

Association for Information Systems AIS Electronic Library (AISeL)

AMCIS 2010 Proceedings

Americas Conference on Information Systems
(AMCIS)

8-2010

Risks and Hidden Costs: A Study of 26 Outsourced Projects

Maliha Haddad

George Washington University, mhaddad@gwu.edu

Follow this and additional works at: <http://aisel.aisnet.org/amcis2010>

Recommended Citation

Haddad, Maliha, "Risks and Hidden Costs: A Study of 26 Outsourced Projects" (2010). *AMCIS 2010 Proceedings*. 135.
<http://aisel.aisnet.org/amcis2010/135>

This material is brought to you by the Americas Conference on Information Systems (AMCIS) at AIS Electronic Library (AISeL). It has been accepted for inclusion in AMCIS 2010 Proceedings by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

Risks and Hidden Costs: A Study of 26 Outsourced Projects

Maliha Haddad

George Washington University
mhaddad@gwu.edu

ABSTRACT

Despite the current unfavorable outlook of the larger economy, there has been a steady increase in information systems outsourcing by organizations which is projected to reach \$97.9 billion in 2012. Ordinarily, organizations outsource their software projects to avoid the risks associated with developing the software internally and to control costs. However, a study of twenty six outsourced projects indicates that such organizations face unique risks and hidden costs that are particular to software outsourcing. This paper describes research done to estimate the effort expended by organizations in overseeing and participating in outsourced software projects and the implications for identifying risks and predicting costs of such projects. For many of the organizations that participated in the survey, uncovering the actual costs and risks of an outsourced project was an eye opener: the hidden costs and risks are surprisingly significant and are typically not managed by the organization.

Keywords

IS outsourcing, risks, hidden costs, outsourced software.

INTRODUCTION

An increasing number of organizations outsource their information systems development projects to outside contractors primarily to avoid the risk of failure due to continuously evolving technologies and lack of in-house expertise. This is particularly true for government agencies, where technical staff pay scales have not kept pace with salaries offered by businesses, thus hindering the development of adequate in-house capabilities. Studies show that between one fourth and one third of software projects in the US involve outside contractors i.e. outsourcing, with an estimated value of \$70.9 billions in 2008 (Gartner, 2009). However, such an environment introduces additional costs and risks attributed to the interaction between the contractor and the customer. The primary additional costs include the contracting organization personnel involved in planning, acquiring, and managing the project as well as the costs of user involvement and participation.

Looking at existing software cost estimation processes, methods and tools in use today, most estimate the cost of the technical and some management and support resources needed for a software project. What is missing in these tools and techniques are estimates of the contracting organization resources that are expended before, during and after the contract award, in particular the cost of user involvement with the contractor during development processes. Such costs are considered *hidden* because they are incurred but not accounted for. This paper addresses the need for improvements in existing software cost estimation processes by making the hidden costs of a contracting organization visible. The results of this research are significant to outsourcing organizations as they show that the hidden costs do exist, are significant, but are not managed. Failure to plan and schedule critical resources related to the hidden costs, such as user involvement, result in introducing risks to the project since these resources might not be available when needed. Additionally, incorporating the hidden costs into economic analyses should significantly improve the decisions making process.

The paper also describes the primary risk sources related to the hidden cost that ought to be managed; internal risks arising within the customer's organization, external risks arising within the contractor's organization, and risks in the interfaces between the customer and the contractor

BACKGROUND

The following subsections provide background information relevant to the study.

Outsourcing of Information Systems Projects

Outsourcing of information systems projects is defined as the practice of hiring a contractor to develop software applications that an organization cannot or does not wish to carry out in-house. Recent findings indicate a steady increase in information systems outsourcing level: \$70.9 billion in 2008, \$76.9 billion in 2009, and \$97.9 billion in 2012 (Gartner, 2009). Organizations are outsourcing significant portions of their application development work for three main reasons: expectation of cost savings, mitigating risk of failure that might arise from continuously evolving technologies, and inadequacy of in-house capabilities. Some of the largest customers of outsourced software projects are government agencies, where technical staff salaries have not kept pace with salaries offered by businesses, resulting in an inability to develop adequate levels of internal capabilities and expertise. Other factors include the desire to adopt new internet-based technologies, web services, and alternative delivery and acquisition models, e.g. software as a service and cloud-based systems. Recent studies forecast significant growth in the application outsourcing market, specifically in the areas of custom applications, legacy application modernization, B2B integration, packaged applications, application testing services, and cloud computing based systems. In such a market, a qualified vendor may offer a cost advantage through economy of scale, ability to overcome political and cultural barriers, and providing best-practice experts.

Software Outsourcing Process Improvement Efforts

Particularly significant to the study described in this paper is the Capability Maturity Model (CMM), developed by the Software Engineering Institute (SEI) at Carnegie Mellon University (Paulk, 1993). The CMM was developed in the late 80s as a result of SEI being tasked by the Department of Defense (DoD) to develop a means to evaluate the software development capability of its bidding contractors. The CMM's success had led to the development of several models addressing systems engineering (SE-CMM), software development (SW-CMM) and an integration model (CMMI). The processes recommended by the SEI in their Capability Maturity Models for Software development (SW-CMM, CMM-I) are now widely accepted as an effective framework for organizations to achieve higher quality software. A more recent focus of the SEI has been aimed at processes involved in software acquisition i.e. outsourcing which resulted in the development of the Software Acquisition Capability Maturity Model (SA-CMM). Whereas the SW-CMM and CMM-I deal with the contractor's role in the software acquisition process, the SA-CMM pertains to the buyer's role in the process. The SA-CMM addresses the functions within the organization that support the outsourcing of software. The SA-CMM is designed to be generic enough for use by any government or industry organization in the acquisition of software. Organizations and individuals involved with outsourcing soon realized that customer involvement meant involvement in all stages of the system development life cycle (SDLC) if a fairly adequate product was to be the expected. It was also realized that the customer involvement requires not only contract oversight, but also user, tester, inspector, and other kinds of participation.

As with the CMM, the SEI sought to aid contracting organizations by defining processes that could improve their software acquisition /outsourcing by specifying the following *maturity levels* for the SA-CMM that characterize the management of such acquisitions:

Initial – The software acquisition process is characterized as being ad hoc, and occasionally even chaotic.

Repeatable – Basic software acquisition project management processes are established to plan all aspects of the acquisition, manage software requirements, track project team and contractor team performance, manage project cost and schedule baselines, evaluate the products and services, and successfully transition the software to the support organization. At this level, the project team is basically reacting to circumstances as they arise.

Defined – The organization's software acquisition process is documented and standardized. All projects must adhere to an approved, tailored version of the organization's standard software acquisition process for acquiring products and services. *Risk management is integrated into all aspects of the project* and the organization provides training needed by personnel involved in the acquisition.

Quantitative – Detailed measurements of the software acquisition processes, products, and services are collected. The software processes, products, and services are quantitatively and qualitatively understood and controlled.

Optimizing – Continuous process improvement is empowered by quantitative feedback from the process and from piloting innovative ideas and technologies.

In the survey conducted for this study, it was found that most organizations were not even aware of the SEI's SA-CMM. Few had formal, institutionalized acquisition planning and tracking processes for software. Contractor selection processes - if formalized - tended to follow traditional selection processes for products and services. Moreover, software requirements documents lacked precision and specificity. Only few projects had formalized acquisition project management, contract

management, or configuration management plans for tracking project activities and artifacts, overseeing internal and contractor personnel, evaluating progress and performance against requirements, or tracking costs. Furthermore, customer inspections of defined milestones were not part of the acquisition processes; risk identification and management processes were missing; and data collection and archiving - if done at all - was the consequence of individual efforts rather than institutionalized requirements for software acquisition

METHODOLOGY

In surveying companies about their contracted software project costs, a questionnaire was prepared and distributed. In some cases, the target-organization returned the completed survey. In others, face-to-face interviews were carried out using the questionnaire as an interview instrument. User's cost estimation processes were identified and examined, and later the users and management activities costs were measured for the surveyed projects. The questionnaire consisted of four major sections. Section 1 collected information on the participant and organization. Section 2 collected general project and contract information such as type, size, schedule, cost and characteristics. This section also collected data on possible cost drivers that could impact the cost of the contracting organization such as product, personnel, organizational and other attributes. Section 3 collected cost data on resources by each labor category involved in the project during the lifecycle phases. The funding source was determined along with the information on whether the resource was scheduled. Section 4 collected data on cost estimation processes, the significance and of certain factors on the estimation process and the impact and risks of not planning contracting organization resources on the project and the organization.

The survey was designed to examine the contracting organizations cost estimation processes, to determine the magnitude of the hidden cost and the impact of not managing the hidden costs, including the risks to the organizations. It focused on the following principal issues:

- The cost incurred by the outsourcing organization for resources needed, before and during the system development life cycle, to acquire, manage, coordinate, control and support the software project.
- The effort expended by management, users, support and other personnel with the required skills and expertise, as well as the hardware and software tools required to support such efforts.
- The cost of managing the contract, contractor, users and product quality for the duration of the project.
- The involvement of the users, user management and user functional experts who have to participate in most activities in the development life cycle, including requirement definition, product reviews, document reviews, and testing activities throughout the life cycle.
- The cost of quality assurance activities by quality assurance personnel.
- The cost of software and hardware tools, travel expenses, user training, acceptance testing, management of deliverables, management of the user and other support and miscellaneous expense items in support of the various activities.

RESULTS

The responding organizations represented a broad range of companies and government agencies. The project costs studied ranged from \$30K to \$50M. The applicable cost estimation methods varied widely, ranging from "No estimation method" to the use of "Lines of Code." The most common project estimation technique stated was "Experience." Regardless of the estimating technique used, 61% of the projects had cost over-runs and 50% had schedule over-runs. The following sections describe the findings related to the major objectives of the study.

Findings Related to the Contracting Organization Cost Estimation Processes and Practices

The findings of the survey with respect to the cost estimation processes and practices of contracting organizations may be stated as follows:

- A majority (88%) of organizations in the sample did not estimate their resources on completed projects.
- A majority (65%) did not have formal processes to estimate, plan and schedule such resources.
- Management resources, which are critical for project oversight, were more likely than other resources to be planned for by organizations. Nonetheless, such management resources were not planned for by 69% of organizations.
- User resources, whose involvement in projects is critical, were not planned for by 92% of organizations.
- The cost of such resources was not included in any economic analysis or feasibility study of the project by 57% of the organizations surveyed.
- The collection of historical data on resources involved in completed projects was not done by 62% of organizations.

In the survey, an attempt was made to identify the source of funding for the various in-house resources working on outsourced software development projects, such as management and user resources, and whether the cost of these resources were included in the final project cost. The results indicate that the funding of these resources, such as a project manager or a user, does not, in most instances, come from the project budget but rather from the department budget. Since such costs are actually incurred but not tracked or included in the final project cost, they are considered hidden costs. Consequently, the contracted price of a software development project does not reflect the actual cost to the organization. In addition there is likely to be a lack of commitment of resources whose functions may be critical for project success and therefore risks are introduced.

Findings Related to the Measurement of the Additional /Hidden Cost

The findings from this part of the study far exceeded what one might expect the value of the hidden costs to be. The study showed that the hidden costs are quite substantial, the mean value being 190% of the total development cost of the system i.e. almost twice as much as the contract cost. The magnitude of the hidden cost is significant enough to motivate organizations to institute formal procedures to account for it, and include it in project planning. The value of the hidden cost must also be taken into account in estimation models used for economic analysis of software projects and in the decision-making processes for assessing project feasibility.

Table 1 shows the magnitude of the hidden cost expended during the various phases of project development, expressed as a percentage of the overall hidden cost. The analysis and implementation phases generate the largest percentages of the hidden cost followed by the design and testing phases.

Phase	Percent of Hidden Cost
Acquisition	7.1
Analysis	18.9
Design	14.2
Programming	5.6
Testing	18.4
Implementation	18.9
Training	8.5
Other	8.4

Table 1: Phase Distribution of the Hidden Cost

Studies have shown that the analysis phase is likely to be one of the most risk prone activities of a system's SDLC, a phase in which the user is heavily involved. The role of the user is also important during the testing of the software product for system acceptance and deployment

The results of the distribution of the hidden cost by labor category in Table 2 show that the users and project management resources are the most significant.

Labor Category	Percent Hidden Cost
Project Management	34
User	48
Quality Assurance	6
Consultants	6
Others	6

Table 2: Distribution of the Hidden Cost by Labor Category

The results draw attention to the need for planning and scheduling the procuring organization's resources. The distribution of the hidden costs by labor category and by phase points to the types, amounts, and scheduling of needed resources over the

project life cycle. The results of this study support and complement results of other studies. For example, a recurring theme in risk studies, such as the one by Statz and Tenneson (1995) is that user participation, customer and user interaction, and user resource allocation, are sources of risks that need to be assessed and managed. Also, risk studies by Jones (1996, 1997) point to the importance of project management which is considered by Jones to be one of the key factors for the success of systems. The principal conclusion of this study is that the hidden costs are substantial, and that the needed resources must be planned, scheduled and adequately managed during the periods in the lifecycle where their participation is critical.

In addition, the study produced a quantitative relationship between the project size and the hidden cost expressed in man-months. For the 26 projects studied, the hidden cost was found to be linearly related to the project size (removing outliers), as expressed by the following function:

$$M = 2.2 * KLOC + 52$$

where M is hidden cost in person-months and KLOC is the project size expressed in thousands lines of code. It should be noted that different projects recorded data about project size primarily as function points or lines of code. The relationship was made uniform for KLOC using widely accepted ratios.

For many of the organizations that participated in the survey, realizing the actual cost of a project was an eye opener - hidden costs are incurred, the costs are significant, and are typically not managed. Organizations that understand inherent costs of contracting software are better positioned to estimate costs of future projects and also improve decision-making processes associated with software contract oversight.

Findings Related to Risk Identification and Management in Information System Outsourcing

In addition to identifying the hidden costs, the customer's assessment and management of risks associated with projects is equally important. One of the results of the study is the identification of risk factors and impacts by the participants, as shown in Table 3.

Risk Factor	Acknowledged (% of Participants)
1-Impact on project cost	91
2-Impact on project schedule	88
3-Impact on deliverables quality	82
4-Impact on product quality	91
5-Impact on quality of project management	91
6-Impact on contractor/customer relations	93
7- Impact on personnel performance and morale	79
8-Impact on personnel working conditions	85
9-Impact on personnel stress level	82

Table 3: Risks Factors Identified by the Study Respondents

For the studied projects the results for cost and schedule overruns are shown in Table 4.

Range of Schedule Overrun	Average Schedule Overrun	Range of Cost Overrun	Average cost Overrun
15-63%	34.95 %	5-80%	30.7%

Table 4: Range of Values and Average for Cost and Schedule Overrun

The results of cost and schedule overruns are not surprising considering the average of the hidden cost on the project at 190%, as determined by this study. The results indicate at a minimum that failure to plan for and schedule critical resources such as users, project managers, and domain experts may pose a risk to a contracting organization and to the project itself.

RISK IDENTIFICATION AND SOURCES

The study of the twenty six information systems projects led to an exploration of the risks associated with outsourcing, with the expectation that recommendations can be devised that will help minimize those risks, or at least help organizations institute mitigation plans.

What are some of the sources of risks? Risks are introduced in three primary areas: internally within the customer's organization, externally within the contractor's organization, and in the interfaces between the customer and the contractor.

The following is a list of risks in each of the three primary areas. Mitigation plans, for these risks are left unspecified since they also vary among projects and organizations:

Customer Internal Risk Sources

- Inaccurate estimations: Customers, even those who use estimation tools, can be characterized as "gutless" estimators of time, cost, and effort needed to produce software.
- Personnel: Unique knowledge, training and skills are needed to successfully launch, monitor, and bring to successful completion a contracted software project.. Few organizations have enough personnel who are trained and capable in software outsourcing processes..
- Users: The greatest risk in this category is the availability and knowledge of system users. Frequently overlooked is the project demands on user time that interferes with the user's normal work.
- User requirements: Customers frequently fail to understand, precisely articulate, or fully specify their software requirements.
- Contract specificity: Related to specification of user requirements, but broader in scope, is the specification of not only software requirements but also processes to be used between the customer and the contractor, and the details of the interfaces between the two.
- Creeping requirements: Also related to specification of user requirements and contract specificity is the problem of changing/expanding requirements. This characteristic of software projects has been exacerbated by web applications where changes to interfaces usually imply underlying changes in functionality.
- Unanticipated coordination efforts: Successful management of a contracted software project, requires a good deal of interfacing between customer and contractor during the entire life-cycle of the project. Few contracting organizations anticipate the rigor demanded in coordination and oversight.

Contractor Internal Risk Sources

Like the availability to the customer of the SEI's SA-CMM, software contractors have the SEI's CMM/CMMI recommendations to adhere to and enable their organization to position itself to successfully complete high quality software projects. We include the contractor subsection here for completeness but intentionally ignore it, concentrating instead on the interfaces between the contractor and the customer that pose risks.

Risk Sources in the Customer/Contractor Interfaces

The interfaces between the customer and the software contractor are the source of many risks that are typically not acknowledged and therefore not managed or mitigated. In an ideal world, customer and contractor would share processes and tools, their activities synchronized, and oversight welcomed. However, the large number of failed contracted projects is testimony to the problems that arise in the relationship between the customer and contractor. Like the list in the above section, the following list is intentionally not prioritized since, depending on the project and the organizations involved, the severity of the risk varies among projects and organizations. In addition, contracts where software components are sub-contracted, and must ultimately be integrated, pose additional risks and complications. The following is a list of potential risks between customer and contractor.

- Mutually- accepted ambiguous contract: In this case, both the customer and the contractor are willing to sign a contract that is not precise in specifying the work to be done
- Ill-defined interfaces: The parties do not specify the personnel interfaces for work coordination, nor do they specify the formal processes for customer-contractor interaction.
- Multiple contacts: This is related to the previous risk and involves interaction of multiple personnel on both sides. Audit trails are lost, multiple conflicting approvals may be given, and procedures, if any, are side stepped.
- Antagonistic interfaces: Over time and during stress, customer-contractor interfaces can suffer because of failed performance- perceived or real- by either side.

- Deficient inspections: Work inspections are central to the success of a contracted project. When inspections (assuming they are scheduled) are ignored, postponed, poorly structured, or ineffectual, trust suffers.
- Loosely defined checkpoints: When a mutually agreed upon project management plan is absent or deficient, agreement on acceptable checkpoints may be a source of contention.
- Testing criteria, processes and data: Frequently overlooked by the customer is the necessity, crucial need to provide application data in a format and in a timely manner that can be used in the contractor's testing processes.
- Shared repositories: The existence of shared repositories of life-cycle artifacts bodes well for a project. However, ownership and control can be a source of problems between the customer and contractor.
- Configuration management: Related to the previous, risk, configuration management controls may be the source of problems about shared documents, and even deployed systems.
- Risk management program: Negotiating agreement on a risk management plan is difficult. The contractor may resent intrusive processes, and the customer may be suspicious of opaque contractor processes.
- Quality assurance program: When the customer has more stringent QA processes than the contractor (or vice versa), quality inspections are likely to be antagonistic.
- Potential for re-use and maintainability: Customers have fewer opportunities for controlling re-use potential unless processes are in place to insure re-use. Control over design approaches for ease of maintenance also presents a risk.
- Missed schedules: With a well developed project management plan based on high task granularity, the contractor's schedules may be fully defined and oversight straightforward. However, frequently overlooked is the responsibility of the customer to provide contract backup in any number of ways. The customer's tasks need to be supported by a thorough synchronized project management plan.
- Cost of tools: System development, project planning, project management, oversight, documentation, and all of the other phases of the system development life-cycle require automated tools for building repositories of project artifacts.
- Incompatible deployment infrastructures: Insuring that the development architecture is compatible with the deployment architecture is often an overlooked aspect of a contracting arrangement, a source of added costs, and a source of contention.
- Security: Breach potential within the system architecture poses short and long term risks. Also, data used for testing or benchmarking presents potential security vulnerability.
- Transient contract and customer personnel: Employees change jobs or are moved within organizations.
- Unanticipated direct costs: Travel, on-site/off-site space requirements, audit requirements, user testing costs, meetings, use of external "overseers", materials and equipment.

BENEFITS OF MANAGING THE HIDDEN COST AND IDENTIFYING RISKS

During the data collection phase of the authors' research, many top managers of organizations who participated in the survey expressed serious interest in the study and the anticipated findings. While completing the survey, they became conscious of the magnitude of the hidden cost which they typically overlooked. Their interest can also be attributed to the following benefits that they can obtain from estimating hidden costs:

- Including the hidden cost in strategic decisions made by executive management can improve the accuracy of estimates and the decision making process before undertaking a software project. A decision is usually based on an economic analysis, which requires an estimate of the money, resources, and time required to complete the project. By including the hidden cost in cost benefit analysis, break-even analysis or make-buy decisions, managers can improve the accuracy of their estimates and confidence in their subsequent decisions.
- Using the estimates of the hidden cost in project management to plan, monitor and control the development of a project can improve the management process. Good planning and effective control require an estimate of the activities required to complete a project, and the resources required for each activity for monitoring progress.
- The cost estimates can also be used for the allocations of adequate funds for projects over time. The relationship between cost estimation and controlling projects are detailed further in DeMarco (1982).
- Using the estimates of the hidden cost in work breakdown structure to allocate resources and establish their commitments.
- Planning for hidden costs can result in improved relations and communications between the procuring organization and the contractor by allocating the needed user and management resources in a timely manner. For members of a project team to work together more efficiently on a project, each member should understand his/her role in the various activities of the project.

- Including the critical resources in the risk prone activities can result in improved risk management.
- Improving the chances for successful software projects: Estimating and providing the necessary resources for a software project may help in avoiding the negative impacts identified in the findings and may improve the chances for developing better quality systems, systems within budget, on schedule, that meet the user requirements.
- Improve the chances for successful software projects: Estimating and providing the necessary resources for a software project may help in avoiding the negative impacts identified in the findings and may improve the chances for developing better quality systems; systems within budget, on schedule, that meet the user requirements.

CONCLUSIONS

A formal, institutionalized, software outsourcing project management process model will require that the organization plan all aspects of the outsourced project, manage software requirements, track project team and contractor team performance, manage project cost and schedule baselines, evaluate the products and services, and successfully transition the software to its support organization. Software project risk management can and should be integrated with an organization's business processes. Especially critical to the success of a contracted project is the understanding of how to identify and manage risks. The cost and risk impacts of not improving software outsourcing processes are substantial.

REFERENCES

1. Aubert, B. A., Patry, M., and Rivard, S. (2005) A Framework for Information Technology Outsourcing Risk Management, *ACM SIGMIS Database* v 36, n.4, October, p.9-28.
2. Bakker, Karel Boonstra, Albert, Wortmann, Hans (2009) Does risk management contribute to IT project success? A meta-analysis of empirical evidence. *International Journal of Project Management*, In Press, Corrected Proof, Available online 18 August.
3. Boehm, Barry, Abts, Chris, and Chulani, Sunita (2000) Software development cost estimation approaches — A survey, *Annals of Software Engineering*, Volume 10, Numbers 1-4, Springer Netherlands.
4. CMMI Product Team (2002). Capability Maturity Model Integration (CMMI) Version 1.1, Carnegie-Mellon Software Engineering Institute.
5. Cooper, J., Fisher, M., and Sherer, S. W., Editors, (1999). Software Acquisition Capability Maturity Model (SA-CMM) Ver.1.02, *Technical Report* CMU/SEI-99-TR-002 ESC-TR-99-002
6. DeMarco, Tom. (1982). Controlling Software Projects: Management, Measurement, and Estimation, Yourdon Press, New York, NY.
7. Dibbern, J., Goles T., Hirschheim, R, and Jayatilaka, B (2004) Information Systems Outsourcing: A Survey and Analysis of the Literature. *ACM SIGMIS Database* 35, 4, Nov. 6-102.
8. Gartner Reports (2009) User Survey Analysis: IT Outsourcing Opportunities Arise from Crisis, North America.
9. Hihn, H. and Agahi, Habib. (1991) Cost Estimation of Software Intensive Projects: A Survey of Current Practices. *Proceedings of the 13th International Conference on Software Engineering*, Austin, TX, 13-17 May, pp. 276-287.
10. Jones, Capers (1994) Assessment and Control of Software Risks. Yourdon Press, Prentice-Hall, Inc. Englewood Cliffs, NJ.
11. Jones, Capers (1996). Patterns of Software Systems Failure and Success, International Thomson Computer Press.
12. Kitchenham, B., Pfleeger, S. and Fenton, N. (1995). Towards a Framework for Software Measurement Validation. *IEEE Transactions on Software Engineering*, Vol. 21, No. 12, December 1995, pp. 929-944.
13. Nielsen, J and Miller, A. (1996). Selecting Software Subcontractors, *IEEE Software*, Vol. 13, No. 4, July, pp. 104-109.
14. Paulk, M.C., Curtis, M. B. Chrissis, and Weber, C.V. (1993) The Capability Maturity Model for Software, Version 1.1, Software Engineering Institute, CMU/SEI-93-TR-24.
15. Statz, Joyce, and Tennison, Susan (1995) Getting Started with Software Risk Management. *American Programmer*, Vol.8, No.3, pp. 23-30 March.
16. Yang, B., Fu, H., and Zuo, M. (2005), The Integration Mechanism of IT Outsourcing Partnership. *Proceedings of the 7th International Conference on Electronic Commerce* (Xi'an, China, ICEC 05, vol. 113. ACM, New York, NY, August, 801-803.