AMCIS 2007 Proceedings

Americas Conference on Information Systems (AMCIS)

December 2007

Risk Perspectives in Systems Development

Merrill Warkentin
Mississippi State University

Ernst Bekkering
Northeastern State University

Allen Johnston
University of Alabama Birmingham

Robert Moore
Mississippi State University

RISK PERSPECTIVES IN SYSTEMS DEVELOPMENT

Merrill Warkentin, Mississippi State University, mwarkentin@acm.org

Ernst Bekkering, Northeastern State University, bekkerin@nsuok.edu

Allen C. Johnston, Univ. of Alabama Birmingham, allencjohnston@gmail.com

Robert S. Moore, Mississippi State University, rmoore@cobilan.msstate.edu

Abstract

Information systems development projects are a significant expenditure for enterprises, and numerous projects fail to achieve their objectives. Systems development risk factors are presented and categorized into technical, resource constraint, organizational, and "other" risks, based on the prior literature. These factors are analyzed and synthesized, and an integrative framework is presented. Experienced systems development engineers and project managers participated in a structured "interview" through an in-depth, open-ended questionnaire. The expert panel's responses provide rich, qualitative data regarding their perspectives on the relative importance of the various risk factors and procedures used to ameliorate them. Common threads and key distinctions characterizing their experiences are identified. Our analysis indicates that risk factors ultimately derive from organizational influences and may be overcome with organizational responses. We also noticed a dichotomy between perspectives of senior managers and those of lower-level systems developers. Systems developers are more likely to characterize risks as fitting into one of the discrete categories described above, while senior project managers are more likely to view all risks as being, at their core, organizational risks. Finally, objectives for future research in this area are presented.

Keywords: *Systems development; project risk; risk factors, success; failure; organization; software, perspectives*

Introduction

Thousands of enterprises undertake Systems Development (SD) projects every year. Some are large endeavors requiring extensive resources (time, personnel, and money), with complex inter-organizational development processes and sophisticated technical requirements, while others are relatively small and simple. Some have a global scope and some may affect only a small department or user group. Some systems are developed "in-house", and other projects may be outsourced. However, there is strong evidence to suggest an unacceptably high rate of system failure, regardless of the size or complexity of the project. Too many projects are abandoned before completion or result in systems that do not meet requirements. A global survey of IT executives found that 23% of projects were cancelled before completion or never implemented (Standish Group International 1995; see also Standish Group International 1999; Standish Group International 2001). Ewusi-Mensah (2003) estimates that one-third of SD projects fail or are abandoned.

Many researchers have sought to identify the key risks associated with SD projects (Barki et al. 1993; Keil et al. 2000; Lyytinen et al. 1998; Schmidt et al. 2001; Wallace 1999). In light of those findings, managers have sought to assess such risks in the systems planning process (Alter 1979; Alter et al. 1978; Segars et al. 1996), systems evaluation process (McFarlan 1981), and managerial assessments (Barki et al. 2001), usually considering each risk to be independent of the others. We propose that risks may independently influence the success of SD projects, interact with other risk factors, or even have their origins in another category. Specifically, many of the technical, resource constraint, and other risks may in fact be seen as organizational risks at their core.

To explore this proposition, we first develop a coherent list of the risks identified in the prior literature, which have been categorized as technical, resource constraint, organizational, and "other" risks. Next, we present the results of our analysis of an expert panel's encounters with these risks. We discuss the findings and conclude with a discussion of future research opportunities.

Systems Development Risk Factors

Technical Risk Issues

The first primary category of SD risk comprises "technical" risks. Alter (1979) identified technology as one of eight SD risk factors, and McFarlan (1981) identified a lack of experience with technology as a significant threat to success. Since that time, technology risks have been included in nearly every SD risk assessment framework. More recently, a PriceWaterhouseCoopers survey of 120 SD project failures revealed that the majority involved "systems failure," where the system did not perform acceptably or did not work at all (Webster 2000). Another survey indicated that 67% of civil legal claims in connection with SD projects were brought for failure to meet the claims of the developer, and 45% of claims were brought for defects that rendered the system unusable (Bednarz 2002).

Technical risks may be encountered in nearly all facets of an SD project. They have been associated with the introduction of new, or "bleeding edge," technology (Schmidt et al. 2001), access control mechanisms (McGraw 2002), the acquisition of new hardware and software (Barki et al. 1993), and the involvement of multiple vendors in an SD project (Barki et al. 1993). Lyytinen et al. (1998) suggest that certain technical risks, such as compatibility, manifest early in the SD lifecycle, while other technical risks such as extendibility, maintainability, and reliability are associated with later phases of the lifecycle.

Resource Constraint Risk Issues

Another category of SD risk involves the limitation of available organizational resources. Haber (2003) asserts that companies must always focus on ROI and "the triangle, and balance, of cost, time, and quality." Laudise & Nuara (2002, p. 299) suggest that "three critical resources – people, time, and money" must be planned in advance and monitored regularly throughout the SD process. Ewusi-Mensah (2003) cites cost overruns as a significant cause of project failure. Difficulties within the development process pose direct burdens on these resources, and left unaddressed, may compound over time.

Cost-benefit analysis has always been a cornerstone of systems feasibility studies, but the accurate assessment of costs and benefits presents many challenges. For example, many of the benefits may be intangible and, therefore, difficult to quantify. Inaccurate cost-benefit analyses can even result in systems with a negative ROI.

Organizational Risk Issues

Organizational factors were first recognized as significant risks in the mid 1970's (Lucas 1975), and subsequent research suggests that they eclipse technical risks (Clegg et al. 1997; Doherty et al. 1998; Doherty et al. 2001a; Doherty et al. 2003b; Ewusi-Mensah et al. 1994; Hornby et al. 1992; Standish Group International 2001). A survey of senior IS executives found that half of the factors leading to canceled SD projects were organizationally-oriented, ranging from ambivalent senior management support to a lack of end-user involvement (Ewusi-Mensah et al. 1994). A recent global survey of IT executives (Standish Group International 2001) found that the reason for most project failures "…was not for lack of money or technology; most failed for lack of skilled project management and executive support" (p. 1). Successful projects, on the other hand, were linked to executive support, a high level of user involvement, the assignment of an experienced project manager, and clear business objectives (Standish Group International 2001). Project managers provide coordination between all stakeholders, and analysts elicit the vision, scope, and business requirements of the project. Clemons et al (1995) extend this to the future in distinguishing functionality risk due to inability to foresee future needs, and the political risk caused by sweeping changes to meet those future needs.

Trust and communication issues have also been cited as significant organizational risks, particularly in outsourcing (Morgan et al. 1994). For some firms, "80-90% of the important issues today are organizational" (Clegg et al. 1997, p. 856). Despite the magnitude of organizational risks, many are not explicitly acknowledged or managed (Doherty et al. 1998). When they are addressed, they are often treated by the wrong people at the wrong time (Doherty et al. 2003a). In some instances, improper treatment of organizational risks leads to a culture of failure with a commitment to ineffective methods and unsuccessful knowledge management practices (Lyytinen et al. 1999). However, King and Doherty (2001b) argue that attention to organizational issues in the SD process is positively correlated with the maturity of an organization's use of the technology.

Systems development impacts organizational culture, structure, business processes, and human-centered issues such as task redesign (Doherty et al. 2003b), creating a wide range of organizational risks. While the majority of systems developers "treat most organizational issues at some point in the systems development process" (p. 57), those who do not are much more likely to encounter project failure. Further, timing is important. While treating an organizational issue in different phases (feasibility, analysis and design, or implementation) appears equally effective, there is evidence to suggest that repetitive treatment is valuable (Doherty et al. 2003b). Or, "those organizations that treat an organizational issue in more than one phase of a systems development project are likely to have higher levels of systems' success than those that do not" (p. 52).

Another study (Al-Mushayt et al. 2001) sought to determine the extent to which the 14 organizational issues adopted from Doherty and King (1998) are addressed, and whether best practices are helpful. Findings indicate a strong tendency to address organizational issues, with organizational contribution issues most frequently treated. Organizations utilizing best practices could treat a wider range of organizational issues than those that did not. Finally, findings indicate a lack of relationship between SD methodology and addressing organizational issues. Structured methods such as SSADM do not assist in treatment of organizational issues; but neither do socio-technical or prototyping methods.

Other Risk Factors

Additional risks arise from environmental, cultural, user acceptance, ethical, and legal factors. Legal risks, for example, become particularly acute in outsourcing. A mismatch of expectations and poorly drafted contracts often lead to disputes, partly because "[t]he vendor goes by defined contractual obligations, whereas the client looks to solve business issues…" (Bednarz 2002, p. 2). Legal disputes also arise in connection with warranty issues, breaches of confidentiality, and the re-use of code by outsourcers in systems built for competitors. While such legal and other risks are significant, further discussion is beyond the scope of this paper.

Methodology

To investigate interaction of risks, an expert panel was convened to examine relationships between risk factors. The process of developing our expert panel questionnaire and member selection is discussed next.

Questionnaire Development

Our synthesis of the prior research resulted in a comprehensive list of technical, resource constraints, organizational, and "other" risk factors distilled in Table 1 below. This table captures the essence of the majority of SD risk research, integrating the pre-eminent research of Barki & Talbot (1993), Schmidt et al. (2001), and Wallace (1999) with organizational and emerging risk research.SD risk factors can be considered as different types of risk, depending on the context. For instance, the item "Project Leadership Problems" can result from insufficient funds to hire a better project manager (resource constraints), internal politics and power struggles (organizational), and even contractual obligations.

The list in Table 1 served as our starting point in the development of an open-ended questionnaire (omitted here for space considerations). The pre-tested questionnaire was designed to focus panelists' attention on the risk factors under investigation and to evoke detailed experiential knowledge covering all categories of SD risk. The document also asked for descriptions of ameliorative actions contributing to success and failure. Each panelist was given the choice of conducting a telephone interview, but all preferred to provide written responses. We asked to describe specific instances illustrating various risk factors and how they were addressed. During pre-testing, we noted that completing the questionnaire would require a large amount of time. Consequently, we requested each panelist to focus on the most relevant issues in their experience.

Table 1: Synthesis of SD Risk Factors from Previous Research Literature

	Technical	Resource constraint	Organizational	Other
Inability to acquire necessary hardware	x	x		
Inability to acquire necessary software	x	x		
Inadequate hardware vendor support	x	x		
Inadequate software vendor support	x	x		x
Technical complexity of project	x	x		
Technical incompatibility with existing systems	x	x		
Technical incompatibility between new system components	x	x		
System requires connectivity between multiple firms	x	x	x	
Large size of project (large number of departments or users)	x	x	x	
Large size of project team (large number of developers)	x	x	x	
Insufficient or inappropriate staffing		x	x	
Team's lack of skills or expertise		x		
Team member communication or compatibility problems		x	x	
Instability of team composition (changing team members)		x	x	
Project leadership problems (inexperience, lack of "people skills," etc.)		x	x	x
Lack of effective development process or methodology	x		x	
No planning or inadequate planning		x	x	
Unclear or misunderstood scope or objectives			x	
Changing scope or objectives during project			x	
Inaccurate or vague user requirements			x	
Organizational transition difficulties			x	
Lack of user involvement during development		x	x	
User expectations don't match project objectives			x	x
Conflict between user departments			x	
Budgetary or financial constraints		x	x	
Lack of top management commitment to the project		x	x	x
Organizational politics			x	x

Expert Panel

A diverse panel of eight experts with extensive national and international experience in project management was recruited from both large and small companies. Each received a standard research consent document and could choose how to be identified. Each was asked to provide details based on their experience. Most participants were very forthcoming in providing answers to nearly all questions. Table 2 provides an overview of the expert panel, using each panelist's choice of self-description. Pseudonyms were assigned to mask their identities.

Table 2: Panel of Systems Development (SD) Experts

Panel Member	Title	Type of Organization	Yrs*	Primary Role
Greg	Enterprise Architect	State Government	15	Project Manager
Harry	Senior Engineering Mgr	Global Medical Technology	8	Technical manager
Steve	Senior Programmer	International Transportation	4	Lead Developer
David	Software Engineer	Distribution/Retail	22	Tech Team Leader
Ellen	Programmer	Large U.S.-based Company	17	Team Leader
Roger	Operations Manager	Boston-based Financial Firm	4	Project Manager
Bill	Company Owner	Independent Database Developer	19	Project Manager
Mac	CIO and CTO	Global Technology Firms	25+	Project Manager

* Number of years experience

The panel comprised a diverse set of developers and SD project managers with experience ranging from four to 25+ years. All reported having been involved in numerous large projects in capacities ranging from senior project managers to junior systems developers. Some projects were highly structured, as in the case of a turnkey system for tracking maintenance and repair of medical systems, while others were less structured, as in the case of a system controlling all live audio and video of a large broadcasting company requiring active standby systems in case of failure. Some were new projects, such as the development of a state eGovernment system, while others involved adaptation of existing systems.

Results

Responses from the panel resulted in over 7500 words. Some members spent more time on certain sections, whereas others focused on other portions, spending from thirty to over ninety minutes completing their answers. Iterative analysis followed by discussion and comparison between the authors revealed several themes, or consistent patterns of responses. Despite the variety of projects, industries, and roles, the panelists concurred on the identification of success factors, such as high client participation, strong executive management support, good technical skills of team members, thorough requirements definition at the start of the project, and good project management during the project. Additionally, failure factors included lack of requirements definition, poor planning, and insufficient project management. In the experiences of our panelists, most projects were not abandoned, but significant effort was spent to solve problems. Steve, a senior programmer with an international transportation company, explains,

> *"Fortunately, I haven't been on any projects that were just a bust. I have been on a number of projects that required a lot of last minute scrambling. Primarily this has been due to lapses in the analysis and design phases."* He adds*, "…we've learned to dedicate more of the schedule to analysis and design. Taking the time on the front-end has helped*

> *make coding more efficient and improved our testing cycles. We have a lot less "gotchas" during the testing cycle."*

When problems do occur, good communication is essential. Roger, the operations manager of a Boston-based financial firm, explains,

> *"Overall our projects have been very successful, with the exception of timing. The end results have always been good, but too often the time it takes to complete a project is longer than anticipated and longer than desired. ... The only way to resolve a time issue is to manage your relationships and understand your leverage. Once you understand your leverage, you take control of the situation. Ultimately you need effective communication channels with your vendors and technology partners. Mutual respect and understanding play a large role in the relationship, but at the end of the day, whoever is footing the bill needs to get a return on their investment."*

Problems cannot always be prevented. Bill, an independent database developer, relates this experience –

> *"A major national insurance company needed a sales reporting application. After full specification ... and substantial progress in development, the client changed the contact person to someone totally ignorant of the types of information needed for the developer to complete the project. The project was terminated half way to completion."*

Sometimes the project itself proves to be too ambitious and complex. David, a software engineer in the distribution/retail industry, explained that an application solution intended to "meet cross sector (e.g., grocery, mass merchandise, specialty), cross geography (North America, South America, Europe, Asia), and even cross industry (retail/distribution, banking, insurance)" failed despite solid executive support and ample time and resources. He continued, "we simply couldn't meet the expectations."

Past problems have led others not to accept projects deemed too complex. Harry, the senior engineering manager at a global medical technology company, stated

> *"I declined a project that I viewed as too complex. At the beginning of all projects, I try to map out the ending I want to have and then work backwards from there to figure out how to achieve it. If the project is so complex that I am not clear on the approach, then I won't take it on."*

Others try a phased approach. Mac, the former CIO and CTO of several global technology firms with over 25 years of experience adds,

> *"There are projects that are too complex in total, and have to be broken down into doable chunks."* But when problems can be foreseen and the risks are accepted, prioritization is still important, according to Harry: *"Identify and focus on the main risks early and find the necessary talent to mitigate the risks. I usually try and find multiple mitigation strategies for high risk areas and I sometimes go after these strategies in parallel until I am comfortable in a final solution."*

Time and schedule delays can also significantly affect projects "mainly when we have a set launch date," revealed Steve. He adds,

> *"Delays directly impact all of the phases of the development cycle, so something is going to get missed. Delays tend to lead to a 'launch and fix' approach. These decision delays also lend to 'scope creep' because requirements are still being finalized throughout the coding phase." Systems developers try to cope with delays by emphasizing the importance of well-understood requirements, increasing staffing and using overtime, and even continuing to work on the project without official approval.*

Similarly, systems developers often try to be as flexible as possible with respect to client requests for changes. David accommodates changes, albeit at a price:

> *"Though you should work hard to avoid it, you should also plan on dealing with some degree of change. We are most successful when we 'train' the customer early on to understand the impact (cost, time, etc.) of requested changes. We rarely reject a*

> *customer request for change since we are a custom services organization. It is even more rare that we accept those changes without compensation."*

Ellen, an experienced programmer at a large U.S.-based company, does not even charge extra –

> *"It is part of the job. You are paid for a job, not hours."* This seems to be a general trend. Ellen acknowledges that cost overruns occur, *"but more and more companies are trying to accommodate the customer. It depends on the importance of the project and the customer. An important customer has a company jumping through hoops!"*

A minority of systems developers are less flexible, exemplified by Harry, who has

> *"refused to incorporate changes if I felt they were unreasonable or not needed. Unless the change is absolutely necessary or easy to incorporate, I almost always reject it in order to keep the project moving along. Changing the scope of the project midstream is very damaging to the team and is something I strongly discourage."*

Good communication is repeatedly mentioned in relation to managing client expectations. Harry notes,

> *"Explicit instruction along with continued communication is very important to having a successful project ... [but it is not perfect]. Providing explicit written expectations is a good start but this still leaves some room for interpretation."*

Communication is especially important if complexity of the project increases due to the participation of multiple parties. Greg, an enterprise architect for a state Department of Information Technology Services, remembers a project with

> *"numerous sub-contractor development firms participating, as well as numerous client agencies participating as customers. My role was to communicate among the client agencies, which turned out to be a very demanding task. Each agency had very unique ideas on how the project should proceed, thus creating a very challenging scenario. Expectations had to be adjusted as the project progressed. This was accomplished through face-to-face negotiation with the customer agencies, as well as the development vendors." Mac encountered this in a project where "everyone had a different agenda," and this "bogged down progress to the point of the project being stopped."*

Legal contracts are not generally considered to be an effective method of ensuring project success.

> *"I have been involved in and observed several 'legal' wins and losses, but all were associated with failed projects," offered David. Sometimes they are viewed with outright scorn. "Contracts are very poor tools that only work to rationalize failure*," added Mac. Or as David put it*, "Ask if in the long run there are any winners after suing or being sued by a client. Congratulations, you win, but your future business is dead."*

The technology itself is seldom seen as a problem, except when a new technology is still unstable. Hardware is never seen as a problem, and software is problematic mainly when multiple vendors are involved. The human element appears to be a primary concern. Mac recommends that if you "focus on behavior and people issues, the technology takes care of itself…" This focus on human factors over technical skills is also evident in the hiring process, where personal attributes like "a cooperative attitude" (according to David) and "problem solving abilities" (according to Harry) are mentioned as frequently as technical skills.

Though all respondents agreed on the general importance of organizational risks over technical risks, the focus of their responses regarding major causes and strategies for risk reduction differed significantly between lower level systems developers and senior project managers. Consider these responses to the question about greatest risks. Lower level project members focused on discrete risks:

> *"Not having the required skills or talent on the team" (Harry)*
>
> *"A lack of thorough analysis and design is probably the biggest risk with projects I've been involved with" (Steve)*
>
> *"One major issue: poorly defined requirements" (David)*

> *"Development cycles are shortened in order to make dates. Additional line items are necessary. Adequate machines to test on. Several times a product runs on several platforms. It is necessary to be able to verify the code works on each of these. This can be very time consuming as well as difficult to acquire the equipment" (Ellen).*

These responses are in strong contrast with the organizational focus of senior managers:

> *"Poor planning and requirements gathering on the front end and poor communications skills during the project" (Greg)*
>
> *"Accuracy. We require 100% accuracy. (...) Once all issues have been resolved (if any) we would role the product out in two waves. We do this in two waves to safeguard against any potential unforeseen issues that might arise down the line. We typically wait three weeks between the two waves of deployment "(Roger)*
>
> *"Management commitment and people motivation" (Mac).*

Similarly, risk minimisation strategies differ between junior and senior team members. Junior members focus on specific measures:

> *"Identify and focus on the main risks early and find the necessary talent to mitigate the risks. I usually try and find multiple mitigation strategies for high risk areas and I sometimes go after these strategies in parallel until I am comfortable in a final solution" (Harry)*
>
> *"Requiring finalized requirements to minimize scope creep" (Steve)*
>
> *"Invest significant time and effort working with the client to define requirements before proceeding to the implementation phase" (David)*
>
> *"Clearly understand what needs to be done prior to providing sizings" (Ellen).*

Senior team members plan to minimize risk in more general organizational terms:

> *"Planning, up-front." (Greg)*
>
> *"We don't like to be the first player on the street to sign on to any new product. We typically wait for some of our partners of industry contacts to run with the product, before we get on board: (Roger)*
>
> *"Focus on behavior and people issues; the technology takes care of itself, assuming you used an iterative development process" (Mac).*

In conclusion, even though junior and senior SD project members recognize and acknowledge the predominance of organizational risks, their focus of attention and risk reduction strategies demonstrate the influence of their position and responsibilities.

Discussion and Implications for Future Research

Our synthesis of the prior literature revealed two themes which were reinforced by the experiences and opinions of our panel of experts. The first theme is that organizational risks appear to overshadow all other risks, and that all risks might be ultimately construed as organizational risks. While certain risks can fit into the technical, resource constraints, and "other" categories, all such risks could be, at their core, organizational risks that might be overcome with organizational responses. There appears to be an interaction where risk factors cannot be appropriately managed without maintaining an organizational perspective. For example, insufficiently specific system requirements can be overcome by increasing collaboration between stakeholders.

Another emerging theme is the difference between senior managers who view risks as organizational in nature, and junior members who view risks mainly as technical or resource issues. Doherty and King (1998; 2001a) implicitly recognized this when they sampled only senior IT professionals with high levels of managerial responsibility to comment on the importance of managerial issues. This difference is illustrated in Figure 1.

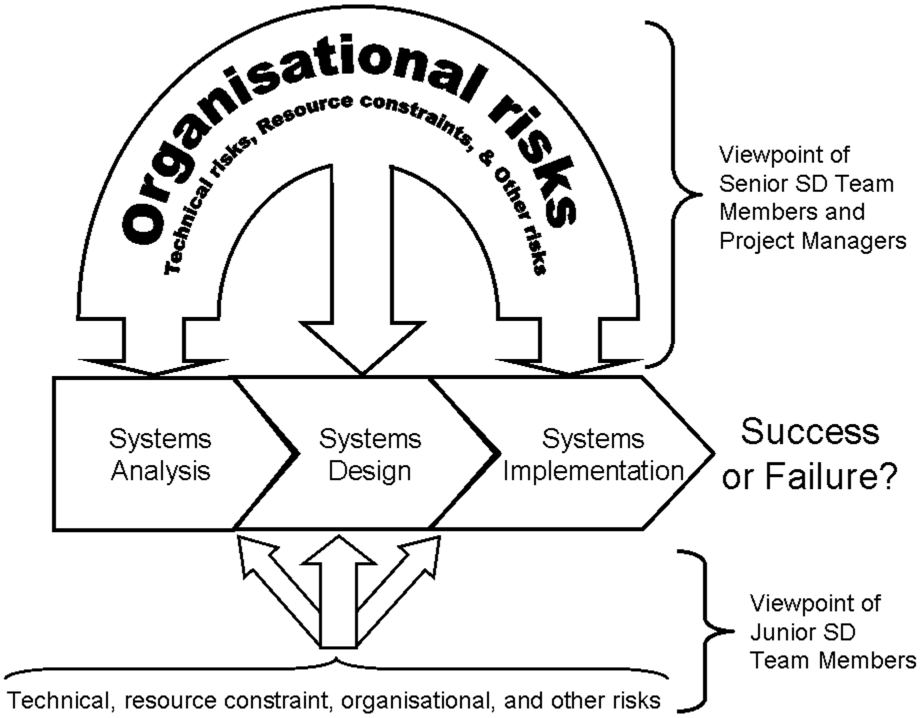
Organisational risks
Technical risks, Resource constraints, & Other risks
Viewpoint of Senior SD Team Members and Project Managers
Systems Analysis
Systems Design
Systems Implementation
Success or Failure?
Viewpoint of Junior SD Team Members
Technical, resource constraint, organisational, and other risks

Figure 1: Comparative Perspectives on System Design Risk

Our findings support theories advanced by other studies, that the organizational aspects of SD risk are the greatest source of failure of SD projects, and that organizational risks effectively subsume other categories of risk. For example, our experts confirm that most technical challenges can be overcome with proper managerial processes and sufficient commitment of resources. Similarly, strong organizational support for a project, coupled with compromises and creative solutions, can usually resolve resource constraint issues. While programmers and less-experienced personnel working in the trenches may view risks as unrelated to organizational issues, senior managers view all risks as having organizational roots and organizational solutions. One of the most experienced experts on our panel succinctly summarized this as “Focus on behavior and people issues; the technology takes care of itself.”

Without organizational commitment and management of human-centered issues, transitional issues, and organizational alignment, even systems without other challenges are likely to fail. A key mediator, both in terms of overall success and risks when they arise, is effective management of relationships between development team and clients, be they intra-organizational (in-house development) or inter-organizational (outsourcing arrangements). Effective communication, strong organizational commitment to the project, and attention to personnel involved are crucial.

The present study illuminates the existing theoretical foundations regarding SD risk by exploring responses from a convenience sample of experts. Future studies could increase the sample size to provide quantitative assessments. Further research can discover ways to integrate perspectives of less experienced, lower-level personnel and senior, upper-level managers to create a cohesive approach. Finally, prevention and remediation are important topics for further study. Acknowledging and identifying SD risks is a valuable first step, but the ultimate goal is to identify successful approaches to reduce risks, thereby improving the success rate of SD projects.

References

Al-Mushayt, O., Doherty, N., and King, M. "An investigation into the relative success of alternative approaches to the treatment of organizational issues in systems development projects," *Organization Development Journal* (19:1), Spring 2001, pp 31-48.

Alter, S. "Implementation risk analysis," *TIMS Studies in Management Sciences* (13:2) 1979, pp 103-119.

Alter, S., and Ginzberg, M. "Managing uncertainty in MIS implementation," *Sloan Management Review* (20:1) 1978, pp 23-31.

Barki, H., Rivard, S., and Talbot, J. "Toward an Assessment of Software Development Risk," *Journal of Management Information Systems* (10:2) 1993, pp 203-225.

Barki, H., Rivard, S., and Talbot, J. "An integrative contingency model of software project risk management," *Journal of Management Information Systems* (17:4), Spring 2001, pp 37-69.

Bednarz, A. "IT malpractice," Network World Fusion, 2002.

Clegg, C., Axtell, C., Damodaran, L., Farbey, B., Hull, R., Lloyd-Jones, R., Nicholls, J., Sell, R., and Tomlinson, C. "Information Technology: A study of performance and the role of human and organizational factors," *Ergonomics* (40:9) 1997, pp 851-871.

Clemons, E., Thatcher, M. and Row, M. "Identifying sources of reengineering failures: A study of behavioral factors contributing to reengineering risks," Journal of Management Information Systems (12:2) 1995, pp. 9 – 36.

Doherty, N.F., and King, M. "The consideration of organisational issues during the systems development process: an empirical analysis," *Behaviour & Information Technology* (17:1) 1998, pp 41-51.

Doherty, N.F., and King, M. "An investigation of the factors affecting the successful treatment of organisational issues in systems development projects," *European Journal of Information Systems* (10:4), December 2001a, pp 147-160.

Doherty, N.F., and King, M. "The Treatment of organisational issues in systems development projects: The implications for the evaluation of Information Technology investments," *The Electronic Journal of Information Systems Evaluation* (4:1), March 2001b.

Doherty, N.F., and King, M. "From technical change to socio-technical change: towards a proactive approach to the treatment of organisational issues," in: *Socio-technical and human cognition elements of information systems,* E.C. Steve Clarke , Gordon Hunter , Andrew Wenn (ed.), Idea Group Publishing, Hershey, PA, 2003a, pp. 22 - 40.

Doherty, N.F., King, M., and Al-Mushayt, O. "The impact of inadequacies in the treatment of organizational issues on information systems development projects," *Information & Management* (41:1), October 2003b, pp 49-62.

Ewusi-Mensah, K. *Software development failures* The MIT Press, Cambridge, MA., 2003.

Ewusi-Mensah, K., and Przasnyski, Z.H. "Factors contributing to the abandonment of information systems development projects," *Journal of Information Technology* (9) 1994, pp 185-201.

Haber, L. "How do you measure ITSM success?," Jupitermedia Corporation, 2003.

Hornby, C., Clegg, C., Robson, J., McClaren, C., Richardson, S., and O'Brien, P. "Human & organisational issues in information systems development," *Behaviour & Information Technology* (11:3) 1992, pp 160-174.

Keil, M., Tan, B.C.Y., Wei, K., Saarinen, T., Tuunainen, V., and Wassenaar, A. "A cross-cultural study on escalation of commitment behavior in software projects," *MIS Quarterly* (24:2) 2000, pp 299-325.

Laudise, T.M., Esq. , and Nuara, L.T., Esq. "How to contract for a successful E-commerce development project: Beating the odds," in: *The Business Lawyer*, 2002.

Lucas, H.C. *Why information systems fail* Columbia University Press, New York, NY, 1975.

Lyytinen, K., Mathiassen, L., and Ropponen, J. "Attention Shaping and Software Risk - A Categorical Analysis of Four Classical Risk Management Approaches," *Information Systems Research* (9:3), September 1998, pp 233-255.

Lyytinen, K., and Robey, D. "Learning failure in information systems development," *Information Systems Journal* (9:2), April 1999, pp 85-101.

McFarlan, F.W. "Portfolio approach to information systems," *Harvard Business Review* (59:5), Sep/Oct 1981, pp 142-150.

McGraw, G. "Managing software security risks," *IEEE Computer* (35:4) 2002, pp 99-101.

Morgan, R.M., and Hunt, S.D. "The commitment-trust theory of relationship marketing," *Journal of Marketing* (58:3) 1994, pp 20-38.

Ropponen, J., and Lyytinen, K. "Components of software development risk: How to address them? A project manager study," *IEEE Transactions on Software Engineering* (26:2) 2000, pp 98-112.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. "Identifying Software Project Risks: An International Delphi Study," *Journal of Management Information Systems* (17:4) 2001, pp 37-69.

Segars, A.H., and Grover, V. "Designing company-wide information systems: Risk factors and coping strategies," *Long Range Planning* (29:3) 1996, pp 381-392.

Standish Group International "The CHAOS Report (1994)," 1995.

Standish Group International "CHAOS: A Recipe for Success (1999)," 1999.

Standish Group International "Extreme CHAOS (2001)," 2001.

Wallace, L.G. "The Development of an Instrument to Measure Software Project Risk," in: *College of Business Administration*, Georgia State University, 1999.

Webster, B.F. "Patterns in IT litigation: systems failure (1976-2000)," Pricewaterhouse Coopers LLP.