nizational entities, whether it is commercial, educational or governmental. There is a dominant voice in the discourses on the web concerning what prevention information should be provided and what populations should be targeted.

Jing Chong *is a PhD candidate in the College of Information Sciences and Technology, The Penn State University. Her research areas are health communication on websites and digital divide. The length of her research program is 4 years on a full-time basis. The current stage of her thesis research is in literature review and data collection. Her expected date of completion of the Ph.D program is August, 2008*

Virtual Project Risk vs. Traditional Project Risk in Software Projects

April H. Reed, DePaul University, School of Computer Science, Telecommunications and Information Systems, Chicago, IL, USA;
E-mail: areed7@cti.depaul.edu

INTRODUCTION

This research is designed to distinguish virtual and traditional project risks and specifically to identify the critical risks in virtual software projects. The resulting list of critical risks will provide guidance on managing risks for project leaders working in a virtual team environment. Three areas are important to the discussion of virtual software project risk: virtual software project teams, project failure and project risk.

Virtual software project teams are a growing phenomenon. They are called virtual because team members are not co-located. "Virtual teams are groups of geographically, organizationally and/or time dispersed workers brought together by information and telecommunication technologies to accomplish one or more organizational tasks (Powell *et al.*, 2004)". Many driving forces are causing increased dependence on these teams, including offshoring, outsourcing, reduced business travel due to security concerns, and improvements in collaborative tools. A task force study on globalization and offshoring indicated information technology (IT) has essentially become "a global field, business, and industry"(ACM, 2006). Therefore, there is a need to study how the unique aspects of virtual projects relate to their success.

The Standish Group over the years has measured the incidence of software project failure in corporations with their well known CHAOS reports. They conduct surveys with industry practitioners in the United States and Europe. The 2000 CHAOS report indicated 23% of projects failed while 49% were challenged (Standish Group International, 2001). The 2004 CHAOS report for the third quarter indicated 18% of projects failed and 53% of projects were challenged (Standish Group International, 2004). A one percent decrease in the number of troubled projects shows little improvement over a four year period. These numbers reinforce the need to investigate causes of project failure and identify the most critical project risks. This need is not just for projects in general, but particularly for the rapidly growing case of virtual projects.

Critical project risks are those factors that will have the greatest impact on the success or failure of a project. Boehm indicates critical risks should be the main focus of a project manager, instead of the entire pool of identified risks (Boehm, 1991). Some risk management advocates are proponents of identifying and analyzing "threats to success", which allows appropriate actions to be taken to "reduce the chance of failure" (Wallace et al., 2004).

RESEARCH PROBLEM

The research problem is:

What are the significant differences between risks in virtual and traditional software development projects? Which of these risks are critical to successful project completion of virtual software projects?

Prior research on project risk has been performed predominately on different types of traditional software projects. A review of prior research, a series of interviews and focus groups, and my own experiences as a professional project manager, have led me to the following hypotheses:

- Some risk factors in virtual and traditional software projects are similar
- Critical risk factors in virtual software projects are different from critical risk factors in traditional software projects.
- Resource issues are critical to virtual software projects.
  Reasoning: Resource issues have an impact on project success (Beise, 2004; Fairley, 1994; PMI, 2004) and are likely to be intensified when resources are not co-located
- Communication issues are critical on virtual software projects
  Reasoning: Virtual projects are often dependent on other, less common, forms of communication because traditional face-to-face communication is usually not an option. (Igbaria *et al.*, 1999)
- Virtual team dynamics are different from traditional team dynamics
  Reasoning: Virtual teams are more likely to be diversified since team members are not co-located and they may never meet face-to-face (Ewusi-Mensah, 2003; Powell et al., 2004)

OBJECTIVES

The main objective of this study is to identify a set of comprehensive yet concise *critical* risk factors for virtual software projects, by conducting a survey of industry practitioners. The list is essential to developing effective risk management for virtual software projects. Risk management, "a collection of methods aimed at minimizing or reducing the effects of project failure" (Addison & Vallabh, 2002), can then be used by organizations to increase the likelihood of project success. Thus, the importance of this study lies in its ability to enable virtual software project managers to avoid major risks and achieve greater rates of project success. The 2004 CHAOS report indicated companies in the United States and Europe spent $255 billion on software projects while the cost of failed projects was $55 billion (SoftwareMag.com, 2004).

LITERATURE REVIEW

Several researchers have identified the important risks in traditional software projects. Boehm, conducted a survey of experienced IT project managers who worked with him at TRW in the early 1990's. The result was his "Top ten software risk items" which is contained in the first column in Table 1(Boehm, 1991). One of the limitations of this study was the sample which consisted of a small number of project managers from the same company.

Barki, Rivard et al. sent their 144 item questionnaire to the largest 100 companies across a variety of industries in Quebec and surveyed 120 software development projects. The result was a list of software project risks grouped in five categories of risk factors/risk dimensions: technological newness, application size, lack of expertise, application complexity and organizational environment. The purpose of the survey was to improve management of software development projects by measuring their risk (Barki *et al.*, 1993). Barki's results are shown in the second column of Table 1.

Wallace, in her dissertation, conducted interviews with software project managers to identify risks and mass distribution of a survey to the Project Management Institute Information Systems Special Interest Group (ISSIG). The result was six risk categories or dimensions: team, organizational environment, requirements, planning and control, user, and project complexity. The purpose of her study was to improve risk management by determining the specific types of risks encountered on different types of software projects. One of the limitations of this study was the sample consisting of members from one group which were very likely to have knowledge of standard risk management practices (Wallace, 1999). Wallace's results are shown in the third column of Table 1. Thus, the three seminal studies conducted to date on the subject of project risk factors have yielded overlapping and not entirely consistent results, as shown in Table 1.

METHODOLOGY

First, a literature review of seminal work was conducted to create an initial list of project risk factors. A survey instrument was created and approved by the university Internal Review Board (IRB), then tested in face-to-face interviews with project managers, to add risks factors from a practitioner's point of view to the list. This was followed by an electronic focus group session to validate and enrich the existing risk factor list. A large volume of rich data was collected then sorted and categorized several times. This data will be compared to the seminal

Table 1

Boehm, Barry (Boehm, 1991) *Risk Factors*	Barki, Henri(Barki et al., 1993) *Uncertainty Factors*	Wallace, Linda and Keil, Mark(Wallace & Keil, 2004) *Risk Dimensions*
Personnel Shortfalls • Personnel availability • Mix of software disciplines represented • Team' expertise • Management's approach	**Lack of Expertise** • Lack of team general expertise • Lack of development expertise in team • Team's lack of expertise with task • Team's lack of expertise with appl • Lack of user experience & support	**Team** • Team member turnover • Staffing buildup • Insufficient knowledge among team members • Cooperation • Motivation • Team communication issues
	Organizational Environment • Extent of changes • Intensity of conflicts • Lack of clarity of role definitions • Resource insufficiency • Task complexity	**Organizational Environment** • Organizational politics • Stability of organizational environment • Organizational support for a project
Developing the wrong functions & properties • Size & complexity of requirements • Level of hardware imposed constraints • Number of system interdependencies • New technology or application • Requirements stability **Gold-plating** • Lack of quality in requirements & product **Continuing stream of requirements changes** • Changing requirements		**Requirements** • Uncertainty surrounding system requirements • Changing requirements • Incorrect, unclear, inadequate, ambiguous or unusable requirements
Unrealistic schedules & budgets • Unrealistic schedules • Unrealistic budgets		**Planning & Control** • Unrealistic schedules • Unrealistic budges • Lack of visible milestones to assess production of intended deliverables • Inaccurate estimates leading to inaccurate resource forecast
Developing the wrong user interface • Poor quality user interface		**User** • Lack of user involvement in development • Unfavorable user attitude toward project
Straining computer-science capabilities • Complex technology	**Technological Newness** • Need for new software • Number of software suppliers • Need for new hardware • Number of hardware suppliers • Number of users outside organization **Application Complexity** • Number of links to future systems • Number of links to existing systems • Technical complexity	**Complexity** • New technology • Complex processes being automated • Large number of links to existing systems • Large number of links to external entities
	Application size • Team diversity • Number of people on team • Number of users in organization • Relative project size • Number of hierarchical levels occupied by users	
Shortfalls in externally performed tasks • Poor quality work from external resources **Real-time performance shortfalls** • Lack of performance quality **Shortfalls in externally furnished components** • Lack of quality in external components		

literature to create a comprehensive yet concise list of risk factors for ranking in the questionnaire: A purchased mailing list of 5,000 names from a project management magazine will provide for mass distribution of the questionnaire to IT project leaders, managers and analysts.

OUTCOMES AND EXPECTED SIGNIFICANCE

The occurrence of virtual software projects will continue to increase as society becomes more global. Virtual software projects will not escape risk; therefore, project failures will occur. Researching and identifying those risk factors most critical to virtual software projects can improve risk management in this new arena. This research seeks to create a validated list of critical risks for virtual software projects that can be used by project leaders to reduce or eliminate risks.

REFERENCES

ACM, A. J. M. T. F. (2006). *Globalization and offshoring of software, executive summary and findings* (No. ACM 0001-0782/06/0200): Association for Computing Machinery (ACM).

Addison, T., & Vallabh, S. (2002). *Controlling software project risks - an empirical study of methods used by project managers.* Paper presented at the Proceedings of SAICSIT 2002.

Barki, H., Rivard, S., & Talbot, J. (1993). Towards an assessment of software development risk. *Journal of Management Information Systems, 10*(2), 203-225.

Beise, C. M. (2004, April 22-24, 2004). *It project management and virtual teams.* Paper presented at the ACM SIGMIS 2004, Tucson, AZ.

Boehm, B. W. (1991). Software risk management: Principles and practices. *IEEE Software*(January 1991), 32-41.

Ewusi-Mensah, K. (2003). *Software development failures: Anatomy of abandoned projects*. Cambridge: The MIT Press.

Fairley, R. (1994). Risk management for software projects. *IEEE Software, 11*, 57-67.

Igbaria, M., Shayo, C., & Olfman, L. (1999). *On becoming virtual: The driving forces and arrangements.* Paper presented at the SIGCPR 1999, New Orleans, LA.

PMI, P. M. I. (2004). *A guide to the project management body of knowledge (pmbok guide)* (Third ed.). Newtown Square, PA: Project Management Institute.

Powell, A., Piccoli, G., & Ives, B. (2004). Virtual teams: A review of current literature and directions for future research. *The DATA BASE for Advances in Information Systems, 35*(1), 6-33.

SoftwareMag.com. (2004). Standish: Project success rates improved over 10 years. *Software Mag.com* Retrieved 10/3/06, 2006, from http://www.softwaremag.com/L.cfm?Doc=newsletter/2004-01-15/Standish

Standish Group International, I. (2001). *Extreme chaos*.

Standish Group International, I. (2004). *Chaos demographics - 2004 third quarter research report*.

Wallace, L. (1999). *The development of an instrument to measure software project risk.* Georgia State University College of Business Administration, Atlanta, GA.

Wallace, L., & Keil, M. (2004). Software project risks and their effect on outcomes. *Communications of the ACM, 47*(4), 68-73.

Wallace, L., Keil, M., & Rai, A. (2004). Understanding software project risk: A cluster analysis. *Information & Management, 42*(1), 115-125.

How are the Impacts of End-User Application Development Managed? A Case Study of End-Users and Their Managers

Sandra Barker, School of Management, University of South Australia, Australia; E-mail: sandra.barker@unisa.edu.au

PROBLEM STATEMENT

End User Computing (EUC) has now become prolific throughout business due to the decreased cost of the available PCs and the introduction of "easy-to-use" software application generators. The most often used definition of EUC is one which incorporates the facts that end user computing involves the interaction of managers, professionals and operational level users with application software within their own working departments (Torkzadeh & Doll, 1993).

The research undertaken over the past 25 years has been particularly in the areas of end-user satisfaction with information systems, end-user computing in general, end user application development, and the identification of who end-users are and the organisational areas which are affected by end-users (Rockart & Flannery, 1983; Brancheau & Brown, 1993; Powell & Moore, 2002) together with issues that impact on end-user development but little regarding how this can be addressed in the current technological environment.

Over this period there has been a significant change in the available technology (hardware and software), the introduction of technology into pre-tertiary education and a change in the information technology culture within organisations (Rockart & Flannery, 1983; Brancheau & Brown, 1993; McBride & Wood-Harper, 2002). The review of literature has shown that there is a need for continued research into these areas.

The research question to be investigated is: *How are the impacts of end-user application development managed?* The research to be undertaken will explore the changes in technology, use of technology and its impact on organisations. The specific questions that will be addressed are:

1. What are the impacts of end user application development on:
   - the end users;
   - the managers?
2. How are these impacts managed by
   - the end users;
   - the managers?

LITERATURE SUPPORTING THE RESEARCH

In the 1970s computing was identified with mainframe computers however the introduction of PCs in the early 1980s lead to EUC being reported as '…a rapidly growing and irreversible phenomenon' (Alavi & Weiss, 1985, p6). Rockart and Flannery (1983, p777-778) identified six classifications of end-users dependent upon their function within the organisation. These classifications were:

- Non-programming end-user
- Command level end-user
- End-user programmers

Related Content