EARLY PREDICTION MODEL TO MANAGE RISKS FROM EARLIEST STAGES TO INCREASE PROJECT SUCCESS

1 NORULHAIDA ABDULLAH, 2 MARZANAH A. JABAR, 3 FATIMAH SIDI, 4 YUSMADI YAH

1 Msc. Student, Department of Software Engineering and information System, Universiti Putra Malaysia, Serdang, Selangor.
2 Assoc. Prof., Department of Software Engineering and information System, Universiti Putra Malaysia, Serdang, Selangor.
4 Doctor, Department of Software Engineering and information System, Universiti Putra Malaysia, Serdang, Selangor.
E-mail: 1 adiahluron937@gmail.com, 2 marzanah@upm.edu.my, 3 fatimah@upm.edu.my,
4 yusmadi@upm.edu.my

ABSTRACT

Today, managing a software project is so complex and risky. Software project is unique compare than physical project out there because it faces a lot of risks and 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. In this research, the problem is how project failure can be minimize or avoided. To solve it, an early prediction model with proactive signal messenger to detect risk in earliest stages. The model was validated 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 earlier and helps to increase project success.

Keywords: Software Project Management, Software Project Risks, Early Warning Signs, Early Prediction Model and Risk Management

1. INTRODUCTION

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]. 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]. 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.

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]. 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 challenge of managing project is to ensure project success. Our research objective is managing risk in earliest stages before it become costliest. In managing risk’s problems, if problem detected at later stage of the project will be costlier and risky to fail. In this research we focused what the symptoms of project failure and we analyzed previous model and framework to find solution to decrease project failure and increase project success. In this study,
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.

Based on previous research of software risk management, they explained that by analyzed threats to success in example risks action can be taken to reduce the chance of failure of a project [1]. 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].

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 research methodology and then we presented the proposed model and gaps in section 4. After that, in section 5 we presented the conclusion and future study. Lastly, in section 6 we discuss 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. 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.

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

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].

Fig. 3: Interconnectedness Concepts (By Nikander (2002))

A. Conceptual Model

Based on literature review above, we come out with conceptual model. In our conceptual 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.

3. 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]. SDRM has five stages which is construct conceptual framework, develop system architecture, analyze and design the system, Build the prototype system, objective and evaluate the system. 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]. Based on Nunamaker and Chen (1990) idea, five steps of research methodology were proposed as shown in figure 4 below.

Fig. 4: The Research Process Proposed By Nunamaker And Chen (1990).

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 I while proposed attributes are shown in Table II.

<table>
<thead>
<tr>
<th>Metrics</th>
<th>Unit</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Efficiency</td>
<td>Defect Removal Efficiency (DRE) = E/(E+D)</td>
<td>Efficiency to detect risks in every phases via signal.</td>
</tr>
<tr>
<td>Effectiveness</td>
<td>Phase Correction Effectiveness for Phase i (PCE) = Number of phase i errors / (Number of phase i errors + Number of phase i defects)</td>
<td>Effective in decrease project failure and increase project success.</td>
</tr>
<tr>
<td>Reliability</td>
<td>Failure Rate (FR) = Number of failure/execution time</td>
<td>Reliable to manage project well than before</td>
</tr>
</tbody>
</table>
Table II
The Proposed Attributes

<table>
<thead>
<tr>
<th>Attribute</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of risks</td>
<td>Number of risk in software project</td>
</tr>
<tr>
<td>Phases</td>
<td>Number of phases include in software process</td>
</tr>
<tr>
<td>Percentage of project success</td>
<td>Percentage of software project success to fulfill user requirement</td>
</tr>
</tbody>
</table>

Table III 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 III
Experiment Result Used To Validate Metrics And Attribute Of Table I And Table II

Calculation for Total of Acceptance Status:-

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

Formulae: \( \frac{n^f}{t} + \frac{n^u}{t} = \frac{(n^f+n^u)}{T} \)

So, total average for this tool ‘agree’ is \( \frac{200}{400} \times 100 = 50\% \)

5. PROPOSED MODEL

Based on conceptual theoretical model above, we improved and motivated to propose an early warning model which added some features as requirement of our study. The contribution of our proposed framework is its had signal name as proactive signal which is will give alert message early in every stages of software development
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.

### A. 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 frame works with previous framework 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.

6. CONCLUSION AND FUTURE STUDY

The proposed model will detect the symptoms or 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. In the future, Early Prediction Model can do some modification and may will be use in others platform such as in business, education, social media, corporate, commercial and etc. and evaluate whether to get satisfactory result as is software project management.

7. DISCUSSION

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.

REFERENCES:


Rocha et al. (Eds.): Advances in Information Systems and Technologies, AISC 206, pp. 523–534. DOI: 10.1007/978-3-642-36981-0_48.

