Efficacy of Risk Assessment Models in Software Development

1Zafar Nasir, 2Noor Ahmed Shaikh, 2Ghulam Ali, 1Shaheen Khan

1National University of Computer and Emerging Science, Karachi Campus
2Department of Computer Science, Shah Abdul Latif University, Khairpur

Abstract: It is a fact that management of software projects has always been challenging and number of them results in failure that in return jeopardizes organization’s prospects. One of the reasons of high failure rate is that critical steps to assess and manage software project risks are not properly adapted. If risks are improperly assessed and prioritized, then it may become a precursor to waste of available time and money, it may also lead to dealing with those risks which are not likely to occur. In this research paper different risk assessment models have been studied, analyzed and compared, to determine their efficacy in the context of prevailing local software development environment. Therefore, models are correlated with the maturity of organization’s processes, using CMMI. Based on the study a model has been derived and suggested. Recommended risk assessment models will be useful for proper risk assessment early in project life cycle with a low budget incurred on risk assessment.

Key words: Risk assessment, risk assessment models, top risk, formal risk assessment, CMMI, CMMI and risk assessment

INTRODUCTION

Research shows that one of the most important reasons of high failure rate is lack of interest of the project managers for proper measures to assess and manage the risks involved in software projects (Mark Keil, 2008). Risk assessment is a major step on which risk management lie, and for proper risk management it’s very important to perform proper risk assessment. If risks are improperly assessed and prioritized, then dealing with those risks which are not likely to occur can waste time and money. Therefore it’s always better to prioritize risk for further assessment and management. This research work comprises of discussing existing risk assessment models, analysis and comparison among them, and the effect of local organizational factors on the efficacy of models based on the CMMI practices an organization is following.

2 Background and Literature Review:

It is commonly known that risks are those events that may harm or put negative impact on a project. Risk, if it does take place, has impact on cost, scope, time and quality of the product.

Risk assessment is the backbone of the risk management process because this is the stage where organizations decide that whether to spent on specific risks mitigation or not. If it is not properly done you may invest in addressing a risk, which may not occur, and ignoring the one, which have great chances of occurrence that ultimately results in devastating the some or all project objectives.

The purpose of risk assessment is to analyze risk to and prioritize them in meaningful order and establishes a probability and impact matrix to find the risk exposure.

Risk Exposure = P x I

(1)

Where, P = Probability of risk occurrence, and I = Impact of the risk if it occurred.

3 Risk Assessment Models for Software Development:

Some of the research in the area of risk assessment has been sighted as fruitful for software industry in assessing risk. In this section three existing models for software risk assessment will be discussed. Models are a framework for Identifying Software Project (FISPeR). A Formal Risk Assessment Model for Software
Evolution (FoRAMSE) (Nogueira, J.C., 2009) and Software Risk Assessment Model (SRAM) (Say-Wei Foo, 2009).

3.1 A Framework for Identifying Software Project Risk (FISPeR):

Boehm’s work is more influential on practitioners than any other’s work. His “top 10 list of software risk items” was found to have two problems (Henri Barki, 2007). First is this that projects and environment are quite different from the time when Boehm presented his work. Second, there is tremendous change in technology and organization structures both. Therefore a research to identify universal risk list in current scenario has been performed with the two objectives. First, to find factors, which project managers, perceive as risk and also identifying that which factors are more important in view of project managers. Second, to categorize risk factors in a way, that common mitigation strategy can be used for each category. Establishing correlation between risk importance and perceived level of control was found important; because even the risk is high, low perceived level of control allows little for mitigating that risk.

3.1.1 Characteristics of FISPeR Model:

This simple and easy to implement model focuses on subjective risk assessment of top risk items by experts. Top eleven list identified by the model is based on expert judgment, but it is quite reliable as the study performed by a panel of experienced software project managers from different parts of the world, representing common top risk items in different work and organizational environment. It is correlating importance of risk and level of control a project manager has on it. It suggested a framework for risk categorization with mitigation strategies for each category.

3.2 A Formal Risk Assessment Model for Software Evolution as FoRAMSE:

In case of evolutionary software development, business and product requirement often change as development proceeds. In fact the evolving perspective has now become the norm in project management and as such a question of schedule and cost overrun becomes critical. The answer lies in implementing proper risk assessment. Current early risk assessment is an amorphous problem, which relies on individual human judgments and impractical hypothesis such as, unchanging necessities and work breakdown structures. To address this problem risk assessment has to be more structured, targeted and objective.

Existing models do not consider requirement volatility aspect which is an essential nature of evolutionary projects. Some other important factors were also ignored like personnel volatility, complexity of the project and productivity of the team working on the project. FoRAMS model addresses all such problems. This model is based on requirement volatility, complexity and efficiency (Bee Bee Chua, 2009).

\[
\text{Requirement Volatility (RV)} = \text{Requirement Birth Rate (BR)} + \text{Requirement Death Rate (DR)}
\]

Where;

\[
\text{BR} = \frac{\text{NR}}{\text{TR}} \times 100
\]

\[
\text{DR} = \frac{\text{DelR}}{\text{TR}} \times 100
\]

Where,

\(\text{NR} = \text{Number of new Requirements}\)
\(\text{DelR} = \text{number requirements deleted}\)
\(\text{TR} = \text{Total number of Requirements}\)

Large Granularity Complexity (LGC) metrics can be used for calculating complexity.

\[
\text{LGC} = O + D + T
\]

Where, \(O = \text{No. of atomic operators (function or state machine)}\)
\(D = \text{No. of data stream (data connection between operator)}\)
\(T = \text{No. of abstract data type required for the system}\)

Efficiency (Productivity) can be calculated as:

\[
\text{EF} = \frac{\text{Direct Labor Time}}{\text{Idle Time}}
\]
According to the model a random variable \( x \) is said to have a Weibull distribution with the parameter \( \alpha \), \( \beta \) and \( \gamma \) (with \( \alpha > 0 \), \( \beta > 0 \)) probability distribution function (pdf) and cumulative distribution function (cdf) are of \( x \) are respectively.

\[
\text{pdf: } f(x; \alpha, \beta, \gamma) = \begin{cases} 
0, & x < \gamma \\
\left(\frac{\alpha}{\beta}\right) (x - \gamma)^{\alpha-1} \exp\left(-\left(\frac{x - \gamma}{\beta}\right)^\alpha\right), & x \geq \gamma 
\end{cases}
\]

\[
\text{cdf: } F(x; \alpha, \beta, \gamma) = \begin{cases} 
0, & x < \gamma \\
1 - \exp\left(-\left(\frac{x - \gamma}{\beta}\right)^\alpha\right), & x \geq \gamma.
\end{cases}
\]

Fig. 1: PDF with Three Parameters

According to the model:

If (EF > 2.0) then \( a = 1.95; \)
\( g = 22 \times 0.32 \times (13 \times \ln(LGC) - 82); \)
\( b = g / (5.71 \times (RV - 20) \times 0.046); \)
else \( a = 2.5; \)
\( g = 22 \times 0.85 \times (13 \times \ln(LGC) - 82); \)
\( b = g / (5.47 - (RV - 20) \times 0.114); \)
end if;

3.2.1 Characteristics of Model:

This model is perfectly suited for projects, which are evolutionary in nature. The results of the model can be validated by COCOMO.

3.3 Software Risk Assessment Model as SRAM:

This model addresses the problem of failure to develop software within schedule and budget. It also addresses another issue that either the software is according to specification and the customer is satisfied.

It considered nine factors that raise different type of risks for the project and affect software schedule, cost and quality. Factors are software’s complexity, staff’s involvement in the project, reliability targets, product needs, estimation process, monitoring process, development process adopted, software usability and development tools. It correlates the Customer Feedback Index (CFI) with the project overall risk and provides following indication. For example if CFI < 4, project is unacceptable, if it is between 4 and 8 then the project may be ended with extensive plan and substandard. If it is between 12 and 8 then project completed within due dates in budgets and as per terms and specifications.
For assessment the risk assessor (usually project manager) answers the questionnaire, suggested by SRAM, according to the nature of the project. Retrieve the numerical value associated with the selected choice. Obtain the normalized value for each of the nine risk element by summing the numerical values of the question assessor attempted and by dividing the total number of question attempted. Then calculate the normalized risk value for the project by using following formula:

$$\text{Normalized } R = R^n = \frac{(R - R_{\text{min}})}{(R_{\text{max}} - R_{\text{min}})}$$  \hfill (9)

3.3.1 Characteristics of the Model:
It provides an objective numerical value for nine areas of the project and also suggested a formula to assess of overall project using the nine-risk element of the project. It can be calibrated. It correlates the risk value with the customer feedback index, which is an indication of the customer feedback that relates to the quality of the software. It has less empirical value when it comes to relating risk value with customer feedback index as results are based on similar type of projects.

4 Effect of Local Organizational Factors on the Efficacy of Models:
Choosing the right model for risk assessment is important, as time and resources are utilized in risk management process and if wrong model is selected, then, rather cutting down the budget and schedule it can increase it. Processes are the major local organizational factor that can affect the efficacy of a model and CMMI is one of the standards that reflect the level of organization in terms of process maturity. The knowledge that organization is following what standards and practices of CMMI can facilitate the activity of selecting right risk assessment model.

4.1 CMM and Risk Assessment:
In software engineering, software process improvement (SPI) is an important concept and it benefits software project risk management significantly (Chun-guang Pan, 2008). Capability Maturity Model (CMM) is one of the most popular process improvement models, developed by Software Engineering Institute (SEI). CMMI (a variation of CMM) enables organizations to measure their “maturity” on a scale of 1 to 5 (defined as initial, repeatable, defined, managed and optimizing) in doing software engineering. Improvement is achieved by action plans for poor areas. Risk management is covered at third capability maturity level that provides an infrastructure for many services, shown in Figure 2.

Fig. 2: Risk management process, based on CMMI model

4.1.1 Models for Level 1 Organization:
Organizations at level 1 define few processes, lack sound management practices and heavily depend on individuals capabilities. Process capability is unpredictable as it is constantly changing with the work in progress. Organizations at level 1 lacks mature project management processes including risk management process and do not support proper risk assessment. At level 1, investing cost and resources heavily on risk
assessment and management is highly risking, as in crisis planned procedures are abandoned to meet the
deadline there fore, it may have a knock on impact on the organization.

As project managers (PMs) are highly dependable therefore it’s fine to use FISPeR, as it is subjective yet
reliable and it relies on project manager’s expertise of relating risk and level of control he has on it. Once a
P.M identify this correlation he can apply the suggested mitigation strategies. FoRAMSE can not be
implemented in an organization which is not following CMM level 2 standards and practices. As
implementation of this model needs measures and metrics which can not be achieved if proper Requirements
Management which is one of the KPA of CMMI level 2 is not in practice.

4.1.2 Models for Level 2 Organizations:
At level 2 basic project management processes are established, planning and managing of new project is
based on historical data and earlier success can be repeated. Therefore informal risk assessment can produce
more reliable results than at level 1. Like level 1, level 2 doesn’t have support for properly structured and
documented risk assessment, though FISPeR model can be used. Unlike level 1, organizations at level 2 can
use FoRAMSE model as they implement requirement management process. Software Product Engineering KPA
of level 2 recommends that the requirement documents be managed through version-control and change control
practices, this can help in calculating metrics which are required for FoRAMSE model.

4.1.3 Models for Level 3 Organization:
Level 3 organizations can be summarized as standard and consistent because both software engineering
and management are stable and repeatable. An organization which has a defined set of standards for risk
management processes and provides tools support for implementing these standards is more mature than an
organization with only informal process definitions.

One can not rely on un-documented risk related metrics and lessons learned, which are remembered by
individuals, it must be documented properly in a system repository so that better practices can be repeatable
and bad practices can be improved, and this is what CMM is all about repeatable practices and continuos
improvement.

Risk assessment and management is a part of project management and project management is a defined
activity of CMM level 3 KPA ‘Integrated Software Management’. Level 3 ensures that the risk assessment and
management process is properly implemented so that success of previous projects can be repeated. A process
which is already matured to manage and assess risk has higher probability to deliver project on time and within
budget. Therefore it can be stated that organization at CMMI level 3 and higher, fully supports risk
management and thus all the presented models can be applied.

5 Validating Models in Different Organizations:
Discussed models are validated in different organizations which are following different CMMI level
standards and practices. A survey has been conducted to validate FISPer and SRAM models.

5.1 Validating FoRAMSE Model:
The model says that its best suited for evolutionary software but it has been applied in a CMMI level 3
organizations, on a project which is incremental in nature. It was found that this model might not produce
reliable estimates of schedule for such types of software. But it doesn’t mean that the model is not effective,
there are some indication found which shows that the model can be applicable on evolutionary software
development. It was identified that for producing Weibull curve the parameter $\beta$ should be much greater than
$\gamma$, and in the model context it’s only possible when requirement volatility is very high which is inherent in
the nature of evolutionary software development. Therefore it is an indication that this model would be
effective for evolutionary software development. This model was validated and calibrated by the author of
research paper, use of this model was also found in other research work (Jiamthubthugsin, W., 2006). During
implementation, some problems with the model were also identified. It is complex, calculating parameter
required by the model like requirement volatility, complexity and efficiency requires understanding of other
different parameters, which if calculated incorrectly results will also be incorrect and the complete effort will
be wasted. It requires more time to implement, while project managers are usually short of time. It is not
suitable for software projects other than evolutionary in nature. On the other hand, some of the shortcomings
can be manipulated. This model should be use only when requirements are changing frequently and the project
is complex, and some objective assessment is required. Use of automated tools in software development, can
automatically calculate the parameter required by the model.
5.2 Validating FISPeR Model:
A survey has been conducted in the software industry, this survey considered two aspects of the risk item list presented in FISPeR model. Table 1 shows the perceived level of control a project manager has on the listed risks, where level 0 means that project Manager has no control over this risk, level 1 means that he can control risk to some extent, while Level 2 implies that PM can control risk completely.

Table 1: Project Manager's Perceived Level of Control on FISPeR Risk Items

<table>
<thead>
<tr>
<th>Risk</th>
<th>Level 0</th>
<th>Level 1</th>
<th>Level 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Lack of top Management commitment to the project</td>
<td>25% (4)</td>
<td>68% (11)</td>
<td>6% (1)</td>
</tr>
<tr>
<td>Failure to gain user commitment</td>
<td>25% (4)</td>
<td>43% (7)</td>
<td>31% (5)</td>
</tr>
<tr>
<td>Misunderstanding the requirements</td>
<td>12% (2)</td>
<td>37% (6)</td>
<td>50% (8)</td>
</tr>
<tr>
<td>Lack of adequate user involvement</td>
<td>18% (3)</td>
<td>81% (13)</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Failure to manage end user expectations</td>
<td>0% (0)</td>
<td>56% (9)</td>
<td>43% (7)</td>
</tr>
<tr>
<td>Changing scope/objectives</td>
<td>31% (5)</td>
<td>37% (6)</td>
<td>31% (5)</td>
</tr>
<tr>
<td>Lack of required knowledge/skills in the project</td>
<td>18% (3)</td>
<td>25% (4)</td>
<td>56% (9)</td>
</tr>
<tr>
<td>Lack of frozen requirement</td>
<td>31% (5)</td>
<td>43% (7)</td>
<td>25% (4)</td>
</tr>
<tr>
<td>Introduction of new technology</td>
<td>18% (3)</td>
<td>56% (9)</td>
<td>25% (4)</td>
</tr>
<tr>
<td>Insufficient/ inappropriate staffing</td>
<td>12% (2)</td>
<td>62% (10)</td>
<td>25% (4)</td>
</tr>
<tr>
<td>Conflict between user departments</td>
<td>50% (8)</td>
<td>37% (6)</td>
<td>12% (2)</td>
</tr>
</tbody>
</table>

Results of the survey reflects the same ranking as suggested by the FISPeR model, this shows that top risks list presented by FISPeR model is reliable. Now the categorization framework needs to be verified. Let’s first consider a single risk item ‘lack of top management commitment to the project’. According to model this risk item has high perceived relative importance and low perceived level of control see Fig depicted from survey results which is also complying with the results obtained from the model.

5.3 Validating of SRAM Model:
SRAM considers nine critical risk elements: staff involved in the development of the project, software’s complexity, requirements of the product, reliability targets, estimation process, monitoring process, development process adopted, software usability and development tools. The SRAM model correlated the overall project risk value with the customer feedback index. For retrieving customer feedback index the model suggested ways, which are out of the scope of this research work. This process was applied to the collected data from some organizations. See table 3:

Table 2: Project Manager's Perceived Level of Importance on FISPeR Risk Items

<table>
<thead>
<tr>
<th>Risk</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
<th>9</th>
<th>10</th>
<th>11</th>
</tr>
</thead>
<tbody>
<tr>
<td>1) Lack of top Management commitment to the project</td>
<td>37% (6)</td>
<td>18% (3)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>0% (0)</td>
<td>0% (0)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>0% (0)</td>
</tr>
<tr>
<td>2) Failure to gain user commitment</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>12% (2)</td>
</tr>
<tr>
<td>Misunderstanding the requirements</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>25% (4)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
</tr>
<tr>
<td>Lack of adequate user involvement</td>
<td>0% (0)</td>
<td>0% (0)</td>
<td>12% (2)</td>
<td>25% (4)</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>6% (1)</td>
<td>6% (1)</td>
</tr>
<tr>
<td>Failure to manage end user expectations</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>18% (3)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Changing scope/objectives</td>
<td>18% (3)</td>
<td>6% (1)</td>
<td>18% (3)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Lack of required knowledge/skills in the project</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>18% (3)</td>
<td>0% (0)</td>
<td>18% (3)</td>
<td>12% (2)</td>
<td>31% (5)</td>
<td>12% (2)</td>
<td>0% (0)</td>
<td>0% (0)</td>
</tr>
<tr>
<td>Lack of frozen requirement</td>
<td>0% (0)</td>
<td>12% (2)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>18% (3)</td>
<td>0% (0)</td>
<td>18% (3)</td>
<td>12% (2)</td>
<td>12% (2)</td>
<td>0% (0)</td>
<td>6% (1)</td>
</tr>
<tr>
<td>Introduction of new technology</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>31% (5)</td>
</tr>
<tr>
<td>Insufficient/ inappropriate staffing</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>50% (8)</td>
</tr>
<tr>
<td>Conflict between user departments</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>0% (0)</td>
<td>0% (0)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>6% (1)</td>
<td>12% (2)</td>
<td>0% (0)</td>
<td>50% (8)</td>
</tr>
</tbody>
</table>

Results of the survey reflects the same ranking as suggested by the FISPeR model, this shows that top risks list presented by FISPeR model is reliable. Now the categorization framework needs to be verified. Let’s first consider a single risk item ‘lack of top management commitment to the project’. According to model this risk item has high perceived relative importance and low perceived level of control see Fig depicted from survey results which is also complying with the results obtained from the model.
Although the model is simple, easy to use its empirical value is low because it based on the results of same types of projects, but the model allows you to calibrate it according to the local project. Another problem with the model is that its questionnaire is quite lengthy and has questions about which project manager might not have deep understanding. A good point of model is that it correlates the overall project risk value to the customer feedback index.

### 6 Derived Software Risk Assessment Model:

Keeping in mind the strategy used by FISPeR model an open ended survey, of organizations standing at different level of CMMI, was conducted. Purpose was to identify more relevant risks and to suggest a framework which can be used by all type of organizations. In the survey project managers were required to specify top three risks which they think can impact the software development project schedule, cost and may cause the project to fail. A number of project managers from different organizations have participated in this survey. Since it was an open ended survey the collected data has high variance in terms of the ranking of the same risks, as different project managers view a similar risk at a different level of priority.

#### 6.1 Strategy Adapted For Identifying Top Risks:

The result of the survey was varied; the same risk was judge by different project managers at different level of priority. To decide that which risks are considered as top risk in local industry a simple strategy has been adapted. Risk occurring at first priority area has been assigned a random weight as three, risk coming at second priority order has been assigned a weight as two, and similarly for risks coming at third priority order weight assigned is 1. Therefore if the same risk occurs in different priority order it will have the total risk weight age value will be calculated by multiplying the risk weight value with the number of people selected it at first priority plus the risk weight value with the number of people selected it at second priority and so on. Then the final risk weight was normalized. See Table 6.

\[
\text{Rank of risk} = (\text{RVF} \times \text{FF}) + (\text{RVS} \times \text{FS}) + (\text{RVT} \times \text{FT})
\]

Where, RVF = Risk value having first priority  
FF = Frequency of risk at first priority order  
RVS = Risk value having second priority  
FS = Frequency of risk at second priority order  
RVT = Risk value having third priority  
FT = Frequency of risk at third priority order

Formula: \( R \text{ (normalized)} = \frac{R - R \text{ (minimum)}}{R \text{ (maximum)} - R \text{ (minimum)}} \)

Managers should first consider those risk which are in quadrant third as they are those risks which have high relative importance and have high perceived level of control. Mitigation strategies based on the nature of risk can be developed. Risks, which are in quadrant one, are those risks on which project manager has very little control; therefore managers cannot do much about them. Steps for implementing the model:
1. Calibrate the risk ranking according to your software.
2. Categorize them in the presented framework format
3. Apply some formal method to those risks which lie in quadrant third

Table 4: Normalized Risk Weights

<table>
<thead>
<tr>
<th>S #</th>
<th>Risk Item</th>
<th>FF</th>
<th>FS</th>
<th>FT</th>
<th>Risk Weight</th>
<th>Normalized Risk Weight</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Requirements Volatility</td>
<td>7</td>
<td>3</td>
<td>2</td>
<td>29</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>Incorrect Estimation of project timelines</td>
<td>3</td>
<td>4</td>
<td>2</td>
<td>19</td>
<td>0.6</td>
</tr>
<tr>
<td>3</td>
<td>Lack of Skilled Resources</td>
<td>2</td>
<td>6</td>
<td>5</td>
<td>17</td>
<td>0.5</td>
</tr>
<tr>
<td>4</td>
<td>Improper Requirements Gathering</td>
<td>2</td>
<td>3</td>
<td>0</td>
<td>12</td>
<td>0.4</td>
</tr>
<tr>
<td>5</td>
<td>Customer involvement</td>
<td>2</td>
<td>0</td>
<td>1</td>
<td>7</td>
<td>0.2</td>
</tr>
<tr>
<td>6</td>
<td>Political, social and market instability</td>
<td>1</td>
<td>1</td>
<td>2</td>
<td>7</td>
<td>0.2</td>
</tr>
</tbody>
</table>

Fig. 4: Top Six Risks in Software Industry

Fig. 5: Risk Categorization Framework

7 Comparison of Models:
    Among three of the models two models were validated and found to be effective in assessing the project risks in view of project managers. The focus of FISPeR is on finding the top risks and then assessing the
project for those risks. It considers that either the risks falls in project manager’s influence, if yes then the manager can take proper mitigation strategies. If manager have high level of control on a risk and that risk has high relative importance than a manager should consider those risk first for mitigation. Because there is very little project managers can do for those risks, which are not in their control. The other two models FoRAMS and SRAM did not consider this factor. All the three models other than FoRAMS are based on subjective assessment. But the problem with FoRAMS is its complexity of calculating the parameters required by the model, besides it’s suitable for software projects, which are evolutionary in nature.

Comparing derived model with FISPeR it was found that derived has an important risk item which is not mentioned by FISPeR that is “Political, social and market instability.”

<table>
<thead>
<tr>
<th>Model</th>
<th>Parameter Complexity</th>
<th>Empirical Value</th>
<th>Early Application</th>
<th>Validated</th>
</tr>
</thead>
<tbody>
<tr>
<td>FISPeR</td>
<td>Low</td>
<td>High</td>
<td>Possible</td>
<td>Yes</td>
</tr>
<tr>
<td>FoRAMS</td>
<td>High</td>
<td>Medium</td>
<td>Possible</td>
<td>No</td>
</tr>
<tr>
<td>SRAM</td>
<td>Medium</td>
<td>Low</td>
<td>Possible</td>
<td>Yes</td>
</tr>
</tbody>
</table>

8 Conclusion and Future Research:

One of the significant rationales behind over budgeted and behind schedule software project is inappropriate risk assessment. This research work is developed with a conscious aim of studying approaches of different risk assessment models and arriving at a solution of their efficacy by comparing them. Two of the models were validated and found to be fairly effective from the input received through an online survey. These models are applicable in the early phases of the project life cycle. It is beneficial to utilize models that are already established, effective and that can be calibrated, instead devising new models from scratch. However, based on the understanding of existing models a model has been derived for organizations standing at different level of CMMI. Processes are the major local organizational factor that can affect the efficacy of a model, therefore it is important to select the right assessment model by relating the CMMI level of an organization with the models. Therefore models are related with different CMMI levels, to facilitate P.Ms for selecting the right models.

Existing research on risk assessment has tremendous potential to grow in many dimensions. As existing models are either too generalized that it would not be beneficial to adapt them or they are too specific, making it difficult to map on a dissimilar projects. Nonetheless, future researcher can work on models for software projects following different process models risks, as a project that follows waterfall or prototyping will be prone to different types of risks than the one which use agile. Developing a general model for all could be very complicated yet challenging, as different types of project depends on different parameters, besides there is a tradeoff between generalization and reliability. Another possible extension is to develop simple and time saving formal risk assessment model to facilitate P.Ms who are generally, heavily involved in other project management and organizational activities or have no background in formal methods. One excellent point of extension is to merge the formal and subjective risk assessment models that can take advantage of both methods and turn out to be as the model that is although easy in implementation but produce objective and reliable outcome.

Needless to say that there is a long way to go before we arrive at a software risk assessment model that will serve software of diverged nature. Scope of developing such a model is extremely wide and challenging.

REFERENCES


