Routledge
Taylor & Francis Group

Journal of Management Information Systems

ISSN: 0742-1222 (Print) 1557-928X (Online) Journal homepage: https://www.tandfonline.com/loi/mmis20

Identifying Software Project Risks: An International Delphi Study

Roy Schmidt , Kalle Lyytinen , Mark Keil & Paul Cule

To cite this article: Roy Schmidt , Kalle Lyytinen , Mark Keil & Paul Cule (2001) Identifying Software Project Risks: An International Delphi Study, Journal of Management Information Systems, 17:4, 5-36, DOI: 10.1080/07421222.2001.11045662

To link to this article: https://doi.org/10.1080/07421222.2001.11045662

Identifying Software Project Risks: An International Delphi Study

ROY SCHMIDT, KALLE LYYTINEN,
MARK KEIL, AND PAUL CULE

ROY SCHMIDT is Assistant Professor of Business Computer Systems in the Foster College of Business Administration, Bradley University. He also served as a founding faculty member of the Hong Kong University of Science and Technology. His research interests include support of the strategic decision process, executive information use, information systems project management, and information systems project risk management. He has presented his research in *Communications of the ACM, Decision Sciences Journal, Information Systems Management, International Conference on Information Systems,* and at major international and national conferences. He also serves as a member of the editorial board of the *Journal of Information Technology.* Before entering academia, Dr. Schmidt managed several large-scale software development projects during a 22-year career in the U.S. Air Force.

KALLE LYYTINEN is a full professor in Information Systems at the University of Jyväskylä, Finland. He currently serves on the editorial boards of *Information Systems Journal, European Journal of Information Systems, Accounting, Management and Information Technology, Information Systems Research, Information Technology & People, Requirements Engineering Journal,* and *Journal of Strategic Information Systems.* Since 1997 he has been a Senior Editor of *MIS Quarterly.* He has published over 70 articles and edited or written six books. His research interests include information system theories, system design methods and tools, system failures and risk assessment, electronic commerce, computer-supported cooperative work, decision-making theories, and diffusion of complex standardized technologies.

MARK KEIL is Associate Professor in the Department of Computer Information Systems at Georgia State University. His research focuses on software project management, with particular emphasis on understanding and preventing software project escalation. His research is also aimed at providing better tools for assessing software project risk and removing barriers to software use. Keil's research has been published in *MIS Quarterly, Journal of Management Information Systems, Sloan Management Review, Communications of the ACM, IEEE Transactions on Engineering Management, Decision Support Systems,* and other journals. He currently serves as coeditor of *Database* and as an Associate Editor for the *MIS Quarterly.*

PAUL CULE is an assistant professor of Management in the College of Business Administration at Marquette University. His research interests are in organizations in transition, information technology, reengineering, software development methods, and management, as well as the roles of IT in changing societies. He has been published in *Communications of the ACM, Database, Information Systems Management,* and the proceedings of ICIS and AIS. Prior to embarking on an academic career he spent 33 years in industry in Britain, Canada, and the United States. His experience

Journal of Management Information Systems / Spring 2001, Vol. 17, No. 4, pp. 5–36.

0742–1222 / 2001 $9.50 + 0.00.

spans manufacturing, marketing, software development, global product planning, and strategy development.

ABSTRACT: Advocates of software risk management claim that by identifying and analyzing threats to success (i.e., risks) action can be taken to reduce the chance of failure of a project. The first step in the risk management process is to identify the risk itself, so that appropriate countermeasures can be taken. One problem in this task, however, is that no validated lists are available to help the project manager understand the nature and types of risks typically faced in a software project. This paper represents a first step toward alleviating this problem by developing an authoritative list of common risk factors. We deploy a rigorous data collection method called a "ranking-type" Delphi survey to produce a rank-order list of risk factors. This data collection method is designed to elicit and organize opinions of a panel of experts through iterative, controlled feedback. Three simultaneous surveys were conducted in three different settings: Hong Kong, Finland, and the United States. This was done to broaden our view of the types of risks, rather than relying on the view of a single culture—an aspect that has been ignored in past risk management research. In forming the three panels, we recruited experienced project managers in each country. The paper presents the obtained risk factor list, compares it with other published risk factor lists for completeness and variation, and analyzes common features and differences in risk factor rankings in the three countries. We conclude by discussing implications of our findings for both research and improving risk management practice.

KEY WORDS AND PHRASES: Delphi technique, IS project risk management, IS risk management, risk assessment.

IT REMAINS A SAD STATISTIC that too many software development projects end in failure [11, 26, 39, 40]. Fully 25 percent of all software projects are cancelled outright [11]. As many as 80 percent of all software projects run over their budgets [40], with the "average" software project exceeding its budget by 50 percent [11, 18]. It is estimated that three-fourths of all large systems are "operational failures" because they either do not function as specified or are simply not used [11]. Hence it is no surprise that avoidance of failure is a dominant theme in the information systems (IS) literature (although it is often couched in more positive terms). Since a large proportion of the causes for late, over-budget delivery of software are management-related [39], the search for appropriate managerial action to solve this problem has been intense. Among advocated methods for improving software project management, the concept of software project risk management has gained prominence (see [4, 13, 19, 31]). Advocates of software project risk management claim that by identifying and analyzing threats to success, action can be taken to reduce the chance of failure [5, 6, 7].

The obvious first step in software project risk management is the identification of the risks to be controlled. Software project risk has been defined as the product of uncertainty associated with project risk factors and the magnitude of potential loss due to project failure [3]. Thus the key elements to be controlled are the project risk

factors [19]. Consistent with the views of March and Shapira [28] regarding management risk, we define a risk factor as a condition that can present a serious threat to the successful completion of a software development project.

Three questions about risk factors stand in the way of developing a more disciplined approach to software project risk management:

1. What are the typical risk factors software project managers face?
2. Which risk factors do software project managers consider more deserving of their attention?
3. Which countermeasures are the most effective in mitigating risk, given a particular set of risk factors?

In this paper, we address the first two questions, in order to lay a foundation for further research and the development of theory that might help us answer the third question. None of these questions has been adequately treated in the past literature. Though several lists of risk factors have been published in the literature (e.g., [2, 3, 4, 14, 29, 30, 32]), our understanding of the typical risk factors is still inadequate. Most of the published lists are relatively old and vary too much in their level of detail and scope to provide a foundation for more systematic risk detection and theory building. Two reasons for this are that they have been derived using limited samples, and the rigor of the data collection and analysis procedures used is questionable. Furthermore, the data collection has relied on samples taken from a single culture, possibly biasing findings. In addition, we currently lack knowledge of the relative importance of different risk factors, because previous studies have not been based on methods designed to yield valid and reliable rankings. In this paper, we begin to address these issues by developing a ranked risk factor list using a rigorous data collection and analysis procedure. The investigation was carried out in three different countries to lessen the effect of a single-culture bias and to broaden our view of risk factors and their ranking. These three countries not only represent different cultures, but also different socioeconomic contexts.

The remainder of the paper is organized as follows. First, we describe the background and motivation for our research. Second, we discuss the research design and methodology used in conducting the empirical study. Third, we present the major findings of the study by: (1) reporting a new list of risk factors and comparing this list with other risk factor lists (i.e., [3, 4, 5, 30]) for coverage and novelty, and (2) discussing the relative ranking of the risk factors. Next, in our discussion section, we speculate as to the source of some of the differences that were observed across the three panels and discuss the implications of our study for both research and practice. Finally, we discuss the limitations of the study, consider the potential for future research, and summarize the conclusions that can be drawn from our study.

Background

DIRECTLY OR INDIRECTLY, both the software project management literature and the IS implementation literature deal with the subject of project risk, but there has been only

limited cross-fertilization between these two streams of literature. Although the unification of these two streams is beyond the scope of this paper, we lay the groundwork for further research to bring them together.

The software project management literature views risk management as a two-stage process: assessing the risk, and taking action to control it [4, 7, 19]. The first stage, risk assessment, consists of three steps: (1) identification of risk factors, (2) estimation of the likelihood for each risk factor to occur, along with potential damage from the risk, and (3) an evaluation of total risk exposure. The sticking point here is the identification of the risk factors that need to be controlled. If there are no good mechanisms to help project managers identify all pertinent risk factors, the value of any risk management exercise is low. Several methods for identifying risk factors have been suggested, such as scenarios, examination of past or analogous situations, brainstorming, or other creative methods (see e.g., [4, 7, 14]). Most of these methods assume that managers have the requisite experience to be aware of all pertinent risk factors. However, this is not necessarily the case. Moreover, many of these methods can be time-consuming and thus too costly to use on a regular basis.

One popular method for identifying risk factors, therefore, has been the use of checklists. With a risk factor checklist, project managers can avoid overlooking some risk factors. Many such checklists can be found in the software project management literature. Boehm [4, 5] provides one typical checklist based on his informal discussions within TRW. Barki et al. [3] developed a comprehensive list of risk factors from the literature (including the implementation literature to that date) and organized them into five general risk categories by factor analyzing a composite list administered in the form of a survey. However, the studies upon which Barki et al. based their composite list have been shown to be flawed in their data collection methods [27]. Pointing up the flaws, Moynihan [30] showed some gaps in the Barki et al. list through an informal survey of 14 Irish software project managers.

The IS implementation literature represents another stream of research that has focused on the identification of factors that can affect project success. This body of literature includes two distinct substreams: "factor research" and "process research" [21, 24, 25]. Here, our focus will be on the factor research stream, since it is more directly related to the identification of project risk factors.

The factor research stream emerged in the late 1960s, in response to the growing number of implementation failures [37]. Both management scientists and IS researchers became interested in understanding the factors associated with success or failure in implementing systems. Factor research led to the identification of many variables that could potentially influence implementation. Alter [1], Davis [9], McFarlan [29], and others contributed their own evaluations of success and risk factors. Zmud [41] groups these variables into four categories: organizational characteristics, environmental characteristics, task characteristics, and individual characteristics. To these, Lucas [25] adds technical characteristics, client actions (e.g., top management support and user involvement), and attitudes toward the system. While the factor research stream has led to the nomination of hundreds of specific candidate factors, top management

support is the one that appears to be most consistent across multiple studies [12, 21]. Factor research has also emphasized high-quality system design and developer–user interaction [24].

The lack of cross-fertilization between the IS implementation factor stream and the risk management stream is somewhat puzzling considering the similarities in the two branches of research. Prior research in both of these streams has contributed to our understanding of project risk factors. However, there is still a lack of consensus regarding the risks that can affect software projects and their relative importance. There is also a lack of synthesis of previous findings. So, we believe that there are at least four compelling reasons for reexamining this topic.

First, in most previous studies the identification of risk factors was either not properly grounded or was based on a review of prior literature that is somewhat dated. When we say some prior studies were not properly grounded, we mean that either they did not collect data from project managers or they did not collect their data in any sort of systematic or scientifically rigorous way. Furthermore, the most widely cited studies on software risk (i.e., [2, 4, 9, 29]) drew their data from large, mainframe-based projects undertaken during the 1970s and 1980s. The organizational and technological landscape, however, has changed considerably during recent years: New organizational forms and systems development environments have emerged, new mechanisms have evolved for acquiring systems (e.g., outsourcing and strategic alliances), and the centralized, mainframe-centric systems architecture has given way to client/server computing [20, 33]. Hence the very dynamics of the software development environment may have rendered previous risk checklists obsolete. Today's applications often reach outside the organization and their development involves numerous parties and complicated organizational structures [10]. At the same time, techniques and technologies available for developing software have undergone significant improvements, which may have reduced the importance of some risk items (such as performance issues).

Second, none of the previous studies on software project risks systematically address the relative importance of various risk factors. Thus the inclusion criteria for factors that should be on a risk checklist and their ranking in terms of relative importance have remained nebulous.

Third, previous attempts to produce a comprehensive list have been limited by the lack of a cross-cultural perspective. Therefore, published lists may be biased by a specific culture's way of channeling management perception and its propensity to handle and manage risky situations. That is, almost all earlier studies are based on U.S. data alone. We believe it is necessary to widen our view and therefore conduct simultaneous surveys on risk factors in culturally different settings.

Finally, the research community would benefit from the unification of the software project risk research stream and the risk factor implementation research stream. Consequently, by comparing our findings to both streams, we can contribute to this unification. For all of the above reasons, there is ample justification for developing a fresh, up-to-date risk checklist.

Table 1. Cultural Dimension Scores for Panel Settings [15, 16]

	Cultural Dimension			
Settings	Power Distance	Uncertainty Avoidance	Individualism	Masculinity
Finland	33	59	63	26
Hong Kong	68	29	25	57
United States	40	46	91	62
Median of 53 countries	62	70	38	50

Design of the Study and Research Method

THE AIM OF THIS STUDY IS TO DEVELOP an authoritative list of risk factors, and to determine which of those risk factors are most important. An obvious source for such information is an expert in the field with years of experience in managing software development projects. However, a single expert is not likely to have the personal experience of all development situations needed to yield a comprehensive list of factors. To ensure a reliable and validated data collection process we must open our inquiry to divergent opinions and seek feedback-based convergence and closure on the factors that really count in software development. Therefore a ranking-type Delphi survey [36], designed to elicit the opinion of a panel of experts through iterative controlled feedback, was chosen as the research method for this study. We formed our panel of experts by recruiting project managers with many years of experience from three different socioeconomic environments. We also used systematic procedures to elicit and rank risk factors.

Composition of the Panels

To achieve variation in respondents' background and cultural settings, Hong Kong (HKG), Finland (FIN), and the United States (USA) were chosen as the target populations from which the panelists were drawn. These choices were not merely ones of convenience. On Hofstede's [15, 16] four-dimension scale of cultural differences (which includes uncertainty avoidance, individualism, power distance, and masculinity) HKG, the USA, and FIN differ markedly, as shown in Table 1. Given such differences, our sample provides the means to transcend the monocultural view of software project risk that dominates the literature. In addition to the cultural differences identified by Hofstede, the three countries also provide a contrast in socioeconomic conditions in that HKG represents extreme laissez-faire capitalism, FIN is a Scandinavian welfare state with considerable state intervention, and the USA is a managed market economy. In spite of these differences, all three countries are very similar in terms of advanced uses of information and communication technologies and each has a relatively long history of exploiting computing applications, which helps to ensure that the findings have some general validity.

We recruited the members of the three expert panels from among experienced project managers in each culture. We asked senior IS executives in major corporations in each locale to identify their most experienced and successful project managers. From among those candidates, we solicited participation from managers with at least five years of experience in project management. The demographics of the three panels (shown in Table 2) indicate that all of the panelists had impressive experience in the area of software project management.

The panels were formed on the basis of their cultural background. For example, we tried to avoid expatriate panelists in Hong Kong. We started with 11 panelists in HKG, 13 panelists in FIN, and 21 panelists in the USA. Since the analyses of the panels' responses would not be affected by panel size [36], there was no need to match the sizes of the panels. The aim was to form panels that were large enough to allow diversity of opinion, but small enough to allow the panelists to handle the volume of feedback. After the first phase, two panelists in HKG emigrated, reducing the panel size to nine. Two USA panelists also dropped out before the ranking phase, leaving the panel size at 19. The FIN panel remained stable throughout the survey.

Data Collection and Analysis Method

Data collection and analysis were based on Schmidt's [36] method, in which the Delphi survey process is divided into three phases, as shown in Figure 1. In the first phase, a *brain-storming* round is conducted to elicit as many items as possible from the panel(s). All panels participate cooperatively in the first phase to ensure that they are all working from a common list of items with common definitions. Each panelist was asked to submit at least six factors, and to provide short descriptions of the factors, to aid the researchers in their collation effort. Two of the authors worked independently to collate the responses. Then the two independently constructed lists were compared and reconciled by all authors working together. Exact duplicates were removed, and an explanation of the groupings was provided to the panelists. The combined list was circulated to all panelists for corrections, additions, and eventually validation. This sort of validation has not been present in any previous study. Thus, we started with a common list with common definitions that we could analyze separately within each of the three panels. Thus we could be assured that when the same factor was selected by more than one panel it would indeed be the same factor, not just a "similar" factor.

In the second phase of the study, we separated the panels by country, allowing each panel to independently pare down the list of risk factors. We sought to *narrow* the list of factors to a manageable number (a target of about 20 factors has been suggested [36]) so that they could be meaningfully ranked by each panel. Rather than force the size of the list ("top ten," etc.) we let the panelists decide how many factors should be included. Working from a randomized list of risk factors, the panelists were asked to choose (but not to rank) at least ten factors that they considered most deserving of their attention and resources. Our criterion for narrowing the list was that factors

Table 2. Panel Demographics

	USA Panel			FIN Panel			HKG Panel			Composite
Characteristic	Avg	Max	Min	Avg	Max	Min	Avg	Max	Min	(Avg)
Total Employees in Panelist's Company	10574	70000	103	3693	15000	260	3039	7000	500	6616
IS Employees in Panelist's Company	292	1200	20	156	500	50	101	400	18	203
Panelist's Work Experience (Years)	15.3	25	7	20.1	30	9	13.2	20	9	16
Panelist's Educational Level*	BD	PD	AD	PS	PD	HS	BD	PS	HD	BD
Number of Projects Panelist Has Managed	35.8	155	2	13.5	40	3	10.3	20	5	22
Smallest Project Managed by Panelist (Person-Months)	14.4	72	0.5	28.1	192	2	16.6	54	1.5	19
Largest Project Managed by Panelist (Person-Months)	887.6	3600	30	484.1	1800	60	335.6	1152	80	633

* "Education Level" is the highest level attained by the panelist: HS = High School, HD = Higher Diploma (usually one or two years past high school), AD = Associate's Degree (two years of university level education), BD = Bachelor's Degree, PS = Some university study beyond the Bachelor's Degree, PD = Post-graduate Degree (Master's or Doctorate).

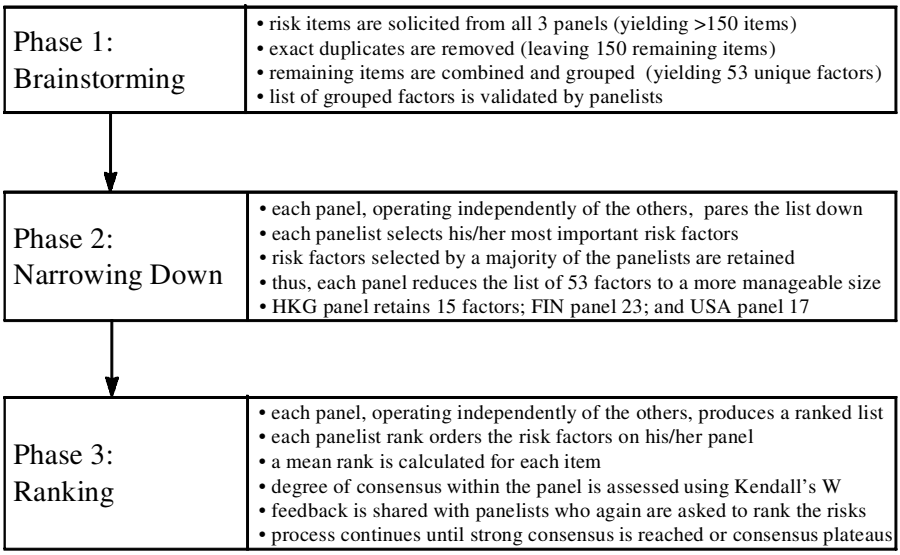

Figure 1. Description of Delphi Survey Process Used in This Study

chosen as important by over half the panelists were retained for the ranking phase. Through this simple majority vote, the panels effectively reduced the lists to manageable sizes and the resulting lists reflected a majority opinion of the importance of these items within each panel. Through this process, the HKG panel reduced the list to 15 factors, the FIN panel chose 23 factors, and the USA panel pared the list to 17 factors (see Appendix 1).

The *ranking* of the selected factors was done in the third phase. Here panels ranked the risk factors in order of priority—that is, the top-ranked factor would be the factor most deserving of the project manager's attention. The panelists were also asked to rate the risk factors according to their relative importance to the successful completion of a project. The importance ratings were used in a separate analysis of risk management behavior, which was reported elsewhere [23]. Multiple ranking rounds were conducted until each panel reached an acceptable level of consensus. In this study we measured the degree of consensus among the panelists using Kendall's coefficient of concordance (W) [36, 38]. To minimize bias in ranking for the first round, the factors to be ranked were provided in a different random order for each panelist. In subsequent rounds the factors were listed in the order of their average received ranks within each panel. The ranking rounds stopped when either: (1) the coefficient of concordance indicated a strong consensus (W>0.70), or (2) the level of consensus for the panel leveled off in two successive rounds. Round-by-round results are shown in Appendix 1, Tables A1.1, A1.2, and A1.3. The USA panel reached strong agreement (W>0.70) after three rounds, the HKG panel had moderate agreement for the second and third rounds (W>0.50), and the FIN panel required four rounds before their consensus plateaued at the moderate level (W>0.50).

Results

New Risk Factor List

THE LIST OF RISK FACTORS IDENTIFIED IN PHASE 1 is shown in Table 3. We have organized it into a set of 14 groups based on the source of the risk.[1] The panelists were asked to make corrections and validate the groups. As a result of this a few changes and clarifications were made to the classification. (See also Appendix 2, Table A2.1.)

As noted earlier, our first objective was to develop a list of risk factors that would be authoritative and have a wide coverage of possible risks. We expected some differences in our list and the union of previous lists. Given the radical changes that have occurred in the computing landscape (and the business environment), we expected to find that (1) some risk items have remained relatively stable while (2) others have declined in importance over time. Also, since previous studies have tended to generate risk lists based on the literature or on unsystematically gathered accounts from project managers, we expected that (3) the list resulting from our disciplined Delphi approach would contain some unique items that were not detected in earlier studies.

To address the three points outlined above, we compared our list to a merger of other risk item lists (a systematic comparison with them is shown in Appendix 3, Table A3.1). Barki et al. [3], Boehm [4], and Moynihan [30] present lists of risk factors. A union of these three lists was therefore used for this comparison. The results of this analysis are highlighted in Table 3 (bold items represent risk factors that are not represented in earlier lists) and illustrated in Figure 2. The analysis suggests that our list is more encompassing, and, due to its elicitation procedure, also more reliable.

The first subset of risk factors we consider are those that we expected would remain stable over time. Although there are 27 risk factors identified by our panels that could be matched in some way with 29 of the 33 factors in the combined list, there is not a strict one-to-one correspondence (see Appendix 3). Two factors, *"project size"* and *"staffing level/team size,"* were not specifically mentioned by our panels. These risk factors may serve as general surrogates for more specific risks that appear on our list. Thus, we have matched these two factors with items that do appear on our list. For example, project size is likely to influence a number of other risk areas. The notion of project size is partially addressed by both *"scope creep"* (5.3) and *"number of organizational units involved"* (5.5). Scope creep is a function of not understanding the size of the development effort properly (task size/complexity) while the number of organizational units measures the organizational span (size) of the project. In a similar manner *"team size"* was not specifically identified in our list, though some of our items do encompass this element indirectly. Also, team size is often directly related to project size. It is also possible that project size is regarded as a "given," rather than something that can be managed, and thus was not nominated by our panelists for inclusion in the list.

In regard to the second issue, our analysis detected four risk factors that have been recognized in earlier studies but are not represented on our list (and are thus not listed

Table 3. Full List of Risk Factors

1. Corporate Environment

1.1 *A climate of change in the business and organizational environment that creates instability in the project.*
1.2 *Mismatch between company culture and required business process changes needed for new system*. A mismatch between the corporate culture and the changes required by the new system.
1.3 *Projects that are intended to fail:* Projects started for political reasons that carry no clear business value, but serve to divert the organization's focus from actual needed change. Such projects are underfunded, not supported, and are not intended to succeed. Projects have no business value and are used as diversionary tactics to avoid facing the real change needs.
1.4 *Unstable corporate environment:* Competitive pressures radically alter user requirements, sometimes making the entire project obsolete.
1.5 *Change in ownership or senior management:* New owners and/or managers set new business direction that causes mismatch between corporate needs and project objectives.

2. Sponsorship/Ownership

2.1 *Lack of top management commitment to the project*. This includes oversight by executives and visibility of their commitment, committing required resources, changing policies as needed.
2.2 *Lack of client responsibility, ownership, and buy-in of the project and its delivered system(s).*
2.3 *Failure to gain user commitment:* Laying blame for "lack of client responsibility" on the project leader rather than on the users.
2.4 *Conflict between user departments:* Serious differences in project goals, deliverables, design, etc., calls into question concept of shared ownership.
2.5 *Failure to get project plan approval from all parties.*

3. Relationship Management

3.1 *Failure to manage end-user expectations:* Expectations determine the actual success or failure of a project. Expectations mismatched with deliverable—too high or too low—cause problems. Expectations must be correctly identified and constantly reinforced in order to avoid failure.
3.2 *Lack of adequate user involvement:* Functional users must actively participate in the project team, and commit to their deliverables and responsibilities. User time must be dedicated to the goals of the project.
3.3 *Lack of cooperation from users:* Users refuse to provide requirements and/or refuse to do acceptance testing.
3.4 *Failure to identify all stakeholders:* Tunnel vision leads project management to ignore some key stakeholders in the project, affecting requirements definition, implementation, etc.
3.5 *Growing sophistication of users leads to higher expectations:* Users are more knowledgeable, have seen sophisticated applications, apply previous observations to existing project.
3.6 *Managing multiple relationships with stakeholders:* Some "clients" are also "partners" in producing deliverables in other projects. Leads to confusion of roles and responsibilities.
3.7 *Lack of appropriate experience of the user representatives:* Users assigned who lack necessary knowledge of the application or the organization.

Continued

Table 3. Full List of Risk Factors (Continued)

4.	**Project Management**
4.1	***Not managing change properly:*** **Each project needs a process to manage change so that scope and budget are controlled. Scope creep is a function of ineffective change management and of not clearly identifying what equals success.**
4.2	***Lack of effective project management skills:*** **Project teams are formed and the project manager does not have the power or skills to succeed. Project administration must be properly addressed.**
4.3	***Lack of effective project management methodology:*** **The team employs no change control, no project planning or other necessary skills or processes.**
4.4	*Improper definition of roles and responsibilities:* Members of the project team and the organization are unclear as to their roles and responsibilities. This includes outsourcers and consultants.
4.5	***Poor or nonexistent control:*** **No sign-offs, no project tracking methodology, unaware of overall project status, "lost in the woods."**
4.6	***Poor risk management:*** **Countering the wrong risks.**
4.7	***Choosing the wrong development strategy:*** **e.g. waterfall, prototyping, etc.**
5.	**Scope**
5.1	*Unclear/misunderstood scope/objectives*. It is impossible to pin down the real scope or objectives due to differences or fuzziness in the user community.
5.2	***Changing scope/objectives:*** **Business changes or reorganizes part way through the project.**
5.3	*Scope creep:* Not thoroughly defining the scope of the new system and the requirements before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.
5.4	*Project not based on sound business case:* Users and developers ignore business requirements, develop system for sake of technology.
5.5	*Number of organizational units involved:* increased number of lines of communication and conflict potential expands the scope of the system.
6.	**Requirements**
6.1	*Lack of frozen requirements*. Because the needs of the users change, the requirements change. Consequently the system will never be moved into production because none of the requirements are ever completed. Alternatively, freezing a subset of the functionality and delivering allows for the completion of the system and update releases as required.
6.2	*Misunderstanding the requirements*. Not thoroughly defining the requirements of the new system before starting, consequently not understanding the true work effort, skill sets and technology required to complete the project.
6.3	*New and/or unfamiliar subject matter for both users and developers:* Lack of domain knowledge leads to poor requirements definition.
7.	**Funding**
7.1	*Underfunding of development:* Setting the budget for a development effort before the scope and requirements are defined or without regard to them (i.e., picking a number out of the air).
7.2	*Underfunding of maintenance:* Support for products in the maintenance phase. If the customer is unprepared or does not budget for this, the project can be judged a failure even if successful in all other aspects.

7.3 *Bad estimation:* Lack of effective tools or structured techniques to properly estimate scope of work. Unrealistic cost estimates cause illogical or suboptimal planning, strategy, and decisions.
7.4 "*All or nothing":* Requires budgeting entire project at the outset, leading to under funding in later years of project.

8. Scheduling

8.1 *Artificial deadlines*. Presence of unrealistic deadlines or functionality expectations in given time period. "Crash projects" in which test time or training time is reduced—using something other than work effort required to determine when the new system should move into production.
8.2 *"Preemption" of project by higher priority project:* Management unable to resolve conflicting schedule demands.

9. Development Process

9.1 *Lack of effective development process/methodology:* Leading to quality problems—Documentation, Software and Testing—poor estimating—insufficient time for up-front work, for example, design—little flexibility for change—insufficient testing.
9.2 *Trying new development method/technology during important project.*

10. Personnel

10.1 *Lack of required knowledge/skills in the project personnel:* for example, technology, business knowledge, and experience.
10.2 *Lack of "people skills" in project leadership:* PM tries to "manage" schedules, technology, requirements, etc., ignoring that management is dealing with people on the team.
10.3 *Poor team relationships:* Strains existing in the team due to such things as burnout or conflicting egos and attitudes.

11. Staffing

11.1 *Insufficient/inappropriate staffing:* Not enough people or people with wrong skills/insufficient skills assigned to project, regardless of availability.
11.2 *Staffing volatility:* At some point in the project, losing the key project manager, analysts or technicians (especially in new technology).
11.3 *Excessive use of outside consultants:* Can lead to a conflict of interest, for example, billable hours vs. budget, or resulting in the internal staff not having significant involvement
11.4 *Lack of available skilled personnel:* People with the right skills are not available when you need them.

12. Technology

12.1 *Introduction of new technology:* Using new, or "bleeding edge," technology that has not been used successfully at other companies, or major technological shift occurs during the project.
12.2 *Stability of technical architecture:* Has to be done before comparable applications.

13. External Dependencies

13.1 *External dependencies not met:* The project's consultants or vendors do not deliver, go out of business, or are unclear as to their roles and responsibilities.

Continued

Table 3. Full List of Risk Factors (Continued)

13.2	*Multi-vendor projects complicate dependencies:* Integration of packages from multiple vendors hampered by incompatibilities and lack of cooperation between vendors.
13.3	***Lack of control over consultants, vendors, and subcontractors:* Schedule or quality problems beyond control of project manager. No legal recourse due to poor contract specification.**
14.	Planning
14.1	***No planning or inadequate planning:* Attitude that planning is unimportant or impractical.**

Note: Bold items represent risk factors not observed in earlier lists (i.e., [3, 4, 30]).

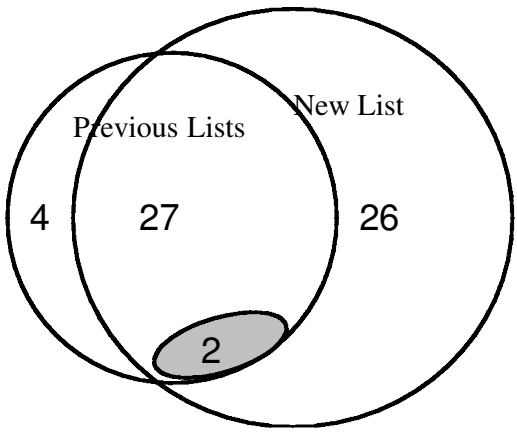

Figure 2. Comparison of Risk Factor Lists

in Table 3). Interestingly, among these are the only two risks appearing on the combined list that directly relate to technological problems—namely *"real time performance shortfalls"* and *"developing wrong interface."* These two items are from Boehm [4]. It appears that the importance of these technological risk factors may have diminished over the last 10 years, perhaps due to better performance and scalability of hardware and software, and the widespread adoption of graphical user interfaces.

The other two risk factors that were not matched with any factors mentioned by the panelists in our study were *"team members have not worked together before"* and *"type of users."* But these two factors were not validated by Barki et al. [3], and thus were dropped from their final list after they had completed their survey.

The third issue involves the risk factors not mentioned in previous studies. As noted, the panels brought up 26 new factors (as indicated by the bold items in Table 3). Thus our list greatly increases the coverage of known risk factors and also suggests that some new elements of risk have emerged during the last decade. Three major groups

of risks and several new, unexplored facets of software project management surface from these 26 new factors.

A large number of the new risk factors relate to *managing user relationships and user expectations.* Though user-related risks have been recognized in the past ("unwilling users," "user resistance") the level of detail in clarifying user-related risks in this study is considerably higher. For example, today users must take responsibility for system development and the resulting system. If this is not possible, the project becomes more risky (Factor 2.3). Users must also be more involved in development activities (Factor 3.2). Moreover, users' expectations must be set realistically (Factor 3.1). Interestingly, many of these findings are similar to those obtained from implementation research [8, 24].

A second major topic deals with inadequate *project management methodologies and project management skills.* This is an interesting finding, as it points out the awareness and need for disciplined management practices and recognizes their absence as an important source of risk (Factors 4.2, 9.1, 9.2).[2] Although such factors have not been recognized in previous risk management research, they have been raised in process improvement research (e.g., [17]). The implementation literature has also stressed the need for better project management skills (e.g., [8]).

The third major topic of new risk items deals with the *turbulence of the business environments* in which systems development takes place. This topic represents a largely unexplored area in software risk management, software project management, and implementation research as well, though it received partial mention by Moynihan's project managers [30]. Nowadays, systems can become obsolete overnight due to changes in business strategy. Therefore the demand for short delivery times is high (Factor 5.2). Also associated with the turbulence in the business environment were changes in business ownership or senior management (Factor 1.5) and impacts of the business environment (Factor 1.1).

Although organizational politics and organizational culture (Factors 1.1, 1.2, 1.3) were not reflected in the previous risk lists, these issues have long standing in the IS literature [8, 24], and thus do not represent new findings per se.

The remainder of the new risk factors address largely unexplored areas in software project risk management. These deal with such topics as the *diversity and multiplicity of stakeholders and user communities* (Factors 3.4, 3.5, 3.6); *critical aspects in the management environment* (Factors 4.6, 4.7, 14.1); and the *evolution of the IT infrastructure* (Factor 12.2). Many of these are not just new factors—they are new themes that invite research and that reflect the dynamic nature of IS development and the challenges it creates for project managers.

Given the number of "new" risks identified in this study, it is important to ask the question, "Are these risks really new?" Had earlier risk studies been conducted in a systematic manner similar to this study, it is conceivable that such risk factors would have surfaced. But it is more likely that the increasing pervasiveness of IT in all aspects of our daily lives has sensitized executives to the need to exercise much more oversight on IS projects. This, in turn, has pushed these risks to prominence in the minds of project managers.

Ranking of Risk Factors

In Table 4 we summarize the eleven factors common to all three panels in the order of their average relative ranks. It is derived from the final results of all rankings, as exhibited in Appendix 1. The table also shows final rankings for the 29 factors ranked by all three panels in whole ranks. Some factors that one or two panels felt to be more deserving of attention were not ranked by others. For example, the top-ranked factor for the FIN panel was "lack of effective project management skills." This was also ranked fifth by the USA panel, but the HKG panel did not select this item for ranking at all. This supports our assumption that sampling from a single culture would tend to leave us blind to many important risk factors. These factors are situation-dependent, so it is difficult to generalize a "top-ten" risk factor list across different cultural settings. For example, if we restrict ourselves to the top five factors for each panel, then the three panels have only one item in common.

If we are looking for something that all three panels had in common, then we should look at the entire list of 53 items in Table 3. A common unifying theme in this list, as noted above, is that not a single one of the risk factors included has much to do with the traditional "technical" aspects of software development, such as choice of programming language, operating system compatibility, equipment constraints, or user interfaces. One major technical concern relates to architectural stability. Another item we can classify as being "technology" combines aspects of managerial choices about technology and the management of change entailed by its introduction into the organization (Factor 12.1).

The ranking exercise, overall, confirms that the lack of top management commitment is the factor that software project managers felt was most deserving of their attention (ranked first for the USA and HKG panels, second for the FIN panel). Although this is not a novel finding (top management support is a recurring theme in the implementation literature), it is an issue that has been ignored in some earlier rankings (e.g., Boehm's list). The emphasis placed on this factor was prominent to the extent that some panelists referred to it as a "fatal" factor in its own right. They chose the term "commitment" rather than "support" to indicate the strong, active role top management must play in the project from initiation through implementation. Given the importance ascribed to this factor, there is certainly justification for further research to determine the means by which project managers can ensure that they have the appropriate type and level of commitment from senior management [35].

Beyond this obvious consensual top choice, it is difficult to order the remaining factors universally across all three panels with much statistical confidence. The panelists were also asked to rate risk factors on a sliding scale of "importance." In doing so they were allowed to assign the same rating to more than one risk factor. These results were not useful for ranking purposes, but did help us group some risk factors according to their relative importance. Using this scale, three factors stood out as receiving a high rating from all three panels: (1) lack of top management commitment, (2) failure to gain user commitment, and (3) misunderstanding the requirements [23].

Table 4. Final Rankings and Composite Ranks

Ranks				
Composite	HKG	USA	FIN	Risk Items
1	1	1	2	Lack of top management commitment to the project
2	3	4	8	Failure to gain user commitment
2	7	2	6	Misunderstanding the requirements
4	2	6	11	Lack of adequate user involvement
5	13	11	3	Lack of required knowledge/skills in the project personnel
6	8	14	9	Lack of frozen requirements
7	5	10	19	Changing scope/objectives
8	12	12	13	Introduction of new technology
9	9	7	23	Failure to manage end user expectations
10	15	13	15	Insufficient/inappropriate staffing
11	10	16	22	Conflict between user departments
	4			Lack of cooperation from users
	5			Change in ownership or senior management
	11		17	Staffing volatility
	14			Lack of effective development process/ methodology
		3	4	Not managing change properly
		5	1	Lack of effective project management skills
		8		Lack of effective project management methodology
		9		Unclear/misunderstood scope/objectives
		15	20	Improper definition of roles and responsibilities
		17		Number of organizational units involved
			5	No planning or inadequate planning
			7	Artificial deadlines
			12	Multi-vendor projects complicate dependencies
			10	Lack of “people skills” in project leadership
			14	Trying new development method/technology during important project
			18	Bad estimation
			16	New and/or unfamiliar subject matter for both users and developers
			21	Poor or nonexistent control

To understand more deeply to what extent the panels agreed on the relative importance of various risk factors, we compared their lists statistically. This analysis focused on the set of 11 risk factors that were ranked by all three panels. Their level of agreement on the relative ranking of these 11 items was moderately strong ($W = 0.689, p < 0.05$) though the absolute ranking of these factors (in terms of rank number given in each panel) differed markedly. This is partly due to the different list lengths, but there are also some obvious disagreements on priority. For example, the fourth item on the composite list, "Lack of adequate user involvement," was ranked second on the HKG list, but was near the middle of the USA and FIN lists.

We used Kendall's rank-order correlation coefficient (T) to make pairwise comparisons between the country panels, in order to find out to what extent each pair of panels agreed on their rankings. The USA panel had good agreement with both the HKG panel ($T = 0.527, p < 0.025$; 11 items), and the FIN panel ($T = 0.505, p < 0.01$; 14 items). But HKG and FIN panels did not agree on the relative ranking of the 12 items they listed in common. This would suggest that using the USA panel's judgment in isolation would not provide a good universal surrogate, as important outlying factors would be ignored. These outlying factors bring new insights into what constitutes project management risk in the eyes of the project managers.

It is clear from these results that the three panels perceive the relative importance of the solicited risk factors differently. This fact in itself vindicates our original decision to draw panels from three different cultural environments. But what, exactly, motivates managers to rank the risk factors as they do? According to the management risk literature (e.g., [28]), the level of control that can be exerted over risks is fundamental to understanding how managers actually regard risk. Generally speaking, risks that cannot be controlled at all are not seen as risks. Instead, managers generally regard risk as something that can be influenced [28].

It is interesting to note that some of the risks included in our list are beyond the control of the project manager. Those *outside risks* over which the project manager has no control are often perceived as "acts of God," like hurricanes and earthquakes. At the other end of the spectrum are *inside risks,* which the project manager can monitor and control. These risks are theoretically manageable to zero. Between these two ends of the spectrum, however, lies a middle ground of *outside risks* over which the project manager has limited, or shared, control and influence. In the "limited control/influence" category lie those risk factors that rely on shared cooperation between the project manager and the rest of the organization. Table 5 shows how the 29 ranked risk factors in Table 4 were classified using these three categories. Two of the authors independently classified the risk factors into these three categories. Then all four authors worked together to reconcile differences in categorization.

An analysis of the rankings using this categorization scheme provides several insights. First, consistent with March and Shapira's [28] observations on how managers view risk, *outside risks* over which the project manager has *no control* were not generally selected for ranking, and those that were selected did not rank very highly. The HKG panel selected 3 such items (out of 15), with an average rank of 8, the USA panel picked 2 (out of 17), with an average rank of 10, and the FIN panel selected 4

Table 5. Comparison of Software Risk Factors Selection

Risks Outside Purview of PM	Risks Within Purview of PM	
No Control or Influence	Limited Control or Influence	Complete Control
Conflict between user departments UHF	Lack of adequate user involvement UHF	Failure to manage end user expectations UHF
Change in ownership or senior management H	Changing scope/objectives UHF	Not managing change properly UF
Staffing volatility HF	Lack of cooperation from users H	No planning/inadequate planning F
Number of organizational units involved U	Failure to gain user commitment UHF	Lack of people skills in project leadership F
Multi-vendor projects F	Artificial deadlines F	Poor or nonexistent controls F
New subject matter F	Lack of top management commitment UHF	Unclear/misunderstood scope/objectives U
	Improper definition of roles and responsibilities UF	Misunderstanding the requirements UHF
	Lack of frozen requirements UHF	Bad estimation F
	Introduction of new technology UHF	Lack of effective project management method U
	Lack of knowledge/skills in project personnel UHF	Trying new development method F
		Lack of effective development process H
		Lack of effective project management skills UF
		Insufficient/inappropriate staffing UHF

Key: U = selected by USA panel, H = selected by HKG panel, F = selected by FIN panel.

(out of 23), with an average rank of 17. Further analysis reveals that the panelists tended to select and rank highly those risk items over which they had limited control or influence. The HKG panel picked 8 such items (out of 15), the FIN panel selected 9 items (out of 23), and the USA panel picked 8 items (out of 17). Consistent with what March and Shapira's [28] observations would suggest, the relative ranks of these items were much higher (HKG 4.7, USA 8.3, and FIN 6.5). Finally, risk items under the direct control of project managers (*inside risks*) were ranked *lower* than outside risks over which project managers had only limited control or influence. The HKG panel picked four such items (out of 15), the USA panel 7 (out of 17), and FIN 10 (out of 23). Here a much higher variation between panels could be observed, as discussed in the next section. This finding is also in consonance with March and Shapira's observations in that the managers appear to be looking at the magnitude of loss associated with the risks and not the expected loss, since the probabilities associated with some of the risks would argue against the relative rankings as assigned by the panels.

Discussion and Implications

IN THIS SECTION, WE FIRST DISCUSS SOME CULTURAL FACTORS that may have contributed to the differences among the panels in their relative ranking of risk factors. Then we examine some of the implications of our findings in terms of research and practice.

Differences Between Risk Factor Rankings

There is a clear need for some explanation of the wide variations between panels when the risks are categorized according to level of control. Although no specific follow-up research has yet been done to seek rigorous explanations, the authors can draw upon their own considerable knowledge of the three cultures to speculate on a number of possibilities as a basis of discussion on the differences in ranking.

Perceived level of control relates clearly with cultural differences in individualism, power distance, and uncertainty avoidance. Cultures with a collectivist philosophy can be expected to avoid attributing risks to an individual. When the project manager is also in a position of pronounced subservience, the perceived loss of control over outside risks is strong. As we might expect, the HKG panel selected 8 of their 15 factors from among those beyond the purview of the project manager (see Table 5). In contrast, both the FIN panel (15 factors) and the USA panel (11 factors) placed a clear majority of their choices among those that were attributable to the project manager.

If we narrow our view to those factors with a mean rank less than 10, we find that the FIN and USA panels chose 6 out of 9 and 7 out of 10, respectively, in the category under the purview of the project manager. The HKG panel placed only 3 out of 11 in this category. This suggests that the choices of risk factors in the HKG panel may be partly attributable to the culturally based philosophy of HKG managers. It thus suggests a fundamental difference between the FIN/USA and the HKG panels in terms of risk factor perception. Furthermore, the HKG panel identified most threats to success

as being due to outside agencies not under the control of the individual. We believe this is also consistent with Hofstede's characterization of cultural differences among the countries in our study on his "individualism" dimension. HKG managers work under the assumption that responsibility is shared. This may also help explain why the HKG panel tended to ignore those risks that are within the purview of the project manager. Finally, the choice of risks outside the purview of the project manager possibly reflects the greater power distance that exists in HKG society, which is influenced by Confucian ethics. Factors that depend on the actions of superiors are considered very risky by HKG managers because they feel at a loss to influence their superiors' actions.

Another possible cultural difference in the selection and ranking of risk factors is seen in the focus on the capabilities of the project manager by the FIN panel, and the lack of such risk factors on the HKG panel's list. The HKG panel (and to a certain extent the USA panel) represents a "masculine" culture, where personal inadequacies are not readily admitted. On the other hand, the FIN panel represents a culture with very low masculinity and a strong Protestant ethic. It is in the nature of people in this culture to be very self-critical, so their focus on their own lack of project management skills makes sense in this light.

In addition to possible cultural biases in the evaluation of risk factors, we must also note that other differences in the socioeconomic environments of the three countries may have affected the choice and ranking of risk factors. Here, some specific aspects stand out. For example, HKG managers have had to cope with a very dynamic staffing situation due to the very mobile population of the territory. Emigration regularly siphoned off experienced people from the workforce. Thus HKG managers may have become more sensitized to the risks posed by personnel turnover as well as sudden changes in ownership or senior management. Accordingly, the HKG managers look on these risks as something they can manage through proper preparation, whereas FIN and USA managers perceive these more as "acts of God."

In sum, the introduction of the possible effects of differences in cultural background and socioeconomic conditions on the three panels enriched our results by widening the scope of risk factors under consideration. Furthermore, the differences in list composition and ranking of factors help us understand the roles responsibility and control play in the perception of the relative importance of risk factors and the fact that they do vary across cultures.

Implications for Research

One of our objectives in conducting this research was to build a foundation for theory-building about IS project risk management. With over 30 years of intensive research, we still lack a basic understanding of *why* certain managerial actions serve to mitigate multiple risks, and under what conditions these actions are no longer effective. When setting about to discover what practicing managers do, if researchers overlook some risk factors, then the subjects of their studies are not asked to respond to those factors. Thus some of their practices are not captured (see, e.g., [34]). By developing a more

comprehensive list of risk factors, we provide a basis for a more complete investigation of risk mitigation strategies.

Another objective of this study was to show that by opening inquiry to multicultural sources we can broaden our understanding of specific problems. The differences in the contributions of specific risk factors by each country panel, as well as their differences in ranking risk factors, demonstrate that the multicultural research approach contributed significantly to the value of our findings.

A third research objective was to contribute to the unification of the software project risk stream and the implementation stream. Our risk factor-oriented research actually confirms many of the factors found to be important in the implementation literature. Many of these factors have not been adequately treated in the risk management literature. The implementation literature continues to expand its set of factors by examining specific application domains, such as client/server computing (e.g., [8]). The collection of a complete set of risk factors is thus a continuous task with a potentially high payoff. In this study, we have shown that both the risk factor stream and the implementation stream can profit from cross-feed of results to provide a broader understanding of software project risk and management practice.

Implications for Practice

This study has several implications for practitioners. First, the list of 53 risk factors identified by our panelists provides managers with a more comprehensive checklist that can be used in developing software project risk assessment guidelines. Boehm [4, 5] and others have argued that risks must be identified before they can be addressed. The value of such a checklist was made clear to us when several of our panelists requested our permission to use the output of this research for exactly this purpose. As a result of such requests, several companies that participated in the study have now incorporated the risk checklist into their software development methodology and other companies can follow suit. The checklist (Table 3) has the merit of being more comprehensive and grounded compared to earlier lists due to its derivation method: It was created by three panels of practicing, experienced project managers using a rigorous methodology. Our list includes some new factors, but also a number of risk factors that have been reported in the implementation literature but not highlighted in the risk management literature.

Second, our study suggests that managers are biased in their attention to particular risk factors at the expense of other factors. IS project managers appear to be spending more of their mitigation efforts on risk factors that are outside their own immediate control, but over which they have some degree of influence. They tend to ignore risk factors that they perceive to be "acts of God," considering these factors to be unmanageable. They also are not worried about risk factors arising from activities under their direct control, which they can "manage to zero." Another factor causing bias in attention to particular risk factors is the fact that our panelists appeared to evaluate the magnitude of a potential loss due to a risk, without regard to the probability of the risk actually having an effect on their project. Rational models of decision-making, such

as the model used by Charette [7], assume that managers evaluate risk factors by assigning them a value based on the product of magnitude of loss and the probability of the event occurring. The actual behavior of our panelists likely reflects the attitudes of most practicing IS project managers, and confirms the findings of March and Shapira [28]. Managers should be aware of this bias, and make a conscious effort to properly evaluate risk factors before prioritizing the risks for management attention.

Third, our data suggest that managers should be aware of possible cultural or environmental factors that affect both the identification and perceived importance of various risks. Project managers should seek to overcome such bias. Highly ranked items in each country will in all likelihood be the ones receiving the greatest management focus. Since managers can be unduly influenced by their own cultural background, this may lead to a potential shortfall in risk recognition, resulting in ineffective risk management as some risk factors will go unnoticed. Thus, IS project managers in different countries/cultures may tend to focus their risk mitigation energies differently. One way to combat this is to use an extensive risk factor list like the one suggested in Table 3, as opposed to relying on a country-specific list that is both narrower and subject to cultural bias.

There is, however, a cost associated with relying on a risk checklist that contains 53 separate items. Such a list may prove too cumbersome and complicated to use in practice both because of its size and its lack of a coherent structure. To overcome these two limitations (i.e., size and structure), researchers have pointed out that risk management can be facilitated if long detailed lists could be grouped by their source [5] into a more systematic framework. In Appendix 2 we show how the risk factors solicited in this study were grouped by source. Managers could use this list of 14 groups to reduce the cognitive complexity of dealing with a large list of risk items.

For many managers, however, even a list of 14 risk groups can be cumbersome to deal with. A simpler classification, which suggests appropriate risk mitigation behaviors, was presented in another, practitioner-oriented paper [23]. However, that framework was developed through speculation based on our collective managerial experiences. Further research is needed to develop a theoretically sound classification.

Conclusions

PRIOR LITERATURE SUGGESTS THAT AN ADEQUATE ASSESSMENT of software project risk is a major source of problems in IS development projects [3, 5, 19, 29]. Keil [22] suggests that in order to reduce an organization's exposure to failure in IS projects, managers should assess risks early and constantly throughout the software development process. However, proper risk assessment and the development of strategies to counter the risks requires an understanding of (1) what the actual risks *are,* (2) which of these risks managers perceive to be more deserving of their attention, and (3) how these risk perceptions differ from one culture to another. In this paper, we have taken some steps toward addressing each of these points.

First we used a systematic and rigorous procedure to identify software project risks, developing for the first time an authoritative list of risk factors upon which

further study and theory development can be based. Since the list was based on input from a multicultural set of 41 practicing project managers, we can be fairly confident that it is comprehensive and well grounded. Second, we have identified those risk factors that the Delphi panelists ranked as being the most deserving of their attention in a software project. Third, we have demonstrated that while there is a substantial area of agreement across different cultures on what *some* of the major risks are, there are also discernible differences in the identification and perceived importance of certain risks.

Since our study employed a different methodology than previous research, this impedes our making direct and strong comparisons with earlier studies. Nevertheless, it is interesting to note how our findings compare with previously published literature in the areas of software project risk and implementation success factors. A comparison of the list of risk factors generated in this study with those reported in previous studies (i.e., [3, 5, 30]) suggests that the current list is a fairly comprehensive one. More risk factors were identified in this study than were validated in previous studies and (with very few exceptions) our list of risk factors includes practically all of the major risk elements that were previously identified.

Further comparisons between the risks identified in this study and those reported in previous studies suggest that project managers' risk concerns may have changed somewhat during the last decade, but of course it is difficult to say this with certainty because of differences in how the studies were conducted. Where we do find a marked similarity, however, is with the contingent success factors reported in the IS implementation literature. These results suggest that project managers are more concerned than ever before with implementation-related risks, such as commitment from top management as well as users (i.e., relationship management). In addition, risks associated with requirements determination, estimation and scheduling, and personnel have remained high on the management agenda. Furthermore, as we noted earlier, the risk factors identified and ranked were almost exclusively nontechnical.

Limitations of the Study

As with any Delphi-type study, the results are based on a limited number of subjects. While subjects were chosen for their experience in managing software projects and their cultural background, we can make no claim about the representativeness of our sample. Our panelists were not chosen randomly and we did not attempt to control for either the type of industry or type of project. Having said this, we believe that the sample is relatively diverse in that it includes project managers with extensive experience managing both small and large projects working for companies in a wide range of industries with both medium and large information systems departments.

Another limitation of the study is that it focuses on three specific countries. Therefore it is difficult to know the extent to which our findings generalize to other countries throughout the world. Furthermore, we can only rely on our own personal knowledge of the cultures involved to speculate as to the differences observed across the three country panels that were included in our study. We also relied upon cultural

differences measured by Hofstede [15, 16] and did not actually make these measurements ourselves.

Another significant limitation is the lack of theory to support our investigation. The software project risk literature provides little help in this regard. We look on the current work as the foundation for developing theory that is specific to software project risk management. Only with the rigorous collection and analysis of the data presented in this paper can we proceed to the next step of trying to tie specific management coping behaviors to the known risk factors. We have confined our theoretical discussions in this paper to an attempt to understand how the panelists perceive risk and how various things may have influenced their relative ranking of the risk factors.

Finally, we chose to conduct our study entirely in English, raising the possibility of varying interpretations in a multilingual sample such as ours. Although this decision did not pose a problem for our USA or HKG panelists, some of our FIN panelists had limited fluency in English and chose to respond in Finnish, necessitating translation of their responses. In spite of the aforementioned limitations, we believe that the results of the study have important implications for both research and practice.

Directions for Future Research

Although we believe the results of this study will prove useful (as they already have to some of our panelists), we believe there are many avenues for further research into IS project risk. One feature that distinguishes this study from previous research is that we have developed systematic indications of what the most important risks are from the perspective of practicing project managers. Thus the results reported here provide a useful foundation for other researchers seeking to improve our empirical understanding of software project risk factors and their variation across time and space. The list of 53 risk factors identified by the three panels (shown in Table 3) provides a useful starting point. Moreover, the 29-factor list representing the union of the ranked country lists of risk factors (shown in Table 4) provides a fruitful means of targeting normative and descriptive IS research into high-risk areas so that effective countermeasures can be developed. Questions worth asking are: "What are the countermeasures that project managers can employ against each highly ranked risk factor? Which of these are deemed most effective, and why? What interactions among risk factors and countermeasures can improve or hinder risk management efforts?" Another interesting angle for future research will be to extend this study by examining perceptions of software project risk from the vantage point of other stakeholders, such as functional area managers and senior executives. It is quite possible that different stakeholders will have divergent opinions regarding what the risk factors are, as well as their relative importance [34]. Finally, the list of risk factors identified in this study provides an excellent baseline for future researchers who wish to investigate the extent to which perceptions of software project risks change over time. Here we include both changes over time to the risk profile for a particular project as well as what project managers perceive to be important risk factors as the field of software development continues to mature.

The study also breaks new ground in providing systematic evidence that risk perception is affected by both cultural and environmental factors. Based on this evidence, we postulate that cultural elements associated with individualism, masculinity, power distance, and uncertainty avoidance—along with philosophical differences such as Taoist beliefs and the Protestant sense of guilt—may be related to the way risk items are recognized and ranked. Similarly, the frequency of some risks (like personnel turnover) may vary across economic environments, thus affecting their recognition and ranking. This clearly calls for model development and empirical studies that would seek to account for observed variation in risk perception and ranking due to cultural, environmental, and individual factors.

Another direction for future research would be to investigate systematically the links between various risk factors and their countermeasures at the behavioral and source level. The results of this study, coupled with recent theory development for risk management (see e.g. [13]), provide a good foundation for further IS research.

As we note above, we still lack theory of risk domains and associated risk behaviors. There is also a need to investigate how managers today are actually managing risks—what works, what does not, and why. Finally, we need to gain a greater congruence in the research streams of IS project risk and IS implementation. Although we touch on this issue here, it clearly warrants more in-depth study.

Acknowledgments: This project was funded in part by grants from the University Grants Council of Hong Kong, the College of Business Administration at Georgia State University, and the Academy of Finland.

Notes

1. This organization of the risk factors was not, however, used to select the risk factors for ranking or during ranking exercises. This strategy was purposefully chosen in order to avoid bias in selecting and ranking the factors. The grouping scheme was developed to organize the risk factor list in a meaningful way.

To accomplish this goal two of the four authors independently assigned our 53 risk factors into a set of common groups. Thereafter, all four authors worked together to reconcile differences in these groupings.

2. As will be discussed later, the HKG panel was an exception in this regard.

References

1. Alter, S. Implementation risk analysis. *TIMS Studies in Management Sciences, 13,* 2 (April 1979), 103–119.

2. Alter, S., and Ginzberg, M. Managing uncertainty in MIS implementation. *Sloan Management Review, 20,* 1 (Fall 1978), 23–31.

3. Barki, H.; Rivard, S.; and Talbot, J. Toward an assessment of software development risk. *Journal of Management Information Systems, 10,* 2 (Fall 1993), 203–225.

4. Boehm, B. *Software Risk Management Tutorial.* Washington, DC: IEEE Computer Society Press, 1989.

5. Boehm, B. Software risk management: principles and practices. *IEEE Software, 8,* 1 (January 1991), 32–41.

6. Boehm, B., and Ross, R. Theory W software project management: principles and examples. *IEEE Transactions on Software Engineering, 15,* 7 (July 1989), 902–916.

7. Charette, R. *Software Engineering Risk Analysis and Management.* New York: McGraw-Hill, 1989.

8. Chengalur-Smith, L., and Duchessi, P. Client-server implementation: some management pointers. *IEEE Transactions on Engineering Management, 47,* 1 (February 2000), 127–145.

9. Davis, G. Strategies for information requirements determination. *IBM Systems Journal, 21,* 1 (March 1982), 4–30.

10. Drummond, H. The politics of risk: trials and tribulations of the Taurus project. *Journal of Information Technology, 11,* 4 (December 1996), 347–357.

11. Gibbs, W.W. Software's chronic crisis. *Scientific American, 271,* 3 (September 1994), 86–95.

12. Ginzberg, M.J. A detailed look at implementation research. Working paper #753/4, MIT Sloan School of Management, 1974.

13. Griffiths, C., and Newman, M. (eds). *Journal of Information Technology,* special issue on software risk management, *11,* 4 (December 1996).

14. Heemstra, F., and Kusters, R. Dealing with risk: a practical approach. *Journal of Information Technology, 11,* 4 (December 1996), 333–346.

15. Hofstede, G. *Culture's Consequences: International Differences in Work-Related Values.* Beverly Hills: Sage, 1980.

16. Hofstede, G. *Cultures and Organizations: Software of the Mind.* London: McGraw-Hill, 1991.

17. Humphrey, W.S. *Managing the Software Process.* Reading, MA: Addison-Wesley, 1989.

18. Johnson, J. Chaos: the dollar drain of IT project failures. *Application Development Trends, 2,* 1 (January 1995), 41–47.

19. Karolak, D.W. *Software Engineering Risk Management.* Los Alamitos, CA: IEEE Computer Society Press, 1996.

20. Keen, P.G.W. *Every Manager's Guide to Information Technology.* Cambridge, MA: Harvard Business School Press, 1991.

21. Keen, P.G.W., and Scott-Morton, M.S. *Decision Support Systems: An Organizational Perspective.* Reading, MA: Addison-Wesley, 1978.

22. Keil, M. Pulling the plug: software project management and the problem of project escalation. *MIS Quarterly, 19,* 4 (December 1995), 421–447.

23. Keil, M.; Cule, P.; Lyytinen, K.; and Schmidt, R. A framework for identifying software project risks. *Communications of the ACM, 41,* 11 (November 1998), 76–83.

24. Kwon, T.H., and Zmud, R.W. Unifying the fragmented models of information systems implementation. In R.J. Boland and R.A. Hirschheim (eds.), *Critical Issues in Information Systems Research.* Chichester, UK: John Wiley & Sons, 1987, pp. 227–251.

25. Lucas, H.C. *Implementation: The Key to Successful Information Systems.* New York: Columbia University Press, 1981.

26. Lyytinen, K.L., and Hirschheim, R. Information systems failures—a survey and classification of the empirical literature. In P.I. Zorkoczy (ed.), *Oxford Surveys in Information Technology, Vol. 4.* Oxford: Oxford University Press, 1987, pp. 257–309.

27. Lyytinen, K.L.; 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), 233–255.

28. March, J., and Shapira, Z. Managerial perspectives on risk and risk taking. *Management Science, 33,* 11 (November 1987), 1404–1418.

29. McFarlan, F.W. Portfolio approach to information systems. *Harvard Business Review, 59,* 5 (September/October 1981), 142–150.

30. Moynihan, T. How experienced project managers assess risk. *IEEE Software, 14,* 3 (May/June 1997), 35–41.

31. Nidumolu, S. The effect of coordination and uncertainty on software project performance: residual performance risk as an intervening variable. *Information Systems Research, 6,* 3 (September 1995), 191–219.

32. Offenbeek, M., and Koopman, P. Scenarios for system development: matching context and strategy. *Behaviour & Information Technology, 15,* 4 (1996), 250–265.

33. Oz, E. *Management Information Systems.* Cambridge, MA: Course Technology, 1998, pp. 277–279.

34. Ropponen, J. *Software Risk Management—Foundations, Principles, and Empirical Findings.* Jyväskylä: Jyväskylä University Printing House, 1999.

35. Sauer, C. Understanding support - lessons from a case study. *Australian Journal of Information Systems, 1,* 1 (September 1993), 63–74.

36. Schmidt, R. Managing Delphi surveys using nonparametric statistical techniques. *Decision Sciences, 28,* 3 (Summer 1997), 763–774.

37. Schultz, R.L., and Slevin, D.P. Introduction: the implementation problem. In R. Doktor, R.L. Schultz, and R.P. Slevin (eds.), *The Implementation of Management Science, Volume 13.* Amsterdam: North-Holland, 1979, pp. 1–15.

38. Seigel, S., and Castellan, N. *Nonparametric Statistics for the Behavioral Sciences.* New York: McGraw-Hill, 1988.

39. van Genuchten, M. Why is software late? An empirical study of the reason for delay in software development. *IEEE Transactions on Software Engineering, 17,* 6 (June 1991), 582–590.

40. Walkerden, F., and Jeffery, R. Software cost estimation: a review of models, processes, and practice. *Advances in Computers, Vol. 44.* San Diego: Academic Press, 1997, pp. 62–94.

41. Zmud, R.W. Individual differences and MIS success: a review of the empirical literature. *Management Science, 25,* 10 (October 1979), 966–979.

Appendix 1. Risk Item Ranking by Country and by Round

Table A1.1. Ranking Results, Round-by-Round: Hong Kong Panel (9 Panelists, 15 Items)

	Mean Ranks		
Risk Factor	Round 1	Round 2	Round 3
1. Lack of top management commitment	3.89	1.56	1.33
2. Lack of adequate user involvement	5.44	4.44	4.11
3. Failure to gain user commitment	6.22	4.56	5.56
4. Lack of cooperation from users	7.33	5.22	6.00
5. Change in ownership or senior management	6.78	5.00	6.33
6. Changing scope/objectives	8.00	7.44	6.33
7. Misunderstanding the requirements	6.78	7.33	7.00
8. Lack of frozen requirements	7.56	8.11	8.11
9. Failure to manage end user expectations	8.67	8.89	8.89
10. Conflict between user departments	10.00	9.56	10.67
11. Staffing volatility	9.22	11.67	9.89
12. Introduction of new technology	9.33	10.44	10.89
13. Lack of required knowledge/skills in project personnel	9.78	11.44	11.00
14. Lack of effective development process/methodology	9.89	11.56	11.78
15. Insufficient/inappropriate staffing	10.44	12.33	13.33
Kendall's W	0.192	0.539	0.511

Table A1.2. Ranking Results, Round-by-Round: Finnish Panel (13 Panelists, 23 Items)

	Mean Ranks			
Risk Factor	Round 1	Round 2	Round 3	Round 4
1. Lack of effective project management skills	6.00	2.92	3.00	2.46
2. Lack of top management commitment	7.46	5.54	5.23	4.46
3. Lack of required skills in project personnel	11.23	8.15	5.62	5.53
4. Not managing change properly	10.31	6.38	5.85	5.85
5. No planning or inadequate planning	12.92	11.15	9.85	8.38
6. Misunderstanding the requirements	10.92	10.38	10.00	8.62
7. Artificial deadlines	9.85	8.69	9.92	9.00
8. Failure to gain user commitment	11.54	11.15	8.85	9.23
9. Lack of frozen requirements	8.46	8.62	8.15	9.38
10. Lack of “people skills” in project leadership	10.62	11.15	10.92	11.15
11. Lack of adequate user involvement	11.00	11.62	12.77	11.92
12. Multi-vendor projects complicate dependencies	10.54	10.15	10.62	12.00
13. Introduction of new technology	10.85	11.46	11.92	12.62
14. Trying new development method/technology	12.15	12.46	11.46	12.85
15. Insufficient/inappropriate staffing	13.00	12.77	14.38	14.46
16. New and/or unfamiliar subject matter	15.38	16.23	17.38	16.15
17. Staffing volatility	13.92	14.92	17.23	16.23
18. Bad estimation	13.08	13.46	15.69	16.38
19. Changing scope/objectives	13.92	16.38	15.54	16.54
20. Improper definition of roles and responsibilities	15.23	17.46	15.62	17.31
21. Poor or nonexistent control	16.31	17.46	18.31	17.54
22. Conflict between user departments	13.85	16.08	17.23	18.15
23. Failure to manage end user expectations	17.23	20.38	19.54	19.69
Kendall’s W	0.178	0.392	0.467	0.508

Table A1.3. Ranking Results, Round-by-Round: USA Panel (19 Panelists, 17 Items)

	Mean Ranks		
Risk Factor	Round 1	Round 2	Round 3
1. Lack of top management commitment	2.79	1.74	1.37
2. Misunderstanding the requirements	5.58	3.63	2.63
3. Not managing change properly	6.84	4.58	4.58
4. Failure to gain user commitment	7.37	6.42	5.58
5. Lack of effective project management skills	6.84	5.79	5.63
6. Lack of adequate user involvement	7.53	6.68	7.16
7. Failure to manage end user expectations	8.00	8.16	7.21
8. Lack of effective project management	7.37	7.47	8.11
9. Unclear/misunderstood scope/objectives	9.00	9.84	8.16
10. Changing scope/objectives	8.53	9.58	8.95
11. Lack of required knowledge/skills	9.74	10.21	11.11
12. Introduction of new technology	10.53	11.05	12.05
13. Insufficient/inappropriate staffing	11.42	12.00	12.32
14. Lack of frozen requirements	11.95	12.63	13.53
15. Improper roles and responsibilities	11.79	12.26	13.63
16. Conflict between user departments	13.21	14.37	15.37
17. Number of organizational units involved	14.79	16.11	15.58
Kendall's W	0.355	0.601	0.735

Appendix 2. Grouping of Risk Items

One result of the study was to develop a set of risk factor groups. This set could be used in the future in theory development and for more detailed prescriptions for risk management. Barki et al. [3] used factor analysis to establish five risk groups based on a set of 23 of the risk factors derived from the literature. In their interpretation they attribute each of the five groups to a particular source of risk. Analysis of our risk factor list (Table 3) shows that it covers all five of these risk groups. The Barki et al. groups of "technological newness" and "organizational environment" have direct analogs in our groups of "technology" and "corporate environment," respectively. Their "application size" group is covered by our "scope" group. Factors relating to Barki et al.'s "expertise" are broken down into project personnel expertise ("personnel" and "staffing"), and project management expertise ("project management"). Finally, we treat the issue of "application complexity" as described by Barki et al. under our groups of "scope" and "requirements." Six of our groups are unique in that they are not in the Barki et al. list. Two of them (i.e., project scheduling, and external dependencies) are covered in Boehm's top ten risk factors list. The others, focusing on relationship management (sponsorship/ownership, funding), and process planning and control (development process/planning) have been understudied by information systems researchers.

Table A2.1. Risk Groups by Source of Risk

Risk Group	Source of Risk, Nature of Risk
1. Corporate Environment	Environment: Changes in the business or political environment or poor alignment of the system with the organizational culture.
2. Sponsorship/Ownership	Mandate: Lack of mandate for the PM to execute the project plan. Lack of trust or poor relationships with the owners of the system.
3. Relationship Management	User Relationships: Lack of trust and inadequate user involvement. Unclear roles and expectations among users or other stakeholders.
4. Project Management	Management: Poor or inefficient management strategy and execution.
5. Scope	System Scope: Unclear, changing, or partial understanding of the system scope and mission.
6. Requirements	Requirements: Inadequate or poor management of system requirements; poor validation of system requirements.
7. Funding	Resource management: Too little or badly estimated resources for SD.
8. Scheduling	Resource control: Poor management of resource consumption and needs. Poor timing.
9. Development Process	Process: Inappropriate or lacking process approach.
10. Personnel	Skills: Inadequate personnel skills in development and process management.
11. Staffing	Staffing: Changes in personnel or staffing levels, unavailability of key personnel resources.
12. Technology	Technology: Inadequate understanding of the chosen technology.
13. External Dependencies	Development environment: Poor management or control over dependencies with external agents.
14. Planning	Planning: No interest or inadequate skills to plan the project.

Appendix 3. Comparison with Prior Lists

Table A3.1

Factors from prior literature (Items 1–10 are Boehm's "Top Ten" [4])	Factors from this study[a]	Composite Rank in this study[b]
1. Personnel shortfalls [4]	11.1	10
2. Unrealistic schedules and budgets [4, 30]	8.1, 7.3, 7.4	–
3. Developing the wrong functions and properties [4, 30]	5.4, 6.2	2
4. Developing the wrong user interface [4]	None	–
5. Gold-plating [4]	5.4, 6.1	6
6. Continuing stream of requirements changes [4]	6.1	6
7. Shortfalls in externally furnished components [3, 4]	13.2	–
8. Shortfalls in externally performed tasks [3, 4]	13.1	–
9. Real-time performance shortfalls [4]	None	–
10. Straining computer-science capabilities [4]	12.1	8
11. Size [3, 30]	5.3, 5.5	7
12. Multiple implementers [3, 30]	4.4, 13.2	–
13. Staffing level/team size [3]	11.1	10
14. New technology/experience with technology [3, 30]	12.1	8
15. Application novelty (ease of solution) [3, 30]	6.3	–
16. Lack or loss of resources [3]	7.1, 7.2	–
17. Unclear task/specifications [3, 30]	5.1	–
18. Team turnover [3]	11.2	–
19. Team members have not worked together before [3]	None	–
20. Team experience/knowledge [3, 30]	3.7, 10.1	5
21. Number of users [3, 30]	5.5	–
22. User turnover [3]	3.7	–
23. Number of user departments [3, 30]	5.5	
24. Type of users [3]	None	
25. Unwilling users [3, 30]	3.3	–
26. Resistance to change [3]	3.3	–
27. Users' feeling of responsibility [3]	2.2	2
28. Conflicting preferences [3]	2.4	11
29. Interpersonal conflicts [3]	10.3	–
30. Lack of top management support [3, 30]	2.1	1
31. Source of control over project [30]	13.2	–
32. Stability of client's business environment [30]	1.1	–
33. Developer's knowledge of the business [30]	10.1	–

[a] Numbers refer to items in Table 3.

[b] Ranks of those items that were selected for ranking by all three panels.