Improving Project Management Through Collaboration-Aware Risk Management Practices

Mojgan Mohtashami, NJIT, Newark, NJ, USA; E-mail: mojgan@adm.njit.edu
Vassilka Kirova, Lucent Technology, Murray Hill, NJ, USA; E-mail: kirova@lucent.com
Thomas Marlowe, Seton Hall University, South Orange, NJ, USA; E-mail: marlowto@shu.edu
Fadi P. Deek, NJIT, Newark, NJ, USA; E-mail: fadi.deek@njit.edu

ABSTRACT

Collaborative software development spanning national, language and cultural boundaries, raises new challenges and risks, which interfere with the success of software projects even when all traditional risk factors are being controlled. Software project management for such collaborative projects must address these concerns. In this paper, we outline a collaborative risk management approach and recommend policies and management processes for its support.

Keywords: Collaborative risk management, Software project management, Collaborative Risk Management Plan (CRMP), Risk Monitoring, Mitigation and Management (RMMM), inter-organizational collaboration, management contingency processes.

1. INTRODUCTION

The globalization of markets, business relationships and technology has given rise to an increasing number of less centralized, collaborative efforts and partnerships for inter-organizational software development. Collaborative relationships, including third party development, outsourcing, off-shoring, and peer-to-peer alliances, require changes to management, technical and support activities, processes, and policies. In particular the following should be considered: (1) appropriate modifications of internal organizational practices to support collaborative relationships and communication, (2) enhancements and integration of project management activities including risk monitoring, mitigation and management (RMMM), and (3) creation or adaptation of management policies and processes to guide and support the above activities.

The complexity of collaborative software development and the need for extensive cooperation and clear technical and management communication intensify the demand for steady flow of information, enhanced handling of risk, and better coordination among partners. These, together with the need to coordinate management policies between participating organizations, implies that collaborative software project management requires more flexibility, yet simultaneously more thorough process and project monitoring (Mohtashami 2006).

Risk management is an essential project management activity. It deals with anticipating, preventing, and mitigating problems related to software products, projects or processes, including difficulties in personnel, communication, and coordination. Traditional risk management has been defined and practiced with varying success in the context of a single organization and its relationships with clients and subcontractors. Risk management in a collaborative setting becomes more complex and critical as issues such as differences in organizational cultures and goals, software development and documentation practices, intellectual property, security, and management conflicts, become more prominent. Detection and mitigation are also complicated by the distributed nature of the problems, and often by the lack of a central business authority[4].

The problems faced in "traditional" project management settings[5] and these additional Collaborative Software Development (CSD) challenges necessitate a broader and more comprehensive approach to project management in general and to risk management in particular. In response, this paper outlines a collaborative risk management framework in which early and ongoing planning, clear policies, and mature collaboration-aware management contingency processes play a critical role.

The rest of this paper is organized as follows. Section 2 briefly outlines critical attributes of collaborative software development. Section 3 discusses risk management in traditional and collaborative software development. Section 4 highlights new principles for collaborative risk management, and Section 5 offers a layered framework for effective risk management in CSD derived from the attributes of CSD and those principles. Section 6 then introduces important aspects of management contingency processes required to support the framework. Finally, Section 7 briefly discusses related work, and Section 8 provides conclusions and suggests future directions.

2. COLLABORATIVE SOFTWARE DEVELOPMENT

Collaborative software development is of ever increasing importance due to the globalization of business, markets and enterprises, cost reduction and expertise utilization efforts, and more. Some significant differences between traditional software development and CSD are highlighted in Table 1[6]. CSD projects are typically characterized with no clear central authority, multiple teams, locations, and management structures, often crossing national, cultural, and language boundaries. A CSD project requires a common product vision and architecture, extensive idea and knowledge exchange, continuous communication, and active use of consultation, approval and consensus [(Higuera 1994), (Gorton, Motwani 1996), (Crampton 2002), (Niederman, Beise 1999)], although limited by intellectual property, privacy, and security considerations. This in turn mandates a detailed early analysis of technological, business and social issues and early planning for risk management. To accommodate all of the above CSD initiatives have to establish and exploit extensive channels of communication and an integrated operational environment.

While not all collaborative development projects will exhibit all of the characteristics displayed in Table 1, each will share some of these attributes and challenges, and can benefit from the proposed risk management framework, suitably adapted to individual project needs.

3. RISK MANAGEMENT IN TRADITIONAL AND COLLABORATIVE SOFTWARE DEVELOPMENT ENVIRONMENTS

Traditional software engineering practices were developed to support project teams formed by functional sub-teams or cross-functional teams operating under a single business authority. Traditional risk management is defined and used in this context.

Major risk management activities include identification and categorization of risk types, and planning for how to avoid risks where possible, and otherwise how

Table 1. Differences between traditional SD and inter-organizational collaborative SD

Perspective	Traditional SD	Inter-organizational Collaborative SD
Organizational culture		
Stakeholders	Stakeholders standard & well-known	Heterogeneous stakeholders with varying roles
Organizational culture and business practices	Homogeneous organizational culture Single set of business practices	Diverse organizational cultures Multiple sets of business practices
Organizational goal	Single organizational goal	Differing organizational goals
Peer support	Internal support and corporate loyalty	Power struggle among participants, Possible lack of support by some individuals or teams
Trust and awareness	Higher degree of trust More sure of procedures and people	Lower level of trust Higher degree of uncertainty
Management		
Management cohesiveness	Unified management	Autonomous organizations, distributed management Multiple management models
Management structure	Clear management hierarchy	No clear central authority
Communication structure	Management communication follows established business practices	Communication between peers across new channels Requires high level of cooperation & communication
Technical platform & development team		
SD practices	One set of SD methodologies	Multiple heterogeneous systems, SD standards, tools and libraries
Technological Communication	Technical communication across established and trusted channels	High volume of technical communication across new & untrusted channels
SD/technical Resources	Single set of resources	One set of resources per partner plus shared resources
Risk Management plan	Single risk management plan with clear management	Multiple risk management plans No central authority for risk mitigation
Work Practices	Known set of organizational and professional standards	Some variation in organizational and professional standards
Social and cultural issues		
Social Culture	Uniform and known social practices, norms and standards	Heterogeneous social practices, norms, and standards partly unknown to other partners
Language and idiom	A single language and idiom	Multiple languages and idioms
Cultural related work norms	Uniform cultural standard for work performance	Differing standards for work performance

to detect, mitigate and recover from problems as they occur. The key risk management functions are outlined in Table 2 below. The last two functions—Risk Communication and RMMM Review—are Supporting Activities integrated throughout the process.

Most of the activities in Table 2 are broadly discussed and applicable to more traditional as well as to CSD projects. Risk Identification and Analysis function warrants some further discussion as its scope and complexity change in CSD. Traditional risk analysis categorizes each risk along several dimensions, using historical data, industry experience, and organizational theory [(Barki, Rivard, Talbot 2001), (Boehm 1989), (Nidumolu 1996)]. The dimensions reflect the origin of the risk (nature and cause), the definiteness (from near-certain to highly unlikely), the anticipated consequences (degree of risk), and the aspects affected. Following the literature (Pressman 2005) and based on our studies (Mohtashami 2006), risk dimensions have been grouped and summarized as shown in Table 3. The last group–Collaborative Impact–is specific to collaborative software development, and is discussed in more detail below.

Collaborative software development entails a comprehensive change in the software engineering practices, from business case and product vision through development processes to management policies. Cooperation and communication issues are significantly different, both in level and kind. CSD depends on shared understanding of product vision, architecture, and implementation strategy, and hence requires extensive and continues exchange of ideas, design decisions and change information. The distributed nature of CSD teams and the diversity of work practices, cultures and regulations, skills and training levels make cooperation and communication more critical yet simultaneously more challenging. Risk management in such an environment must therefore address these new challenges. Detection and mitigation are complicated by the distributed nature of problems, and by the lack of central authority. More importantly, risk management must guard ongoing relationships between partners, rather than just the success of a single product or project.

This overarching concern for collaboration and ongoing relationships led to the introduction of the two new risk dimensions (and a new dimension group)—the degree to which the effects of a risk extend beyond organizational/team boundaries (contagion), and the degree to which the ongoing relationship between partners may be affected (trust). For our extended classification structure see Table 3.

Collaborative software development affects all aspects of risk management, and introduces changes to the traditional dimensions of Table 3, including:

Table 2. Risk management functions

Risk Management Phases	**Risk Identification and Analysis**	Elicit, identify, and classify (as below) major project and process risks. Process risk data into decision-making information. Determine the values of impact, likelihood, and timeframe [SEI/team].
	Risk Planning	Translate risk information into decisions and actions (both present and future) and implement those actions.
	Risk Avoidance	Where possible, modify to minimize likelihood/impact of particular risk type.
	Risk Monitoring	Track risk indicators and mitigation actions. Anticipate increasing likelihood of particular risks. Detect (impending/actual) occurrence where possible.
	Risk Mitigation	If a problem occurs, take steps to limit its scope and impact. In particular, try to prevent cascade of related problems.
	Risk Management, Recovery and Control	Once problem has occurred, take steps to get project/ product back on track. Correct for deviations from the planned risk actions.
Support Activities	**Risk Communication**	Provide information and feedback internal and external to the project on the risk activities, current risks, and emerging risks.
	RMMM Review	Review and update risk management strategies, plans, and activities, based on current and past feedback and environmental changes.

- **Form, View, and Source:** Additional perspectives, reflecting the CSD specifics (e.g., language barriers) must be considered; also resources to mitigate associated problems must be allocated
- **Level, Impact, and Scope:** Change in likelihood and effect can be observed—certain kinds of risks and effects become more likely and significant; others less significant.
- **Source, Driver, and Type**: Risks are likely to arise at interfaces between collaborating partners, rather than entirely within a single organization, and these new problems must be identified and classified.
- **Definiteness**: An advantage in collaboration-aware risk management is that some previously predictable risks become known risks, and can be avoided. Likewise, some previously unpredictable/unknown risks become predictable and a specific strategy for addressing them can be developed (Mohtashami, et al. 2006a).

The new dimensions must also be considered carefully:

- **Contagion and risk confinement**: Many risks, even within a single organization, may have wider effects. A risk that cannot be confined to a single organization must be addressed collaboratively.

Table 3. Dimensions for classifying risk

	Dimension Group	Dimension	Key question	Categories
Classical Dimensions	Nature and cause of risk	Form	What factor is stressed?	Resource, technical, business, environmental, platform
		View	In which aspect of the process will the problem occur?	Project[1], technical/product[2], business[3]
		Source	Which activities or constraints causes the problem?	Product definition, business impact/environment, process definition, development environment, innovation, staff skills/training, legal/regulatory
	Definiteness	Definiteness	Known in advance?	Known, predictable, unpredictable, unknowable
	Degree of risk	Level	How likely to arise?	Estimated probability range
		Impact	How serious if occurs?	Negligible to significant to catastrophic
		Scope	How much affected?	Isolated component to subsystem to entire system
	Location of effects	Driver	What business aspects are most affected?	Market, performance, support environment, cost, schedule, deployment, relationships
		Type	How does it manifest?	Functional specification/ expectation Performance or other extra-functional requirements Schedule, budget, process compliance
New Dimension	Collaborative impact	Contagion	Where are the effects?	Intra-organizational, interface, global
		Trust	How is ongoing cooperation/trust affected?	Unaffected, recoverable, damaged, unrecoverable

- **Trust**: Once collaboration has deteriorated, it is difficult or impossible to restore a good working relationship. Each risk must be examined for its potential effects on existing relationships.

In addition, collaborative software development further stresses the support activities in Table 2 that is, Risk Communication and Risk Planning, and RMMM Review. Appropriate planning, risk management strategy, and well-structured comprehensive risk plans are needed to address each of the above problems effectively. A collaboration-aware risk management plan thus becomes a critical binding and facilitation tool supporting collaboration and project management activities. Monitoring, mitigation, and recovery are specialized for an individual risk or a *risk class*, depending on the probability and potential effects of that risk (Table 3). Serious risks with high likelihood receive a dedicated specialized "*risk control plan"* (SPMN 2005) for monitoring, mitigation, and control, while less likely, less catastrophic, or more generic groups of similar risks can be handled together (Mohtashami, Marlowe, Kirova, Deek 2006a).

4. PRINCIPLES OF COLLABORATIVE RISK MANAGEMENT

Successful collaboration requires collaboration-aware management, intra- and inter-organizationally, and collaboration-aware risk management, which must extend traditional risk management [(Higuera1994), (Higuera1994a)] with means for handling the specifics of collaborative software development efforts—although standard risks, and existing management policies and practices, must still be addressed and considered.

The importance of communication, both generally and as specifically related to risk management, has long been recognized: "In the continuing application of the risk management process to large software development programs, the most dramatic effect has been in opening the communication channels for dialogues within organizations relating to risk and risk management" (Higuera 1994a). Collaborative risk management extends this need across institutional boundaries, calling for new management approaches to inter-organizational information exchange and to risk management activities themselves. This has an additional benefit in establishing trust and handling cultural and language problems. (Cultural familiarity and trust are consistently identified among the top four important success factors for collaboration (Powell, Piccoli, Ives 2004).) In CSD, project management and its risk management function must be supplemented and enhanced with new communication protocols, standards, policies and strategy, in order to:

- Help with establishment and growth of trust
- Evaluate the suitability and adequacy of management and IT processes for support of technical and social aspects of inter-organizational communication
- Address organizational, social, cultural, linguistic, and legal/regulatory differences
- Manage or at least monitor disagreements over power and responsibility among organizations
- Decrease conflict and define norms
- Let information flow effortlessly, precisely, and in a timely way
- Select metrics and tools for monitoring and measuring the success of collaborative communication and risk management

These considerations have led us to extend existing risk management principles with new principles for CSD, see (Mohtashami et al. 2006a). They augment the traditional and team-based risk management principles (SEI 2005) with collaboration-aware guidelines focused on:

- Building effective collaboration through establishment of trust, cultural sensitivity, and open communication channels, and
- Institutionalization of effective management processes, including aligned management support and responsibility for risk management–building trust, shared product vision, and consensus among partners.

Collaborative risk management provides support for addressing risk in CSD, including risks triggered by CSD-specific or intensified risk drivers and sources: cultural differences (both social and organizational), the quality of trust, communication, and IT support, and difficulties with project and risk management themselves. These principles form the basis for the layered risk-management framework sketched in the next section.

5. A LAYERED APPROACH TO RISK MANAGEMENT FOR CSD

An effective Risk Management framework for CSD should be based on collaborative-risk management principles, discussed above, and provide clear definition of decisions, actions, and responsibilities related to risk management functions. A key means in implementing the framework is a collaboration-aware RM plan, which must (1) address traditional intra-organization risk identification and management in collaborating agencies, (2) handle risks identified as introduced or intensified by CSD, including single-organization risks, resulting from interfaces, communication and collaboration, (3) handle collaborative risks not well-managed intra-organizationally, (4) drive incremental modification of policies, processes, and activities as needed, and (5) support negotiation to resolve conflicts and to assign responsibilities for risk management, while still addressing traditional concerns.

In (Mohtashami, et al. 2006a), we introduced a layered structure partitioning risk classes with minimal overlap into those best monitored and managed internally, and those requiring inter-organizational, collaborative oversight. The resulting layered Collaborative Risk Management Plan (CRMP) has three components:

1. Modified and enhanced individual, intra-organizational risk management plans incorporating collaborative risks.
2. A shared inter-organizational plan to address risks likely to be missed inside individual organizations, or which affect multiple organizations in different ways, or seriously affect cooperation and collaboration.
3. Establishment of a structure for administration and management of the shared plan, and (perhaps separately) for conflict resolution, to mediate/arbitrate conflicting organizational interests.

Thus, our collaborative RM framework has three major components: modified internal plans—one per partner, a shared inter-organizational plan (with both traditional and collaboration-specific features), and a structure for mediation and resolution. This layered approach can only be successful if accompanied by effective project management practices, including a collaboration-aware RMMM Review process.

It is not sufficient, however, to simply change the risk management functions. Changes in management contingency processes and policies are necessary, to support collaborative risk management, to reduce risk, and otherwise to conduct effective collaboration-aware management and development.

6. CONTINGENCY PROCESSES FOR CSD

To support CSD and collaborative RM effectively, project management depends on management contingency processes and policies, constituting a collaboration-aware management contingency profile (MCP). The MCP, the continuous, iterative, and interactive communication, particularly across organizational boundaries, and the creation of shared knowledge and awareness are the keys to effective collaborative project management. In addition to the typical objectives of a management contingency profile [(Piccoli, Ives 2000), (Powell, et al. 2004)], a collaboration-aware MCP must guide and support management efforts in addressing the CSD challenges discussed earlier.

CSD projects require more management involvement, plus management processes that permit a higher level of dynamic behavior than may be customary (Piccoli, et al. 2000). Moreover, the nature of the development process, and the required level of formality, must be determined with care. Although high-risk complex (but non-safety-critical) projects ordinarily benefit from a more dynamic, less formal management profile, Cohen (Cohen, Levinthal 1990) and Piccoli (Piccoli, et al. 2000) suggest that management intervention may nonetheless be beneficial. Barki (Barki et al. 2001) in fact argues that formal planning is important, particularly when cost management is at least as important as system quality. Not only management profile, but also the development process must allow for some flexibility, particularly at boundaries between collaborators.

The major risks, and thus the major focus of management processes and policies, fall into four primary categories: (1) technical risks, (2) risks arising from com-

munication, trust, and culture, (3) integration and planning risks, and finally (4) risks arising from risk management itself (risk reflection).

Technical risks–tools, product and process: Technical risks arise from difficulties with development platforms and tools, challenges with process coordination and design methodology alignment, or from problems with product compatibility, functionality and dependability (performance, reliability, scalability, etc.). These risks are exacerbated by strong process interdependence or complexity, by high rates of changes in requirements or development environment, and by lack of availability or clarity of information (Barki, et al. 2001).

Problems such as poor inter-institutional planning, or lack of effective inter-institutional risk management strategy, further increase these risks. CSD is also vulnerable to management resistance to cross-institutional technical integration, or to unification of protocols and communication modes/standards, and also to poorly specified inter-institutional technical interfaces.

Inter-organizational and inter-cultural communication and trust: The key factors in this category—trust, cultural differences and miscommunication—are mutually reinforcing. Our studies (Mohtashami 2006) indicate that, with risk management and management contingency profile, these factors contribute significantly to CSD product and process success.

Toffolon (Toffolon, Dakhili 2000) emphasizes the importance of management of communication and coordination for effective collaborative software development. Social communication at both management and technical levels is also important in building familiarity, trust, and employee morale. Policies for promoting communication and trust are discussed in (Mohtashami et al. 2006a).

Cross-organizational integration and planning: Cross-organizational integration of functions, as well as platform and information, should be defined and managed from both organizational and software development perspectives. Organizational integration deals with linking resources across physical boundaries, so that scattered organizations and teams can share and exchange information seamlessly. Two important aspects are (1) sharing and broadcasting information, and (2) collective decision making, planning and scheduling. Software development integration relies on compatibility of tools and methodology. Finally, technical platform integration is important primarily insofar as it supports these activities.

In general, CSD calls for formal planning activities, management intervention, and integration [(Barki et al 2001), (Bogia, Tolone, Kaplan, de la Tribouille 1993)]. Because of the scope, impact and contagion of many risk factors, formal planning must nonetheless support dynamic management of collaborative activities. Therefore, CSD requires a project management profile that supports high levels of planning, adaptability and integration of management approaches. Formal control effectively contains the complexity of CSD, while adaptability controls unexpected variation—the challenge is remaining flexible enough to adapt to CSD's chaotic nature.

Risk reflection: The increased complexity of risk identification, classification, and management, as well as complexities introduced by cooperative decision-making for risk mitigation, may lead to additional complications. New risks can arise from bad, inappropriate, or overly constrictive RMMM plans and policies. While this paper does not address the issue further, RMMM planning, review and evolution is an essential requirement for a successful collaborative project management.

The contingency processes, required to address the above added complexity and challenges, include behavioral control management policy (rules and procedures), collaborative management policies, and management of differences in institutional practices; risk management overlaps each of these categories. These aspects of management profile are responsible for initiating or effectively controlling the following activities: management of organizational behavior, management of processes, policies for integration, policies for IT support, and policies for risk identification and management.

Management of organizational behavior deals primarily with social and technical training, and with establishment of a cooperative culture. It includes creation of inter-organizational trust, recognition of and adaptation to cultural differences, establishment of training programs, and demonstration of senior management support for the collaborative effort.

Management of processes includes establishment of a proper fit between project risk and management profile, enhancing task clarity (clear specification of global requirements, partner responsibilities, and interfaces), and establishment of communication protocols and standards, as well as tool and development process standardization.

Policies for IT support need to respond to the high uncertainty of CSD. They should address provision of tools to support and manage non-routine activities (such as event-driven group conferencing). They must deal with increased needs for security and integrity, as well as protection of privacy and intellectual property. In addition, they must support tool and process integration, and sharing of a wide variety of software and management artifacts.

Policies related to risk identification and management require definition and institutionalization of a collaborative RM framework, institutionalization of collaboration-aware risk identification and analysis, monitoring of risk arising from inter-organizational communication and interfaces, and cross-organizational sharing of risk-related information, events, and changes. These activities are essential to CSD, and are discussed in detail in (Mohtashami et al. 2006a).

7. RELATED WORK

Related work falls primarily into the following areas: risk management and software project management, distributed software development (SD) and software teams, collaborative software development and cultural differences and sensitivity.

Extended discussions of project management, traditional risk management, and RMMM planning, can be found, for example, in [(Pressman 2005), (Sommerville 2006)], in publications focused on software project management [(Tsui 2004), (Royce 1998)], or in the publications of the Software Engineering Institute (SEI 2005) and the Project Management Institutes (PMI 2006). Ranky (Ranky 2006) offers a practical introduction to collaborative project management, and Fang (Fang, Nunamaker, Romano, Briggs 2003) examines the problems associated with traditional project management approaches given business globalization and information technology advances, and highlights the benefits of collaborative project management.

Distributed software development and its risk management are addressed in [(Barki et al. 2001), (Gotterbarn 2005), (Higuera 1994), (Lee, DeLone, Espinosa 2006)]. Good analysis of virtual teams management and collaborative software development issues can be found in [(Chopra, Meindl 2001), (Beranek, Broder, Romano, Reinig 2005), (Cantu 1997)]. Selection of software process models for distributed and collaborative software development is also a subject of ongoing discussion (see for example (Ramesh, **Cao, Mohan., Xu** 2006)).

Risk management for distributed SD, particularly focused on customer-supplier relationships and team-work (Team RM) is addressed in [(Beranek, et al. 2005), (Higuera 1994), (Higuera 1994a), (SEI 2005)]. The Team RM approach emphasizes collaboration, teamwork, as well as negotiation and use of shared team risk management plans. However, it does not consider such issues as lack of a single authoritative decision-making entity or cultural differences, which are specific to CSD and are addressed in the collaborative risk management framework presented in this paper.

8. CONCLUSIONS, IMPLICATIONS AND FUTURE WORK

We have introduced an enhancement of current risk management practices and policies to handle complications arising in large, multi-organizational/multi-enterprise, collaborative software development projects.

In support of collaborative project management and collaborative RM, in particular, we have introduced the concept of a layered risk management plan as a critical means of collaborative risk management. The effective execution of such a layered plan requires appropriate management and mediation policies and processes. Understanding the role these play in CSD, and augmenting the CRMPs with effective collaboration-aware management contingency profiles, further adds the dynamic control and flexibility needed for successful CSD projects.

As a preliminary study, this work leaves many open questions, which we (and we hope others) will explore in the future. Important areas of future work include:

- Further exploration of risk classification and risk management issues for collaborative software development (Mohtashami et al. 2006a).
- Exploration of approaches to develop specific recommendations for management contingency processes and policies
- Investigation of strategies for reconciling mitigation actions taken by different authorities (e.g., in multi-company CSD projects) and the changes needed in MCPs to support these strategies

- Consideration of the impact of Software Development Lifecycle process models on the success of CSD projects and their management
- Investigation of the interactions between collaborative risk management, management contingency processes and policies, and overall software project management activities.

The range of open questions, and the difficulty in collecting solid statistics on the experience of on-going collaborative efforts, makes full validation of this proposal challenging. Nonetheless, we expect that practitioners will be able to use the preliminary results and ideas presented in this and prior papers in developing more effective, collaboration-aware risk management plans and management contingency policies for their specific efforts, resulting in improved CSD project management practices.

REFERENCES

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

Beranek, P. M., Broder, J., Romano, N., Reinig, B. (2005). Management of virtual project teams: Guidelines for team leaders. *Communications of the Association for Information Systems*, 16(10), 247−259, 2005.

Boehm, Barry W. (1989). Software Risk Management. *IEEE Computer Society Press*. Los Alamitos, CA.1989

Bogia, D. P., Tolone, W, J. Kaplan, S. M., de la Tribouille, E. (1993). Supporting Dynamic Interdependencies among Collaborative Activities, *Proceedings of the Conference on Organizational Computing Systems***.** 108-118, 1993.

Cantu, C. (1997). CSWT Papers, Virtual Teams, Centre for the Study of Work Teams, University of North Texas, Available at: http://www.workteams.unt.edu/reports/Cantu.html, 1997.

Chopra, S., Meindl, P. (2001). Supply Chain Management, Strategy, Planning and Operation, *Prentice-Hall Inc*, Upper Saddle River, New Jersey, 2001.

Cohen, G. W. M., Levinthal, D. A. (1990). *Absorptive Capacity: A New Perspective on Learning and Innovation*. Administration Science Quarterly, Vol. 35, 128 –152, 1990.

Crampton, C. D. (2002). Finding Common Ground in Dispersed Collaboration. *Organizational Dynamics*. 39(4). 356-367. 2002.

Fang C., Nunamaker, J.F., Jr., Romano, N.C., Jr., Briggs, R.O. (2003). A collaborative project management architecture, *Proceedings of the 36th Annual Hawaii International Conference on System Sciences*, 2003.

Gorton, I, Motwani, S. (1996). Issues in co-operative software engineering using globally distributed teams. *Information and Software Technology*. 38 (10). 647−656, 1996.

Gotterbarn, R. (2005). Responsible Risk Analysis for Software Development: Creating the Software Development Impact Statement, *Communications of the Association for Information Systems*, 15, 730−750, 2005.

Higuera, R. P. (1994). Introduction to Team Risk Management, *Special Report CMU/SEI-94-SR-1*, May 1994,

Higuera, R. P. (1994a). Team Risk Management: A New Model for Customer-Supplier Relationship, *Special Report CMU/SEI-94-SR-5*, July 1994, http://www.sei.cmu.edu/pub/documents/94.reports/pdf/sr05.94.pdf.

Lee, G. DeLone, W., Espinosa J. A. (2006). Ambidextrous coping strategies in globally distributed software development projects. *Communications of the ACM*. SPECIAL ISSUE: Flexible and distributed software processes: old petunias in new bowls? Volume 49, Issue 10, P: 35 - 40 ISSN:0001-0782 October 2006.

Mohtashami, M. (2006) The Antecedents and Impacts of Information Processing Effectiveness in Inter-Organizational Collaborative Software Development, PhD Thesis, Rutgers University – School of Management, June 2006.

(Mohtashami 2006a) Mohtashami, M., Marlowe, T., Kirova, V., Deek, F.P. (2006a) Risk Management for Collaborative Software Development*, Information Systems Management*, 23(4), 20-30, Fall 2006).

Niederman, F., Beise, C. M. (1999). Defining the Virtualness of Groups, Teams, and meetings. *Proceedings of the 1999 ACM SIGCPR Conference on Computer Personnel Research*. 14-18, 1999.

Nidumolu, S. R. (1996). A Comparison of the Structural Contingency and Risk-Based Perspectives on Coordination in Software-Development Projects. *Journal of Management Information Systems,* 13 (2), 77–113., 1996.

Piccoli, G., Ives, B., (2000). Virtual teams: Managerial behavior control's impact on team effectiveness, *Proceedings of the twenty-first international conference on information systems*. Brisbane, Queensland, Australia**.** 575- 580, 2000.

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)*, 2004.

(Pressman, R. S. (2005). Software Engineering: A practitioner's approach. *6th edition. McGrawHill,. ISBN 0-07-301933-X*, 2005.

PMI 2006 (2006). Project Leadership: A Practical Guide to Communication, Influence and Collaboration, http://www.pmi.org/prod/groups/public/documents/info/pdc_sw_td_projleader07.asp

Ranky, P. G. (2006). Collaborative Project Management Methods, Tools & Technologies with USA & International . Examples in Engineering, IT, Management and Service Industries, *CIM ware USA, Inc.* March 5, 2006

Ramesh B., Cao, L., Mohan, K., Xu, P. (2006). Can distributed software development be agile? *Communications of the ACM***,** SPECIAL ISSUE: Flexible and distributed software processes: old petunias in new bowls? Volume 49 (10), 41 - 46 ISSN:0001-0782, October 2006.

Royce, W. (1998). Software Project Management: A Unified Framework, *Addison-Wesley Professional*, 1998.

Sommerville, I. (2006). Software Engineering, Addison-Wesley, England, 8th Edition, 2006.

Standish (2004-2005) The Standish Group. *The Chaos report.* West Yarmouth, MA:, 2004/2005.

SPMN 2005: 16 Critical software practices for performance-based management, Software Program Managers Network, 2005. http://www.spmn.com/16CSP.html.

Toffolon, C., Dakhili, S. (2000). A framework for studying coordination process in software engineering. *Proceedings of the 2000 ACM Symposium Applied Computing*, 851– 857, March 2000.

Tsui, F. (2004). Managing Software Projects, Jones and Bartlett Publishers, Inc., March 2004.

ENDNOTES

1. Project includes process, resources (including personnel), communication, and systems analysis.
2. Product requirements and technical environment.
3. Corporate health, business case, senior management support, etc.
4. Clear project management structure does not necessarily entail authority to make process changes across organizational boundaries required to reduce or handle risk. Such changes depend on organizational decisions and require the involvement and support of business management, whose span of control is restricted by the organizational boundaries.
5. The failure rates of contemporary projects (Standish 2004-2005), indicate problems even with in-house project development despite the existence of developmental methodologies, frameworks and risk management practices. Given such problems – with time, cost, functionality and quality—development in a dispersed or virtual setting is even more challenging.
6. The comparison is based on extremes – a simple case of traditional software development is compared with a complex CSD effort – in order to identify as many potential areas of new or intensified risk as possible. There is a spectrum from "fully local" to "fully distributed/fully collaborative" efforts, and many of the factors we identify affect single-enterprise projects in the middle of this spectrum.