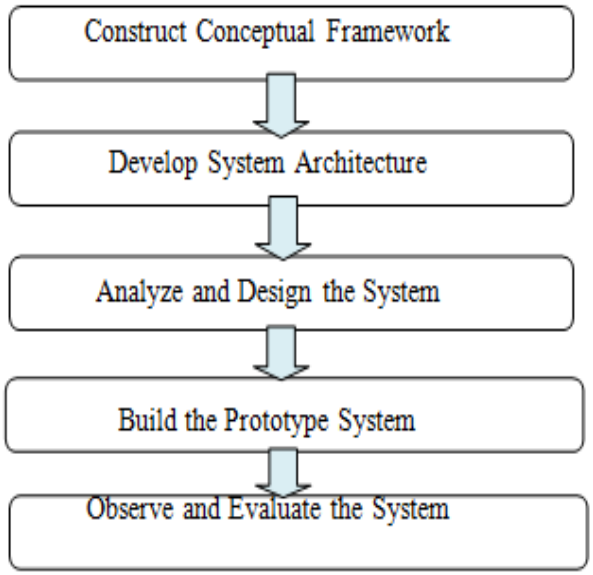

Figure 7: The research process proposed by Nunamaker and Chen (1990).

Table 1: The Proposed Metrics

Metrics	Unit	Description
Effeciency	**Defect Removal Efficiency (DRE)** = E(E+D) -E is the number of errors found before delivery -D is the number errors found after delivery	Effectiency to detect risks in every phases via signal.
Effectiveness	**Phase Containment Effectiveness for Phase i (PCEi)**= Number of phase i errors/(Num. of phase i errors+Num. of phase i defects). **Fault-slip-through(FST)**=Number of defect not found when it was most cost-effective	Effective in decrease project failure and increase project success.
Reliability	**Failure Rate (FR)**=Number of failures/execution time	Reliable to manage project well than before

Table 2: The Proposed Attributes

Attribute	Description
Number of risks	Number of risk in software project
Phases	Number of phases include in software process
Percentage of project success	Percentage of software project success to fulfill user requirement

Table 3 shown the experiment result analysis used to validate the metric 1, metric 2 and metric 3. The number of risks has shown the efficiency of proposed model and tool to detected risks. The phases and every stage that must be through by users push them to complete all task and then it shown the tool effectively will increase project success automatically. The percentage of project success shown that the reliability of the tool to help project team to manage the project and at the same time they can manage risks also. Besides that, the data in table 3 show that number of resource influence the duration of project completing and type of risks.

Table 3: Experiment Result Used To Validate Metrics and Attribute of Table 1 and Table 2.

Project	Num. of Resources	Duration	Num. of Risks Detected	List of Risk Existed in Project Development	Num. of Task Uncompleted	Percentage of Project Success
1	5	9 (months) 29(days)	3	Lack of top management. Weak required skills of team members. Project resources have been assigned to the higher priority project.	3	84.21%
2	10	6 (months)	6	Lack of top management. Weak required skills of team members. Communication breakdown among stakeholders. No stakeholder involvement. Subject matter is overscheduled. Weak Commitment of project members.	6	68.43%
3	7	8 (months)	4	Ineffective and not efficiency schedule planning of the project. Weak required skills of team members.	4	78.95%

4.1 Pilot Study

One pilot study was proceeded with choose one company that had long experience in software engineering and project management background to got experts opinion about our model and tool proposed. Data collection was achieved through the use of structured questionnaires followed TAM's format [17] which asked respondents question, aims at achieving the above objectives. Besides that, we also invited a few respondents from IT background and master students to answers questionnaires about model functionality (question 1-5) five questions and model usefulness (question 6-10) about our model also five questions. Some data was collected and presented into table 4 below.

Table 4: Result Analysis of Model Validity Test by 10 Respondents

Management Experience	Number Respondents	Functionality Agree (Ratio)	Usefulness Agree (Ratio)	Totally Agree	Agree Percentage %
0-3	3	0.6	0.8	1.4	70 %
3-5	4	0.4	0.6	1.0	50%
5-10	2	0.4	0.4	0.8	40%
10 or above	1	0.4	0.4	0.8	40%

Calculation for 'agree' percentage:-

Number of (Perceive ease of use) or functionality questions: n^f
Number of (Perceive usefulness) questions: n^u
Total of questions: t
Total of all questions: T

Formulae: $n^f /t + n^u /t = (n^f + n^u)/T$

So, total average for this tool 'agree' is 200/400%=50%

5. Conclusion

The proposed model is adopted from previous early warning sign model and risk factors framework. The model will detect the symptoms/risk factors during software development process in earliest stages before it's become costlier. The model automatically makes project manager know what actually symptoms cause and effect of project failure and project success. The contribution of this study is project failure will be decrease earliest and the cost to manage system will be reducing also. Besides, the model is quick simple and easy to use by any project. Early Prediction Model may be will be use in others platform such as social media, corporate, commercial and etc. and evaluate whether to get satisfactory result as is software project management.

5.1 Limitations

In this paper, we presented the Early Prediction Model for software project management to managing risks factors and ensure project success will be increase and project failure will be decrease before to become costlier. We proposed the 'Software Project Early Prediction System' and TAM questionnaire format to validating and verifying the model.
However, our research has the following limitations:

1. The tool, "Software Project Early Prediction System" just detect the most influential risk factors only and give the signal as project alert message, but the users need to make decision-making and risk control action manually.
2. The model may only works in certain situation, hence, if extreme result occurs, the experts must make adjustment before the implementation of the risk management plan.

References

[1] Roy Schmidt, Kalle Lyytinen, Mark Keil, and Paul Cule, "Identifying Software Project Risks: An International Delphi Study." Journal of Management Information System/ spring 2001, Vol. 17, No. 4, pp. 5-36.

[2] Leon A. Kappelman; Robert McKeeman; Lixuan Zhang, "Early Warning Sign of IT Project Failure: The Dominant Dozen." Information System Management Fall 2006; 23.4; ABI/INFORM Global Pg-31.
[3] Tom Philip; Erik Wende; Gerhad Schwabe, "Identifying Early Warning Signs of Failure in Offshore Software Development Projects-A Delphi Study." Proceedings of the sixteenth Americas Conference on Information System, Lima, Peru, August 12-15, 2010.
[4] Wen-Ming Han, Sun-Jen Huang, "An empirical analysis of risk components and performance on software projects." The journal of system and software 80 (2007) 42-50.
[5] Paul L. Bannerman (2008), "Risk and risk management in software projects: A reassessment." The journal of system and software 81 (2008) 2118-2133.
[6] Ingrid Spjelkavik; Bjorn Andersen; Lars E. Onsoyen, Tom Fagerhaug, Helge Marheim, "Project Performance Management System: A Conceptual Framework for early warnings in projects." No-7465, Mei 5.
[7] Mihir Kothari, "Early Warning Sign in Software Projects." A thesis for the degree of Master of Computer and Information Science (MCIS) 2010.
[8] Nikander, Iilmari O. (2002) "Early Warnings: A Phenomenon in Project Management." Dissertation for Doctor of Science in Technology, Helsinki University of technology, Espoo, Finland
[9] Xiaoqing (Frank) Liu, Gautam Kane, Monu Bambroo, "An intelligent early warning systems for software quality improvement and project management." The journal of systems and software 79 (2006) 1552-1564.
[10] Haneen Hijazi, Shihadeh Alqrainy, Hasan Muaidi, Thair Khdour, "Risk Factors in Software Development Phases." European Scientific Journal January 2014 edition vol.10, No.3 ISSN: 1857 – 7881 (Print) e - ISSN 1857- 743.
[11] L. Wallace, M. Keil, A. Rai, "How software project risk affects project performance: An investigation of the dimensions of risk and an exploratory model". Decision Sciences 35 (2) (2004) 289–321.
[12] Yong Hu, Jianfeng Du, Xiangzhou Zhang, Xiaoling Hao, E.W.T. Ngai, Ming Fan, Mei Liu, "An integrative framework for intelligent software project risk planning". Decision Support Systems 55 (2013) 927–937.
[13] Mohamed Saad, Sherif Mazen, Ehab Ezzat, Hegazy Zaher,"Towards a Conceptual Framework for Early Warning Information Systems (EWIS) for Crisis Preparedness". Á. Rocha et al. (Eds.): Advances in Information Systems and Technologies, AISC 206, pp. 523–534. DOI: 10.1007/978-3-642-36981-0_48.
[14] Limbu, D. K., "Contextual Information Retrieval from the WWW". Unpublished Doctoral dissertation, Auckland University of Technology, Auckland, New Zealand (2008).
[15] Nunamaker, J. F., & Chen, M., "Systems development in information Systems research". Paper presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI, USA (1990).
[16] Viswanath Venkatesh, Fred D. Davis, "A Theoretical Extension of the Model Technology Acceptance: Four Longitudinal Field Studies" Management Science vol. 46, No.2, February 2000 pp.186-204.

Author Profile

NorulhaidaAbdulah is Master Degree student of Department of Information System at University Putra Malaysia. Born on June 30, 1986 and she received Bachelor of Computer Science major in Software Engineering from Universiti Putra Malaysia in 2011. She currently continued study in Master of Science focusing on Software Engineering at University Putra Malaysia. Her research interests are focusing on managing risks in Management Information System.

Marzanah A. Jabar is Associated Professor of Department of Information System at University Putra Malaysia. Born on December 17, 1962 and she received her PhD in Management Information System, from University Putra Malaysia, Serdang, Selangor in 2008. Her field of interest is Software Engineering, Management Information System and Knowledge Management. She starts her position as senior lecturer at FSKTM, UPM on September 2008 until present. Her latest publications are Proceedings of Malaysian national Conference on Databases (MaNCoD) 2014, Knowledge Management System Quality: A Survey of KMS Quality Dimensions 2014, Acculturation, An Effective Location-Based Information Filtering System on Mobile Devices and etc.