Towards Evaluating the Risks of Scftware Services Outsourcing Industry*

J. Dasgupta[1] & R. P. Mohanty[2]

Abstract

Modern knowledge based economy is fuelled by Information and Communication Technologies (ICT). One of the principal constituents of ICT is software. Software industry can be classified into software products and software services industries. Software services industry provides outsourced software services to clients, and has grown rapidly in the last few decades. In order to maintain sustainable global competitiveness, software services outsourcing industry must successfully counter and mitigate myriad risks in conducting business and sustaining competition. Existing risk assessment methods rely on measuring probability of risk events, which is difficult to measure in real life. This paper presents a more practical method of risk assessment through articulation of risk factors involved at every stage of the software project management, from bid to completion of the project. Use of this method has been demonstrated through a real life case study.

Keywords: Software services outsourcing, Software projects risk management, Software project life cycle, Risk incident, Risk factors, Software project risk index

1.0 GLOBAL SOFTWARE SERVICES OUTSOURCING INDUSTRY: CURRENT STATUS AND FUTURE POTENTIAL

Software industry can be divided into software products industry, and software services industry. Software product vendors (Microsoft or Oracle for example) develop and market software products and may offer services around their own products such as implementation or customization. Software services vendors do not develop their own products for marketing (although they may develop products of their own to augment their service capabilities) and only provide software related services to other organizations.

From the software user or client point of view, software services can be obtained from in-house departments, or from a vendor. Procuring software

* Received November 7, 2008; Revised August 17, 2009

1. Research Scholar, ITM - BIT Collaborative Research Program, Risk Management Group, Patni Computer Systems Limited, Mumbai, email: jyotidasgupta@yahoo.co.in

2. Dean, Institute of Technology Management, Navi Mumbai, email: rpmohanty@gmail.com

services from vendors is termed outsourcing. Software Services Outsourcing (SSO) is a common practice any where in the world. By the beginning of this century, over three quarters of large firms were engaged in long term software services outsourcing contracts. Primary drivers for outsourcing are desire to reduce costs or increase profitability, desire to focus on core competency, access to special expertise, speeded up delivery, relieving resource constraints and many others (Davies, 2004). Estimates vary but most estimates agree that the global outsourcing market is pegged upwards of a trillion US dollars. According to one study, 57 % of this market was serviced by the US, 4% by India, 3% by China, Philippines and SE Asia, and 36% by other countries. (Blackbook, 2007) Another study concluded that the worldwide IT services spending aggregated nearly USD 1.7 trillion, and computed a growth of 7.3 per cent over the previous year. Two major components of this market were found to be (NASSCOM, 2008):

a) Software Services Outsourcing: Software and other services including Business Process Outsourcing (BPO) at USD 1.2 trillion - over 71 per cent of the total spend in 2007.

b) Hardware spends, at USD 478 billion, accounted for over 28 per cent of the total worldwide IT services spending in 2007.

Principal objective of this paper is to take a comprehensive look at some of the important concepts in risk identification and measurement that apply to 'Software Services Outsourcing' (SSO) industry from vendors' perspective, and to propose a risk-handling framework for the SSO industry. This paper also deals with *a priori* articulation of risks involved in software project management from start to finish of the entire project management life cycle, and presents a method for risk assessment. This has been demonstrated through a real life case study.

2.0 WHY SSO INDUSTRY MUST MANAGE RISK?

It is an established fact that SSO is a mega growth industry. This industry has seen scores of young technocrats build multi million dollar enterprises. SSO business models are very sophisticated, and have given extra momentum and impetus to innovative financing models including venture capital and equity markets, both private and public. Although most IT companies begin as 'start ups' answerable only to their early stage financiers, they soon become answerable to millions of shareholders after going public. Revenue predictability and preserving shareholder value become most important to such companies.

SSO companies have witnessed higher than average revenue, and margin growth over the last several years. Increasingly, however, the growth rates are under threat. Recent events such as the 'dotcom bust', terrorist strikes including 9/11 or the more recent meltdown in global financial markets have meant cautious capital expenditure spends, and postponing technology upgradation plans.

Consequently, ability of SSO industry to face adversities must increase to maintain sustainable global competitiveness. Other more traditional industries such as Finance or Manufacturing have been using sophisticated risk management techniques for decades; and have benefited immensely. Indeed, it is impossible to visualize industries such as Banking or Insurance without risk management systems, although recent events in these industries illustrate the need to further strengthen these systems.

SSO vendor is a business entity and therefore effective analysis of financial performance is of much importance to it. Assessing risks and incorporating the same in the final decision is an integral part of financial analysis (Chandra, 2003). Risk management techniques are used in some major enterprises and considerable knowledge base exists on how to effectively assess and mitigate risks.

SSO units the world over are also implementing risk management systems. In fact, industry associations and standards organizations such as the International Standards Organization (ISO), Software Engineering Institute (SEI), Project Management Institute (PMI) and several other such organizations have devised risk management frameworks that have been in use for some years now.

3.0 TYPOLOGIES OF SSO SERVICES

Software projects undertaken by most SSO companies can be classified into the following categories according to the nature of services provided:

a) Business Transformation and Consultancy Services

b) Application Services: Applications are software used by client organizations that are custom designed for in-house use by the clients themselves and not available for others:

   i. Development: Developing and implementing new applications

   ii. Maintenance: Enhancements, modifications and bug fixing of in-house applications

iii. Re-engineering: Making application systems work with or without additional features on a new technology or platform

iv. Localization/ Globalization: Making the software serve different geographies and languages

c) Software Products: are developed by software product companies, for use by their customers. These are also referred to as 'packages'.

i. Development: Developing new or next generation products

ii. Sustenance: Similar to application maintenance, but usually tasked with maintaining several past versions through out the life cycle of the software product.

iii. Re-engineering: Making products work with or without additional features on a new technology or platform

iv. Localization/ Globalization: Enabling products to serve different geographies and languages

d) Package Implementation: Software products such as ERP, BI Tools etc. require extensive customization for client specific purposes. This work is referred to as package implementation.

e) Testing: These projects require extensive manual or automated testing of software applications or products.

f) Production Support: These projects monitor and fix applications in use (often referred to systems in production) such as an online credit processing system, or HR management system etc. on 24-x7 basis. Often involves minor modifications or bug fixes as well.

g) Engineering and Hardware design Services: Offer services such as digitization, CAD/ CAM, PCB design, VLSI design etc.

h) Business Process Outsourcing: Data processing and call handling services for industries such as financial services, airlines, hospitality etc.

Many SSO projects are done using presence of coordinators and analysts at client locations (called onsite) with significant portion of software work done at company owned development centres and are referred to as onsite-offshore projects.

Most SSO projects are usually billed on the following basis, with variations such as per transaction, or profit sharing etc.:

a) Time and Material (T&M) billing model is used when the scope of the project can not be defined precisely, or for repetitive maintenance or production support type work that go on for years. Services are charged on per person hour or person day basis.

b) Fixed Price (FP) projects are used when the scope can be defined with reasonable precision and efforts/ schedules estimated in advance.

4.0 SOFTWARE PROJECT LIFECYCLE

Mohanty *et al.* (2005) have shown that Research & Development (R&D) projects go through three phases namely basic, applied and development. In basic phase, knowledge concerning technology and processes are collected, and economic viabilities of different process plans are evaluated. In applied phase; laboratory research, feasibility study etc. are done, and in Development phase, new products are developed as per the plans drawn up and learning from the earlier phases.

Somewhat similarly, software projects can be divided into three stages: Proposal, Finalization (or Contract Acceptance for vendor driven projects) and Execution, The importance of various attributes and criteria varies with the phases.

4.1 Proposal Phase

In this phase, project proposals are evaluated for benefits and attendant risks. For outsourced (vendor driven) projects, a decision to bid or not bid may be taken, based on the benefits that would accrue to the vendor organization vis-à-vis the risks that the vendor organization would have to accept. Based on such evaluation, organization may decide to go ahead, or not go ahead with the project.

4.2 Finalization or Contract Finalization Phase

In this phase, the organization takes a final decision to proceed with the software project, with modifications in scope or plans as appropriate to reduce risk of project failure. This phase is often named 'Contract Finalization' phase for vendor driven projects.

4.3 Project Execution Phase

In this final phase, projects are executed as per the plans drawn up in the previous phases. In this phase, requirements capture, design, quality, capacity planning etc. are considered.

These various phases of a software project are termed 'Software Project Life Cycle'. Many life cycle models have been proposed in the past (examples include Sequential or Waterfall, Prototyping, Spiral or Iterative, Object Oriented, Cleanroom, 4th generation techniques such as Agile or Extreme programming etc. (Pressman, 2005; Humphrey, 2006) and every model has advantages and disadvantages over each other. Latest technology trends such as Service Oriented Architecture are challenging older life cycle models, and newer models are being proposed regularly. Some of these newer models take a more holistic and complete view of the project life cycle and includes business aspects as well (Strosnider *et al.* 2008). The main objective in defining the life-cycle is to control the software engineering process to minimize risk.

Most software project life-cycles focus only on the phases 'Requirement Gathering' onwards. In reality, software project life-cycle originates at the time the software is first conceived. Although researchers have recently started focusing on this phase of projects as well (Abrahams *et al.* 2009) there is a need to start assessing project risks right at this time, to allow a 'go-no go' decision based on the benefits and attendant risks.

In this paper, we focus on risk assessment right from the first phase of a software project, namely the Proposal phase. After a literature review of software project risks assessment methods, we classify the different methods into three categories Qualitative, Quantitative and Semi-Quantitative methods (Mohanty *et al.*, 1993). We then describe a risk hierarchy applicable in the three project phases, following a scheme proposed by Mohanty *et al.* (2005). Next, we describe a risk assessment tool that has been tested in a software projects company. Test results are presented through a case study.

5.0 SOFTWARE RISK ASSESSMENT

Problems with software projects, whether in-house or vendor driven, have been reported over the past several decades. This same period has witnessed the emergence of a new mega industry- the software industry; and has also seen a steady flow of advice for software project managers on project management, software methodologies, and risk management techniques. Risk management practice has been identified as one critical factor of the success of software projects (Taylor, 2007). IEEE has published risk management glossary to provide guidance and a degree of uniformity in the terminology used by software risk professionals (Fairley, 2005).

Objective of risk management is to identify projects that are less likely to meet objectives. Typically software project objectives may be defined as one or more of several factors such as cost, schedule, effort, quality, service levels such as down time etc. Risk management has grown increasingly popular in recent years due to recognition that risk should be actively managed as an important attribute of business performance (Calandro *et al.*, 2008). An Enterprise Risk Scorecard modeled on Balanced Scorecard concept (Calandro *et al.*, 2006) is an example of that. Few of the many other interesting models for measuring and representing software related risks are Threat Modeling (Ingalsbe *et al.*, 2008), Security Meter (Sahinoglu, 2005) or a CORAS (http://www2.nr.no/coras) based model for Information Security Risk (Yong *et al.*, 2008). Risk management systems are expected to identify and assess various risks encountered by a project, or a group of projects. Quantitative assessment or measurement of risks is an important component of risk management systems. Reporting risks in a manner that can be easily understood by software project professionals is another requirement of risk management systems. Such reports can be used to accurately assess the risks that expose the project to higher damage, allowing management to initiate mitigation actions in time to limit the possibility of loss.

5.1 Various Software Project Risks

Risk is defined as 'the possibility of something bad happening in future; a situation that could be dangerous or have a bad result; any business venture has an element of risk' (Oxford, 2005). The major ingredient of risk is uncertainty. If the consequences of an action or a decision depend on the possible occurrence of other events, then such action or decision is termed as 'risky', if nothing can be told in advance whether those events will happen or will not happen. (Copas, 1999). Although several risk classification schemes exist (COSO, 2004; Fight, 2004), a simple classification that is necessary for most purposes, and is often used by professionals, recognize three major types:

a) Market Risk: Prices will move in a way that has negative consequences

b) Credit Risk: A customer, counter party, or supplier will fail to meet its obligations, and

c) Operational Risk: People, processes or systems will fail, or an external event (e.g., earthquake, fire etc) will negatively impact the project.

In general, risk managers consider market risk and credit risk as financial risk, and group all other risks as part of operational risk (Lam, 2003). Dominant risks in software projects are best classified as operational risks. Researchers in the area of software risk management have been very active. Many give credit to Barry Boehm and Tom Demarco for laying the foundation of Software Risk management (Boehm, 1989; Boehm and Demarco, 1997).

One of the initial attempts to identify risks in software projects was made by Henri Barki, Suzanne Rivard and Jean Talbot (1993). They identified 24 risk factors after a survey of 120 software projects. This list remains a much respected and cited list till date. This list was revalidated by Jiang *et al.* (2002) over 152 software projects, and 6 factors were found through Exploratory Factor Analysis. These factors were found to be technical acquisition, project size, lack of clarity of role definition, lack of user experience on system development, lack of user support, and lack of team expertise.

Several researchers have provided further insight into risks found in in-house or outsourced software projects (Mulcahy, 2003; Ethiraj *et al.*, 2005; Gefen *et al.*, 2008). Of particular interest is the research by Hazel Taylor (2007). She has pointed out that while many risk factors for IT projects in general have been identified in the literature little thought have been given to the risk factors that are of higher concern for managers of vendor driven (or outsourced) projects. She has identified 43 top risks in 'ERP implementation' type outsourced projects in Hong Kong. These 43 risks can be classified into 6 factors i.e. project management, solution, technology, relationship, location and commercial environment. Additional software project risks have been identified by international standards such as the International Standards Organization (www.iso.org) or the Integrated Capability Maturity Model CMMI® for software services from the Software Engineering Institute, Carnegie Mellon University (SEI, 2009).

Researchers have identified hundreds of software and IT related risks. Assessing risks for all the factors are neither feasible, nor useful for most software projects. We have therefore consolidated the major factors emerging from these above body of knowledge into 5 major categories. The list, and explanation of these categories, is presented in Table 1. These 5 categories have been used in the case study presented in this paper, and have been found to be all encompassing as well as applicable for all types of software projects. The 5th category is applicable only to vendor driven projects.

5.2 Measuring Software Project Risks

Several researchers (Cong *et al.*, 2008; Dia *et al.*, 2008; Kahraman *et al.*, 2007; Saghafian *et al.*, 2005; Rainer *et al.*, 1991; Bellman *et al.*, 1970) have attempted to identify and assess IT and software related risks both qualitatively and quantitatively.

We can divide these various methods into three categories, following a classification scheme proposed for management decision justification methods (Mohanty *et al.*, 1993):

a) Qualitative: These methods generally stress long-term strategic concerns of the organization. They expose the decision makers to a variety of factors and attributes which otherwise would have been normally ignored. However, in real-life situations, it is difficult to incorporate all the factors and attributes since their effects are more intuitive than quantitatively measurable.

b) Quantitative: These methods make use of various scientific decision making models like monetary, engineering, economic or mathematical models. The complexities of such models some times become deterrent to their use.

Table 1. Software Projectl Risk Factor Categories

1. Requirements Clarity	How clearly requirements of the software project have been understood, and how stable these are likely to remain over the entire project life cycle.
2. Solution Complexity	How complex the software solution being developed is likely to be.
3. Execution Capacity	Ability to organize necessary skills, infrastructure and logistics
4. Customer Related	How supportive is the end customer, whether internal or external, likely to be; and whether project funding is likely to be an issue
5. Contract Related	Are there clauses in contract that would be difficult to meet?

c) Semi-quantitative: These methods are used to transform subjective judgments into simple measures. Some of the widely used methods are Linear Additive Models (LAM), Multi-criterion Q Analysis II (MCQAII), Analytic Hierarchy Process (AHP) (Mohanty *et al.*, 1993), Fuzzy Analytical Network Process (Fuzzy ANP) (Mohanty *et al.*, 2005) etc.

5.2.1 Few Quantitative Risk Measures

Annualised Loss Expectancy (ALE)

This method is useful is assessing the risk relating to business continuity plan and disaster recovery planning (Rainer *et al.*, 1991). As per this method, all IT assets needed for the project are listed. Then, potential threats to those assets are analysed along with the loss that would result from the realization of those threats. The vulnerability of each asset to a threat is expressed as some probability of occurrence per year. Multiplying the probability of occurrence per year by the expected loss yields the expected loss per year from a particular threat/ vulnerability pair. The summation of the expected losses represents the total IT risk exposure.

$$\textit{Total IT risk exposure} = \sum_{i=1}^{n} \left(V_{i}\ {}_{x}\ El_{i}\right) \text{....(1)}$$

Where vulnerability V_i is probability of occurrence every year and EL_i is expected loss of the vulnerability pair (i^{th}).

Expected Utility Theory

According to the expected utility theory – the optimal decision is one that maximizes expected utility, which is essentially the product of the probability of the adverse event and the utility (negative loss) resulting if that adverse event occurs. Boehm (1989) proposed an approach, which is in agreement with the Expected Utility Theory, and defined software risk exposure (RE) as:

*RE=Prob (UO)*Loss (UO). ……….. (2)*

Where, Prob (UO) is the probability of an unsatisfactory outcome,

And, Loss (UO) is the loss to the parties affected if the outcome is unsatisfactory.

For most software projects, calculating Prob (UO) was found to be difficult. This is an experience shared by many practitioners, but it continues to be widely used (Mulcahy, 2003). However, this method continues to be the most widely used in software industry.

5.2.2 Few Semi Quantitative Risk Measurement Methods

Fuzzy Set based methods

Bellman *et al.* (1970), Saghafian *et al.* (2005), Mohanty *et al.* (2005) Bottani *et al.* (2006), Kahraman *et al.* (2003, 2007), Cunbin *et al.*(2008), Cong *et al.* (2008), Dia *et al.* (2008) have done extensive research in the area of fuzzy set based methods. However, these methods have not been used in software projects risk assessment. Recently, Analytic Hierarchy Process (AHP), which is a powerful multi-criteria decision making method (MCDM), is being used along with Artificial Neural Network (ANN) technique for assessing risks faced by potentially dangerous equipment such as high pressure boilers, cranes etc. (Zhang *et al.*, 2008).

Semi-quantitative Crisp Risk Index based methods

Dictionary definition of index (Oxford, 2005) is 'a sign or measure that something else can be judged by'. Spiegel (Naik, 2004) defined index as 'a statistical measure designed to show changes in variables or a group of related variables with respect to time, geographic location or other characteristics.' There are several other definitions and indexing methods given by Spiegel, John I. Raffin, A. M. Tuttle, Maslow, Croxton and Cowden, Lawrence J. Kaplan, B. L. Bowley, Horace Secris, G. Simpson and F. Kafka, L aspeyere, Paasche, Edgeworth-Marshall, Fisher and others (Naik, 2004) (Gun, et. al. 2005). These covered arithmetic, geometric or harmonic means.

In this paper, we will use a semi-quantitative crisp risk index to measure software project risks.

6.0 CASE STUDY

A software company was using a risk assessment method based on 'Expected Utility Theory' given in Equation 2 earlier. This method was found to be not very useful, for the same reasons discussed earlier in this paper i.e. computing probability of a risk event was a largely 'gut feel' based exercise, as accurate data was hard to come by. A decision was taken to employ a 'risk factor' based index that did not depend upon computing probability, and relied on the presence or absence of risk factors.

6.1 Software Project Risk Index based on Risk Factors

A risk calculator for use during the entire software project life cycle was constructed. The index was named 'Project Risk Index'. The objectives of this index are to:

a. Divide every project into one of the three categories, namely

   i. *Normal* projects that do not have too many risk factors. These projects therefore can go through reviews with middle level managers during project execution,

   ii. *Risky* projects that have many risk factors present. Definition of *many* would be determined by looking at past project performance data. We would explain this concept in greater detail later in the paper. Risky projects need to go through special reviews by senior level managers.

   iii. *High Risk* projects that have too many risk factors present. Definition of *too many* would be determined by looking at past project performance data. We would explain this concept in greater detail later in the paper. High Risk projects would necessarily need reviews with highest management levels.

b. At proposal stage of the life cycle, decide whether to submit a proposal or not, that is take 'go - no go' decision

c. Have good understanding of the attendant risk factors at the proposal time itself, allowing timely initiation of risk mitigation actions.

d. Monitor the risk factors identified over the entire project life cycle that is through contract and execution stages.

The basic structure of the risk index measurement is given in Figure 1 below.

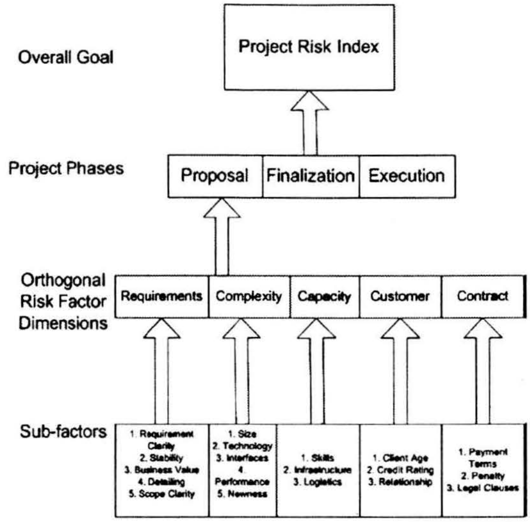

Figure 1. Project Risk Index – Factor Hierarchy

6.2 Risk Event, Risk Factors and Capability

At this stage, it is necessary to clearly differentiate between two terms, Risk Event and Risk Factors. Risk Event is that an adverse event takes place meaning that the risk eventuates. In the case of Software Projects, a Risk Event would mean that the software project did not meet objectives. Project objectives defined for the case study have been described in later sections.

Risk Factors are attributes of the project, which based on deep domain knowledge of more than 30 experienced software project managers, were considered to be factors that increase project risk by being present, and decrease project risk by being not present.

Capability is the usual ability of the organization to successfully execute different types and sizes of projects. Ethiraj *et al.* (2005) describe in great detail how capabilities get developed in software project organizations.

6.3 Methodology

This above risk index measurement structure was devised based on extensive discussions undertaken with more than 30 senior project managers, and risk factors proposed in the literature. These discussions resulted in arriving at the following key considerations for identification of project risk factors:

a) Identify major risk factor categories that are independent with no or minimal interaction between each other. These have been shown in Figure 1.

b) Identify sub-categories within each of these major categories, as shown in Figure 1

c) Assess presence or absence of these factors by asking a series of 'yes' and 'no' questions, that determine presence or absence of these risk factors

d) Do not have subjective and judgment based questions that require potentially imprecise answers such as a scale of 1 to 5, whether crisp or fuzzy

6.3.1. Risk Factor Questionnaire

Accordingly, a questionnaire was devised. The questionnaire is shown in Table 2.

As shown in Table 2, the questions can be answered only in yes or no, making judgmental calls unnecessary. Each 'yes' answer indicated presence of a risk factor, and added 1 to the score. Each 'no' answer added 0 to score. Maximum score possible was 19, which was scaled up to a score of 100, giving a percentage type risk score.

6.3.2. Piloting the questionnaire

As there were 19 questions, the questionnaire was validated through 'retrospective' analysis of 210 completed projects. Managers in these projects were first trained on filling these questionnaires, and were then asked to fill.

While deciding on the projects for the validation phase, 70 projects were chosen that were known to have experienced more than 10% variation in one or more of the following project objectives:

a) Cost

b) Quality

c) Schedule

It was further observed that any of the above three ultimately resulted in the project not meeting its financial target. Hence it was decided to

Table 2: Project Risk Questionnaire

1. Answer all questions in 'Yes' or 'No'
2. Each 'Yes' answer shall add 1 to the risk score

1. Requirements Clarity	1.1 In your opinion, is the requirement unclear 1.2 Has customer stated that the requirement will change 1.3 Is customer unable to explain business value accruing from this project 1.4 Is the requirement presently at high level, and would require further detailing 1.5 Is the project boundary and scope unclear at this time
2. Solution Complexity	2.1 Is it a large project (Note: This would depend upon the organization capability. For this company it was defined as more than 300 person-months) 2.2 Will this project require a technology or product that is new in the market 2.3 Will the solution need more than 3 interfaces to other systems (Note: This number would be dependant of organization's capability) 2.4 Are their performance criteria to be met 2.5 Is the solution technology or domain new to the organization
3. Execution Capacity	3.1 Does the company lack one or more of the required skills in techno, domain or methodology 3.2 Does the company lack required development or testing infrastructure 3.3 Is the necessary logistics (equipment or personnel) difficult to organize in the given timeframe
4. Customer Related	4.1 Is it a new customer, and not a new project for an existing customer 4.2 Does the customer have bad credit rating 4.3 Is our relationship with this customer not very good
5. Contract Related	5.1 Are the payment terms not acceptable to us 5.2 Are there penalty clauses specified 5.3 Are there legal clauses that we not comfortable with

redefine the project success as meeting financial (or project profit) target to within + 10%.

The remaining 140 were able to meet objectives within 10% of the targets set, so were considered to be 'successful' projects. These exact numbers would again depend upon the organization's 'baseline' and past performance data.

6.3.3 Results

Based on the actual scores reported by these 210 projects, a scatter diagram was created. The scatter diagram is shown in Figure 2.The scatter diagram indicated that the questionnaire had good discrimination power to identify risky projects.

a) No project scored more than 70 %, so these scores were classified as 'High risk' score.

b) Looking at the exact scores of projects, imaginary lines were drawn on the scatter diagram. At scores below 35% very few unsuccessful projects were found. Therefore scores between 35 and 70 were considered to be candidates for risky project proposals.

c) Above 35%, very few successful projects were found. So scores below 35% were considered to be scores for non risky or normal projects.

d) Exact distributions have been given in Table 3.

6.3.4 Analysis of the results

As this is a limited pilot, it was important to establish the reliability of the risk score bands. This was done through first creating a contingency table, as given in Table 4.

6.3.4.1 Notations used in the case study

We define the following notations:

N11 = Number of unsuccessful projects

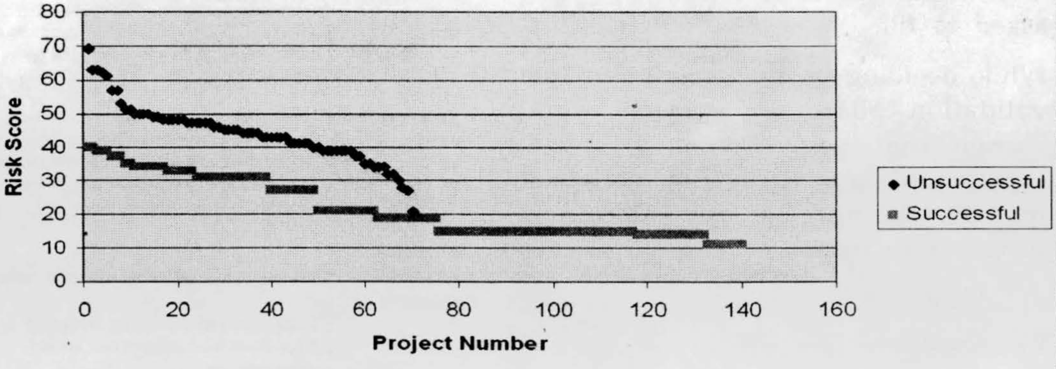

Figure 2: Scatter Diagram

that have risk scores between 35 and 70

N12 = Number of successful projects that have risk scores between 35 and 70

N21 = Number of unsuccessful projects that have risk scores less than 35

N22 = Number of successful projects that have risk scores less than 35

N1. = Marginal of row 1 (N11+N12)

N.1 = Marginal of column 1 (N11+N21)

N2. = Marginal of row 2 (N21+N22)

N.2 = Marginal of column 2 (N12+N22)

N.. = Marginal of Table (N1.+N2. or N.1+N.2)

α = False positive or probability that successful projects are in band 35 to 70

β = False negative or probability that unsuccessful projects have scores less than 35

Ω 1 = Probability that an unsuccessful project would have score between 35 and 70

Table 3: Questionnaire Results

Score Band	Number of Unsuccessful Projects	Number of Successful Projects	Marginals
More than 70%	None	None	
Between 35 and 70%	57	13	
Less than 35%	13	127	

$\Omega2$ = Probability that an unsuccessful project would have score below 35

Ω = Odd of risk event that is the odd that unsuccessful project would have score between 35 and 70, against having a score below 35

Table 4: Results Analysis

Score Band	Number of Unsuccessful Projects	Number of Successful Projects	Marginals
Between 35 and 70%	N11=57	N12=13	N1=70
Less than 35%	N21=13	N22=127	N2=140
Marginals	N.1=70	N.2=140	N..=210

6.3.4.2 Equations used for case study analysis

We calculated the Relative Risk, the ratio of false positive and false negative as

Relative Risk = α/β ..... (3)

As we can see from Table 5, Relative Risk is 2, which is an acceptable number.

We next computed the odd of risk event as $\Omega = \Omega1/\ \Omega2$ ...... (4)

As we can see from Table 6, odd of risk event is 64.3 which is quite high.

Above analysis establishes that the project risk instrument devised is showing good discriminatory power between unsuccessful (Risky) and successful (Non-Risky) projects.

7.0 CASE STUDY FINDINGS

7.1 Findings

Retrospective study described above indicates good discrimination power of this project risk index. However, further 'prospective' studies are required to further calibrate the risk score instrument.

7.2 Contributions

Contributions of this paper are the following:

a) Literature on software risk management presents a bewilderingly large number of risks faced by software projects. This makes it difficult for the software project managers to correctly identify the risks that are relevant to a particular project, or a group of projects. This paper captures these various risk categories, and presents a comprehensive classification scheme for identifying major risks. Using this comprehensive list, software project managers would be able to identify the specific risks to projects more easily and accurately.

Table 5: ·Risks Ratio

Probability of successful projects in band 35 to 70	N12/N1=	0.185714286 α
Probability of unsuccessful projectsin band less than 35	N21/N2=	0.092857143 β
Relative Risk i.e. Ratio of false positive and false negative	α / β	2

Table 6: Risk Odds Ratio

Ω1	Probability of unsuccessful projects in 35 - 70 band	N11/N12	4.384615
Ω2	Probability of unsuccessful projects in less than 35 band	13/127	0.068182
Ω	Odd of Risk Event	Q1/Q2	64.30769

b) A semi-quantitative Software Project Risk Index has been constructed and tested in a mid sized software projects vendor company. This index can be used in both in-house and vendor driven software projects of all types.

The software company has experienced following benefits by using this framework:

i. Projects with higher risk exposure are identified during proposal and contract finalization stages itself. This allows the management to take go - no go decision, or initiate mitigation actions on time ensuring project success

ii. Through effective actions and frequent risk audits on the risk factors identified during proposal and contract finalization stages, risky projects are brought under control quickly and more easily during the execution stage.

iii. Organization wide mitigation actions are made possible as generic difficulties are identified more easily through the risk management process.

7.3 Limitations

a) The methods described in this paper can be easily implemented in both in-house or vendor driven software projects. However, we have not tried generalisability of the methods or the risk management framework presented.

b) More empirical studies are necessary to establish the validity and applicability of these methods over wide range of projects and risk situations.

c) Further research is necessary in using recent advances in Fuzzy Set based methods in constructing risk indices.

7.4 Discussions on the improved method

We can see that by using the above method, software companies:

a) Do not need to find probability of risk events, which is difficult in real life

b) Can use instead a 'risk factors' based method that have causative correlation with project success.

c) Can collect data on these factors across all projects at reasonable cost using automation, and with appropriate training provided to the Project Managers.

8.0 CONCLUSIONS AND FUTURE DIRECTIONS

SSO is a major global industry, and generates huge revenues across the world. It also provides employment to millions. Continued well-being of this industry would require these enterprises to reduce costs through reduction of losses. Effective risk identification and management techniques can be one of the techniques of interest.

In this paper, we reviewed some important risk identification and assessment models currently in use in software and other industries and discussed relative merits and demerits of these models. All statistical models depend on authentic and reliable data, and collecting such data entails cost. We have also to look at practicality of the models in use. We have presented an alternate model that does not require the projects to compute probability of risk incidents, but instead uses a more practical and easy to use risk factor based model.

Summarily, we conclude that SSO industry today is in a growth path barring the current slow-down period. Therefore, it is imperative that SSO industry must focus on quality, cost and innovations. Those apart, the project risks must be recognized more objectively. For that, we will require executive-level analytical skills that can build capability in recognizing, interpreting, and modeling multiple risks in the portfolio of project activities with distinctive management needs. The industry must be prepared to invest in such skills building process beyond its ICT training. Risk management in ICT projects pose a challenge of providing a unified view of diverse elements of risks that is genuinely useful and goes beyond providing a solution to a situation specific project. It is a concept and a set of evolving models and constructs. It requires consideration of three aspects such as; usefulness and breadth of applicability, alternate frameworks and methods, better links between business analysis and technical system analysis. It needs to be driven by a passion for discipline, predictability, and variation reduction. This paper is only a representation in that direction.

9.0 ACKNOWLEDGEMENT

We acknowledge the reviewers for their comments that helped to improve the contents and focus of this research paper. Further, we thank Prof. Brajaraj Mohanty for his suggestions. We are thankful for the support received from the company where the case study was conducted.

REFERENCES

Abrahams, A.S., Macmillan I.C., (2009), IT-DDP: A Novel Methodology for Assuring Economic Value from Entrepreneurial Information Technology Projects, *The Journal of Computer Information Systems,* Spring 2009, Vol. 49, No. 3, pp. 1 - 9

Barki, H., Rivard, S. and Talbot, J. (1993) Toward an Assessment of Software Development Risk, *'Journal of Management Information Systems,* Fall 1993, Vol. 10, No. 2, pp. 203-225

Bellman R. E., Zadeh, L. A. (1970), Decision making in a Fuzzy Environment, *Management Science,* December 1970, Vol. 17, No. 4, pp B141 - B164

Black Book of Outsourcing (2007), Brown-Wilson Group, www.theblackbookofoutsourcing.com

Boehm, Barry (1989), *Software Risk Management,* IEEE Computer Society Press, 1989

Boehm, B. W. and DeMarco, T. (1997) Software Risk Management, *IEEE Software,* May/ June 1997, pp: 17-19.

Bottani, E., Rizzi, A. (2006), A fuzzy TOPSIS methodology to support outsourcing of logistics services, *Supply Chain Management, An International Journal,* Vol. 11 No. 4, pp. 294 - 308

Calandro, J., Lane, S. (2006), Insights from the Balanced Scorecard, An introduction to the Enterprise Risk Scorecard, *Measuring Business Excellence,* Vol. 10 No.3., pp. 31 - 40.

Calandro, J., Lane, S., Dasari, R. (2008), A practical approach for risk-adjusting performance, *Measuring Business Excellence,* Vol. 12 No. 4., pp. 4 - 12

Chandra, Prasanna (2003), *Finance Sense - Finance for Non-Finance Executives, Third Edition,* Tata McGraw-Hill Publishing Company Limited

COSO (2004) *COSO Enterprise Risk Management - Integrated Framework. 2004,* www.COSO.org

Cong, G., Zhang J., Chen T., Lai, K (2008), A Variable Precision Fuzzy Rough Group Decision-Making for IT Offshore Outsourcing Risk Evaluation, *Journal of Global Information Management,* Volume 16, Issue 2, pp. 18-34

Copas, J. (1999) Statistical Modeling for Risk Assessment, *Risk Management: An International Journal,* Vol. 1, No. 1, pp: 35-49.

Cunbin, L., Jianjun, W., Li, l., Xueyan, L (2008), Triangular fuzzy number method for measuring risk element criticality by Credibility degree in project network, *The 2008 International Conference on Risk Management & Engineering Management,* IEEE Computer Society, pp. 498 - 501

Davies, Paul (2004), *What's this India Business? Offshoring, Outsourcing and the Global Services Revolution,* Nicholas Brealey International, UK

Dia, M., Zeghal, D. (2008), Fuzzy Evaluation of Risk Management profiles disclosed in Corporate Annual Reports, *Canadian Journal of Administrative Sciences, August 2008,* pp 237 - 254

Ethiraj S., Kale, P., Krishnan, M. S. and Singh, J. V. (2005) Where do capabilities come from and how do they matter? A study in the software services industry, *Strategic Management Journal,* No. 26, pp: 25-45.

Fairley, R. E. (2005), Software Risk Management, *Software Engineering Glossary, IEEE Software,* May/ June 2005, pp. 101

Fight, Andrew (2004), *Credit Risk Management,* Butterworth-Heinemann, USA

Gefen, D., Wyss, S., Lichtenstein, Y. (2008) Business Familiarity as Risk Mitigation in Software Development Outsourcing Contracts, *MIS Quarterly,* September 2008, Vol. 32 No. 3, pp 531-551

Gun, A. M., Gupta, M. K., Dasgupta, B. (2005), *Fundamentals of Statistics, Volume 2,* The World Press Private Limited, Kolkata - 700073

Humphrey, W. S. (2006), *A Discipline for Software Engineering,* Dorling Kindersley (India) Pvt. Ltd., Delhi - Licensee of Pearson Education Inc.

Ingalsbe, J. A., Kunimatsu, L., Baeten, T. (2008), Threat Modeling: Diving into the Deep End, *IEEE Software,* January/ February, pp. 28 - 34

Jiang, J., Klein, G., Ellis, T. S. (2002), A Measure of Software Development Risk, *Project Management Journal, Project Management Institute,* September 2002, Vol. 33, No. 3, pp. 30-41

Kahraman, C., Cebeci, U., Ulukan, Z. (2003), Multi-criteria supplier selection using fuzzy AHP, *Logistics Information Management,* Vol. 16, No. 6, pp. 382-394

Kahraman, C., Ates, N. Y., Cevik, S., Gulbay, M., Erdogan S. A. (2007), Hierarchical fuzzy TOPSIS model for selection among logistics information technologies, *Journal of Enterprise Information Management,* Vol. 20, No. 2, pp 143 - 168

Lam, James (2003), *Enterprise Risk Management,* *From Incentives to Controls,* John Wiley & Sons, Inc., Hoboken, New Jersey, USA

Mulcahy, Rita (2003), *Risk Management - Tricks of the Trade ® for Project Managers, A Course in a Book ™*, RMC Publications, Inc., USA

Mohanty, R. P., Venkataraman, S., (1993), Use of the Analytic Hierarchy Process for Selecting Automated Manufacturing Systems, *International Journal of Operations & Production Management,* Vol. 13, No. 8., pp. 45-57

Mohanty, R. P., Agarwal, R., Choudhury, A. K., Tiwari, M. K. (2005), A fuzzy ANP-based approach to R & D project selection: a case study, *International Journal of Production Research,* Vol. 43, No. 24, pp. 5199 - 5216

Naik, S.P. (2004), *Economics,* Vipul Prakashan, Mumbai.

NASSCOM (2008), *Indian IT-BPO Analysis,* NASSCOM, www.nasscom.org

Oxford (2005), *Oxford Advanced Learner's Dictionary of Current English,* Oxford University Press, U.K.

Pressman, R. S. (2005), *A Manager's Guide to Software Engineering,* Tata McGraw-Hill Publishing Company Limited, New Delhi

Rainer R., Snyder C., Carr, H. (1991), *Risk Analysis for Information Technology,* Journal of Management Information Systems, Summer 1991, Vol. 8, No. 1, pp. 129 - 147

Saghafian, S., Hejazi, S. R. (2005), Multi-criteria Group Decision Making Using A Modified Fuzzy TOPSIS Procedure, *Proceedings of the 2005 International Conference on Computational Intelligence for Modelling, Control and Automation, and International Conference on Intelligent Agents, Web Technologies and Internet Commerce (CIMCA-IAWTIC'05) IEEE*

Sahinoglu, Mehmet (2005), Security Meter: A Practical Decision-Tree Model to Quantify Risk, *IEEE Security & Privacy, May/ June 2005,* pp. 18 - 24

SEI (2009), *Capability Maturity Model Integrated CMMI® for services version 1.2,* Software Engineering Institute. Carnegie Mellon University, www.sei.cmu.org

Strosnider, J. K., Nandi, P., Kumaran, S., Ghosh, S., Arsanjani, A. (2008), Model-Driven Synthesis of SOA Solutions, *IBM Systems Journal,* Vol. 47, No. 3, pp 415 - 431

Taylor, H (2007) Outsourced IT project from the Vendor Perspective: Different Goals, Different Risks, *Journal of Global Information Management,* April - June, 2007, Vol. 15 No. 2, pp. 1-27

Yong, Q., Long, X., Qianmu, L. (2008), Information security risk assessment method based on CORAS frame, *2008 International Conference on Computer Science and Software Engineering, IEEE Computer Society,* pp. 571 - 574

Zhang, G., Qiu, C., Li, X., Zhu, W. (2008), The Risk Assessment Model of Special Equipment Based on F-AHP and ANN, *Fourth International Conference on Natural Computation, IEEE Computer Society,* pp. 540 - 545.