DISRUPTION OF INFORMATION TECHNOLOGY PROJECTS:

THE REACTIVE DECOUPLING OF

PROJECT MANAGEMENT

METHODOLOGIES

by

Kurt W Schmitz

Presented to the Faculty of the Graduate School of

The University of Texas at Arlington in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT ARLINGTON

December 2013

Acknowledgements

It is with deep appreciation that I thank my wife, Peggy, for enabling and encouraging this initiative. While my journey involved the compounding of a full time career with seminars, conferences and the various investigations that go into building a scholarly research program, she contributed innumerable hours of proof reading, grammar correction, spell checking, and management of the many obligations of family and life that I would have missed or forgotten without her. She has graciously extended the partnership of our marriage to facilitating this endeavor and helping assure its success. The words “Thank you” are too small to capture the debt I owe!

I am also grateful for the encouragement and support from my employer Alcon Laboratories, to provide resources, scheduling flexibility, access to the organization as a subject for study, and faith that this undertaking would be successful and create mutual value. Special thanks to Eduardo Blas and Matt von Wellsheim who supported my request to begin this endeavor, and championed endorsement by the company leadership. In his capacity as Alcon CIO, Matt approved multiple study ideas and assured access to the organization for data gathering that has been instrumental in the study presented here. I am similarly indebted to Remy Evard, CIO of Novartis Institute of Biomedical Research for allowing access to IT project teams in that organization for the case study. Special thanks for Dr. Richard Newton who supported my studies in his capacity as my supervisor at Alcon. Richard's involvement has been much deeper as he participated in brainstorming discussions regarding the mechanisms that influence project managers. His contribution includes his role as a member of my dissertation committee and encouragement through a thousand small acts of support during daily interactions at the office.

The above noted contributions created an environment of support and encouragement that was a necessary condition for the accomplishment this dissertation represents. In addition, the guidance of my University of Texas at Arlington professors in the College of Business provided training and directed my energies to productively absorb the body of knowledge and research traditions that form a foundation for progress in the behavioral sciences. Thank you Dr. Mary Whiteside and Dr. Kenneth Price for providing an essential foundation in behavioral sciences research and analysis techniques with a maturity of rigor, competence and collegiality that welcomed me into this domain of scholarly pursuit and solidified my interest and desire to join their ranks. Thank you Dr. James T. Teng for countless hours reviewing manuscripts, collaborating on construct development, nudging theory building and guiding a scholarly writing style. Thank you Dr. Marcus Butts for encouraging sustained quality and excellence needed for high quality research. I am honored to have had the attention and mentorship of such accomplished and recognized scholars.

Collectively your example and reminders of best practices provide a benchmark I seek to emulate.

Above all I am indebted for the hours of mentoring and challenging discussions provided by both Dr. Radha Mahapatra and Dr. Sridhar Nerur. In addition to plunging me into the deep end of Information Systems scholarship, you have provided a patient and forgiving sounding board allowing numerous unconventional ideas to develop from sometimes rebellious sparks into workable constructs and theories that have the potential to trigger new directions of research. I have benefited greatly from engaging and challenging discussions that have taken me out of my comfort zone to seek new insights from seemingly commonplace phenomenon. Thank you for mentoring my introduction to the tools your domain offers in the search for bits and pieces of underlying truth.

I am also grateful for my classmates for making seminars intellectually inspiring, for my workplace colleagues for demonstrating and sharing how the real world functions and for my children, Ashleigh and Mark, for providing balance, enthusiasm and passion for new experiences that keeps me younger than my hair color suggests.

October 9, 2013

Abstract

DISRUPTION OF INFORMATION TECHNOLOGY PROJECTS:

THE REACTIVE DECOUPLING OF

PROJECT MANAGEMENT

METHODOLOGIES

Kurt W. Schmitz, PhD

The University of Texas at Arlington, 2013

Supervising Professors: Radha Mahapatra and Sridhar Nerur

Information Technology projects have migrated toward two dominant Project Management (PM) methodologies. Plan-driven practices provide organizational control through highly structured plans, schedules, and specifications that facilitate oversight by hierarchical bureaucracies. In contrast, agile practices emphasize empowered teams using flexible methods aligned to organizational values reinforced by cultural controls and rituals. While project teams mix and combine practices from both traditions, a stylistic bias remains in the initial project management practices established at the onset of IT projects. A review of existing literature on IT project management serve as a filter on the stylistic biases of these dominant PM paradigms that is organized as a two dimensional PM Style framework. This study employs a two-phase multi-method

approach that builds on this framework to develop and test a theoretical explanation for the influence of inevitable unexpected challenges that emerge to disrupt IT projects. The first phase is a comparative case study contrasting two agile projects with two plan-driven projects. A Critical Realist method organizes the events and structures revealed in the case study with insights from existing theories of Loosely Coupled Systems to propose a new theory of Reactive Decoupling that describes the shift in PM style triggered by unexpected challenges. A confirmatory study using survey techniques collecting data from 268 professional IT project managers is analyzed to provide empirical support for the new theory. Findings support the theory's proposition that fluctuations in the form of unexpected challenges unfreeze the baseline project management style and encourage a shift toward decoupled systems behavior. This shift allows some plan-driven projects to behave as loosely coupled systems and gain the performance advantages of greater requisite variety and flexibility. Where the intensity of fluctuations is sufficiently great, the shift may drive projects out of the loosely coupled systems arena toward fully decoupled systems status, where results are dependent upon the self-control and good-will of ad hoc project team behaviors.

Table of Contents

List of Illustrations

List of Tables

Chapter 1

Introduction

Underperforming IT investments have led to an examination of risk and what organizations can do to mitigate and manage risk. Unexpected events and challenges appear to be ubiquitous, even when a priori effort is applied to anticipate and mitigate potential risks (Pavlak 2004). This study pursues the idea that a project team's response to unexpected challenges is a key determinant of success. In addition, unexpected disturbances provide opportunities for individuals, teams and organizations to innovate and change (Grote et al. 2004), making these momentary annoyances particularly important to the process of implementing new information systems which are themselves agents of change within organizations.

IT systems and capabilities are implemented within organizations by teams operating in the context of an identifiable project. In the earliest days of computers, talented pioneers had nearly unlimited flexibility in how they approached and executed their projects. Over time organizations instituted control over these projects through bureaucratic imposition of process and measurable deliverables. These methods have been variously labeled as traditional, highly-structured, and *plan-driven* (Boehm and Turner 2003). Other project methodologies emphasize informal interactions and quick response to change. Practitioners and advocates of these flexible methods self-identify with the label *agile* (Highsmith and Cockburn 2001). Many specific practices and conditions are common across both plan-driven and agile methods, yet literature and practitioners consistently recognize a distinction (Doherty 2012; Fernandez and Fernandez 2009; Meso and Jain 2006; Nerur et al. 2005). The distinction between the two methods is theoretically a juxtaposition of *bureaucratic* and *concertive* organizational controls (Barker 1993; Ouchi 1979). A visible manifestation of this distinction can be found in a planning and documentation philosophy that serves as an influential bias affecting many aspects of how projects are carried out. These practices chosen for a project can and do change. This study investigates project management methods and behaviors change in response to unexpected challenges.

1.1 Research Questions

The core question of this study is: What are the mechanisms through which project management philosophies and biases engage with unplanned challenges to affect project outcomes? Of particular interest is the role of project management style favoring either plan-driven or agile practices. A secondary question is: How do project management behaviors change in response to unexpected challenges, and what mechanisms guide that change?

1.2 Importance of Research

Interruption and unexpected events are inevitable for all but the least complex projects (Dvir and Lechler 2004; Hallgren and Maaninen-Olsson 2005; Sauer and Reich 2009). A project team's reaction to unexpected challenges is arguably the pivotal point for project success or failure. While much attention has been applied to prescriptive practices of risk management and project management, a theoretical understanding of the reaction of information technology (IT) and information systems (IS) project teams faced with disruptive events is nascent. This study seeks to provide a behavioral theory starting point to advance understanding of IT project disruptions.

1.3 Overview of the Dissertation

Chapter 2 provides a review of literature and introduces a Project Management (PM) Style Framework setting the stage for a comparative case study exploring project responses to unexpected disruptions. The subsequent Critical Realist assessment combines observations with existing ideas of boundary spanning and loosely coupled systems to develop a proposed theory of Reactive Decoupling in Chapter 3. Chapter 4 introduces specific research models, constructs and hypotheses inspired by the PM style framework and the theory of Reactive Decoupling. The first model characterizes the effects of project management style using a contingency approach. The second model seeks to illustrate Reactive Decoupling directly in terms of the dialectic properties of responsiveness and distinctiveness. Chapter 5 presents a study design to further investigate the framework and provide more broadly based empirical support for the new theory. Chapter 6 provides the results of a cross-sectional survey of professional IS project managers. Chapter 7 presents a general discussion of the study’s findings with implications for both theory and practice. Limitations of this study and suggestions for future research are also provided.

Chapter 2

Literature Review

2.1 IT Project Management Styles

The culture of IT project management has evolved over a 60-year period. This evolution has created philosophical biases rooted in the methods, practices and behaviors that reflect the choices of individuals and organizations. This literature review begins with a summary of this history instrumental in establishing the structures that guide today's IT project managers.

2.1.1 Pioneers Used Ad Hoc Code-and-Fix Methods

Early IT projects were developed by talented pioneers wielding nearly unlimited flexibility in how they approached and executed their tasks. The community was dominated by scientists and mathematicians clustered around major universities. The role models were extremely bright cowboy programmers who would work long hours and weave together software programs with a seat-of-the-pants ad hoc approach characterized as a code-and-fix method that reflected the footloose and fancy-free culture of the time (Boehm 1988, 2006; Overmyer 1990). This pioneer era started near the dawn of computer programming in the 1950s and progressed into the 1970s. During this time programmers and system developers had great autonomy to roam creatively wherever their imagination and interest led them. While this environment, sometimes characterized as anarchy or "adhocracy" (Macomber and Howell 2003; Mintzberg and

McHugh 1985; Stacey 1993), was highly innovative and satisfying for creative individuals, it also led to many project disasters where the functional requirements of users were not met, or the cost, time and effort required were not aligned with the needs of the sponsor organization (Chapman et al. 1993; Fitzgerald 2000).

2.1.2 Traditional Plan-Driven IT Project Management

Instead of depending upon the benevolence, good will and self-control of developers, organizations stepped in and imposed project management discipline in the form of bureaucratic controls (Avison and Fitzgerald 2003; Boehm 1986, 2002; Harris et al. 2009b). Engineering project management practices had existed in the form of craft for centuries. The 1950s saw the application of management science techniques and new project management tools in the form of Critical Path Method (CPM) and Program Evaluation and Review Technique (PERT), followed in the 1960s and 1970s by Earned Value Planning (EVP) and Monte Carlo simulations (Kerzner 2004; Kwak 2005; Pich et al. 2002). Practices from engineering were adopted by software projects in the form of project lifecycle models with discrete phases. Championed by large multi-organization projects, these ideas were formalized in development methodologies that prescribed careful planning, up-front specification and a step-wise progression through phases of requirements, analysis, design, coding, testing and delivery. The Waterfall model (Royce 1970) was particularly influential and served as a framework for many government mandated software engineering processes, culminating in the mid 1980s with DoD-STD-2167 in the US, along with PSS-05 and ISO-9000 in Europe.

2.1.3 Planned Flexibility: Incremental Development and Iterative Delivery

While software engineering standards continued to emphasize fully specified stepwise project management until the 1990s, incremental and iterative practices present in various forms within the software development community since the 1950s have also matured (Larman and Basili 2003). Incremental development involves breaking a large project into components that are developed in a series of repeating cycles. Early forms of incremental development called for full documentation of requirements and designs preceding a stepwise development exercise that was repeated for a gradually expanding set of requirements. The repeating nature of incremental development provides flexibility by allowing many aspects of the design to be deferred until deep into a project. Another innovation came in the form of iterative delivery whereby the system would be delivered into the user environment early in an abbreviated form. Succeeding development and implementation cycles react to evolving user feedback by adding or modifying functional capability on a periodic basis. Project methodologies were advanced based on these techniques in either a highly structured framework with full documentation for each development cycle (e.g., Spiral Development: Boehm 1986; Phased Design: Dennis et al. 1987) or as a totally unstructured development philosophy (e.g., Prototyping: Naumann and Jenkins 1982; Evolutionary Project Management: Gilb 1976). The value of incremental and iterative approaches eventually led to the removal of the linear Waterfall bias from many software engineering standards (e.g., MIL-STD-498 replaced DOD-STD-2167 in 1994).

2.1.4 Agile Paradigm Shift

In certain sectors of the software industry, the flexibility of incremental and iterative methods intersected with the highly innovative era of the Internet boom of the 1990s. Thousands of startup companies formed quick acting teams pursuing new product ideas, particularly in the areas of online marketplaces and digitized media. The competition for web traffic and consumer eyeballs assured the failure or irrelevance of firms that did not adapt quickly. In this environment, many software developers moved aggressively to cast off the bureaucratic controls prescribed by the plan-driven approaches. Some software developers returned to the cowboy programmer behaviors of the 1960s, while others advocated a value-centric “samurai programmer” approach eventually codified in the Agile Manifesto (Cullom and Cullom 2006). Formalized agile methodologies such as Extreme Programming (XP)(Beck 1999) and Scrum (Takeuchi and Nonaka 1986) prescribed a mixture of outcome based controls (e.g., time-boxed iterations, burn down, and formal acceptance testing) with ceremonies (e.g., sprint planning meetings, daily standup meetings, a planning game, retrospectives) and jargon (e.g., “refactoring”, “user stories”, “low-fidelity prototypes”) to reinforce the values of continuous informal team interaction, and quick response to changing requirements.

2.1.5 Project Management Bias

By 2001 when the Agile Manifesto was formalized, many of the practices embraced by the agile philosophy had also been adopted and codified by mainstream standards based methodologies (e.g., iteration, user involvement, delegation, empowerment, quality, continuous improvement, and learning) (Forsberg et al. 2000). Yet despite apparent agreement between practitioners and advocates of agile and plan-driven methods, there is a lingering and clear distinction between the approaches. The difference is rooted in the spirit and intensity with which practitioners identify with specific practices. Contemporary examples of plan-driven methods include the Project Management Institute's Project Management Body of Knowledge (PMBOK 2008), The IEEE Computer Society's Software Engineering Body of Knowledge - SWEBOK (Abran and Bourque 2004) and V-model (a.k.a. Vee model) (Forsberg et al. 2000; Rowen 1990) that emphasize planning and documentation. The top-down approach encourages a sequential progression through phases, repeated through iteration and supported by ongoing management activities of measuring, monitoring, controlling and reporting. At the heart of planning and orderly development is the drumbeat of formal documentation firmly entrenched in a plan-driven philosophy that believes "if it's not documented, it doesn't exist" (Fried 1992). The plan-driven community has nurtured a bias that elevates the importance of formally documented specifications and plans. The sustained nature of this bias is captured in language that is both colorful and unequivocal, paraphrased in Table 2-1.

Agile methods take a bottom-up approach that expects requirements to emerge and evolve as the development process takes place. The practices rely heavily on continuous dialog and interaction as an intentional strategy to replace or minimize formal documentation. The agile community culture establishes a bias to minimize all forms of

Table 2-1 Plan-Driven project management bias

Source	Plan-Driven Project Management Bias
Royce (1970) *[Waterfall]*	• The first rule of managing software development is ruthless enforcement of documentation requirements. (p332) • If the documentation is in serious default, my first recommendation is simple. Replace project management. Stop all activities not related to documentation. Bring the documentation up to acceptable standards. Management of software is simply impossible without a very high degree of documentation. (p332) • An acceptable written description forces the designer to take an unequivocal position and provide tangible evidence of completion. (p332)
Cooper (1984)	• To treat computer programming documentation casually is to invite disaster. (p23)
Cort, Goldstone, Nelson, Poore, Miller, & Barrus (1985)	• The methodology emphasizes the development and maintenance of comprehensive documentation. … Documentation is fundamentally more important than the code. (p1439) • Documentation is the cornerstone of any successful development methodology. (p1440) • Exhaustive documentation is progressively generated for each phase, and the coding activity is demoted to consume minimal project resources. (p1442)
Card, McGarry & Page (1987)	• Effectively document each phase of development. Documentation improves software reliability at little or no net cost. (p849)
Sole & Bist (1991)	• Although on first appearance it might seem that a design document simply adds one more thing to do on a project, the added effort is more than compensated for by the amount of time and money saved in not having to modify information at a much later point. (p73)
Fried (1992)	• If it's not documented, it doesn't exist. (p72) • It's never too early to plan. (p74)
Fowler (1999)	• Every project depends on documentation. It is the lifeblood of development. You haven't completed a project if the documentation isn't done. (p53) • Plan each document. Do it early, do it often or die. (p56)
Nelson (2003)	• Design documentation and reviews are indispensable to a project's success. (p44)
Abran & Bourque (2004) *[SWEBOK]*	• Software requirements specification permits a rigorous assessment of requirements before design can begin and reduces later redesign. (p2-8) • Different stakeholders, including representatives of the customer and developer, should review the document(s). Requirements documents are subject to the same software configuration management practices as the other deliverables of the software life cycle processes. (p2-9) • In an environment where change is an expectation rather than a shock, it is vital that plans are themselves managed. This requires that adherence to plans be systematically directed, monitored, reviewed, reported, and, where appropriate, revised. (8-5)
Nerur, Mahapatra & Mangalaraj (2005)	• Traditional methodologies also produce a large amount of documentation that codifies process and product knowledge. Communication among project participants is formalized through these documents. (p75)
Pomeroy-Huff, Cannon, Chick, Mullaney & Nichols (2009) *[PSPSM]*	• Software designs must be documented, along with the related requirements, constraints and rationale. (p55) • A design specification should be precise. The lack of a precise design is the source of many implementation errors. For best design precision, specify and document design decisions before beginning the coding step of the process. (p56)

written documentation. The consistent nature of this bias since the Agile Manifesto is captured in the literature summarized in Table 2-2.

Table 2-2 Agile project management bias

Source	Agile Project Management Bias
Muller & Tichy (2001)	• XP breaks with a number of traditional software engineering practices. First, documentation is almost entirely non-existent. (p1) • Second, there is no software specification. (p1)
Orr (2002)	• There should be minimal documentation beyond the code. (p6) • A strong minimalist bias exists. (p6)
Briand (2003)	• XP relies exclusively on oral communication, tests, and source code to communicate system structure and intent. In other words, if faithfully applied, there are no Analysis and Design documents. (p2)
Nerur, Mahapatra & Mangalraj (2005)	• Agile methods discourage documentation beyond the code. (p75)
Ratanotayanon, Kotak, Sim (2006)	• Agile software processes do not produce and maintain any high-level documents other than source code and comments … The fact that there is no explicit medium for knowledge transferring, and most important knowledge resides only within team members' heads raise an important question. (p200)
Selic (2009)	• Developers don't like to do documentation, because it has no value for them.
Ramesh, Cao, Baskerville (2010)	• Agile development assumes extensive documentation and models are counterproductive. (p451) • Formal documentation of the specifications is seldom created. (p456) • Instead of using formal requirements documents, many projects use prototyping as a way to communicate with their customers. (p460)
Stettina, Heijstek & Faegri (2012)	• We found that writing documentation was perceived as an intrusive task leading to task specialization and allocation of documentation to less qualified team members.
Strode, Huff, Hope & Link (2012)	• Sprint planning meetings, open office space, and daily meetings provide efficient communication. Used together, these practices were found to promote informal communication and substitute for documentation as a communication mechanism.
Ambler (2013)	• Unlike traditionalists who often see documentation as a risk reduction strategy, Agilists typically see documentation as a strategy which increases overall project risk. (p1) • The agile strategy is to defer the creation of all documents as late as possible.

The philosophical distance between plan-driven and agile practitioners is most extreme in the enthusiasm with which planning and documentation are either formalized or deemphasized. Plan-driven practitioners (Planners) see documentation as a risk reducing strategy (Ambler 2013; West et al. 2011); Agile practitioners (Agilists) see documentation as a strategy that increases overall project risk and prefer exploratory trial-and-error as means to mitigate the risk of building the wrong thing (Armitage 2004; West et al. 2011). Planners see the documentation of product and processes as key components of software quality (Whittaker and Voas 2002; Visconti and Cook 1993); Agilists view documentation as a constraint on mid-project or late-project flexibility necessary to build in quality. Planners contend that minimum rework is a sign of process quality that results from careful planning and design (Stephens 2003); Agilists view continuous refactoring as an accepted way of improving design quality after initial product delivery (Fowler and Beck 1999). Planners believe their methods lead to lower overall cost and timely delivery of projects (Germain and Robillard, 2005); Agilists contend that relying on interactions compensates for minimizing documentation to produce more appropriate results and less expensive designs (Highsmith and Cockburn 2001). Planners see documentation and structure as a means to promote learning (Bunderson and Boumgarden 2010); Agilists believe the ritual of project retrospectives conducted face-to-face facilitate individual growth and learning (Highsmith 2000). The dialog and language of advocates has created a group identification boundary around the project management philosophy that is reified in the planning and specification practices for Planners, and autonomy and iterative refactoring for Agilists. This distinction is aligned with the organizational control mechanisms which underlie each philosophy.

2.2 Organizational Control

2.2.1 Controls and Organizational Theory

Organizational theorists suggest that individuals left to their own devices pursue private interests that diverge from those of the firm (Cyert and March 1992). A variety of mechanisms are available to managers in order to guide and encourage independent actors to support, promote and advance the interests of the organization. Control Theory (Ouchi 1979) identifies three mechanisms: market transactions, bureaucratic control and clan control. Market mechanisms involve arms length transactions based on competitive bids to perform defined services for a fair price. The organization receives a specified product or service from the independent actor in exchange for payment. Such transactions involve friction and inefficiency in the form of formally negotiated contracts. Some activities are more effectively performed by actors within the firm, where the cost of contracts is removed and work processes activities are under the control and discretion of the organization.

Bureaucratic controls establish rules for behavior and standards for output and quality with which to measure and monitor actual performance. In this way supervisors conduct surveillance, monitor observable behaviors and outcomes, and then provide corrective direction or coaching to subordinates who are rewarded or punished based on degree of compliance. An alternate mechanism is clan control which involves informal social structures to establish and reinforce alignment on objectives for actors who have been recruited into the “clan” based on perceived shared values.

Clan control is part of a family of socio-ideological models of control that operate at the level of beliefs and values (Kärreman and Alvesson 2004). Clan control has been described as using concertive methods (Kohli and Kettinger 2004). Concertive control is evident in environments that move toward empowerment and flat organizational structure with "more ideologically based designs drawn around unimpeded, agile authority structures that grow out of a company's consensual, normative ideology, not from its system of formal rules" (Barker 1993). This control approach begins with communal values and replaces bureaucratic rules with social conduct rules and accountability to peers for shared objectives (Tompkins and Cheney 1985). Some scholars contend that concertive control is more powerful than bureaucratic control, even while being less intrusive (Barker and Tompkins 1994). Criticism from organizational moralists is harsher:

> "As a key component of moves toward more flexible structures of accumulation, corporate culturism expects and requires employees to *internalize* the new values of 'quality', 'flexibility' and 'value-added' – to adopt and cherish them as their own – so that, in principle, their uniquely human powers of judgment and discretion are directed unequivocally toward working methods that will deliver capital accumulation. (p519) … Within organizations, programs of corporate culturism, human resource management and total quality management have sought to promote or strengthen a corporate ethos that demands loyalty from employees as it excludes, silences or punishes those who question its creed. (p519)… *as a medium of domination, the scope and penetration of management control is in principal, considerably extended by corporate culturism*. No longer restricted to authorizing and enforcing rules and procedures, it ascribes to management the task and duty of determining how employees should *think* and *feel* about what they produce. (p522) … In the installation of corporate culture/human-resource-management/total-quality-management programs, every conceivable opportunity is taken for imprinting the core values of the organization upon its carefully selected employees. To the extent that [it] succeeds in this mission, corporate culturism becomes a medium of nascent totalitarianism. (p523)" (Willmott 1993, emphasis in original)

2.2.2 Control Theory Applied to IT Project Management

The pioneering days of software development is perhaps an extreme example of self-control where projects advanced at the benevolence and good will of group members alone in the absence of structure and organization control (Sault et al. 2006). Plan-driven methodologies align closely with the behavioral processes and measurable output standards associated with Control Theory's bureaucratic controls (Choudhury and Sabherwal 2003; Kirsch 1996, 1997; Nidumlou and Subramani 2003). The stereotypical plan-driven approach establishes a behavioral process in the form of planned project stages that progress in sequence. Plan-driven approaches emphasize a philosophy of generating formal plans, budgets, task lists and schedules, against which progress and outcomes can be objectively measured. Requirements specifications and traceability matrices provide explicit criteria against which validation and acceptance testing is performed.

In contrast, the agile approach aligns with the informal social reinforcement mechanisms associated with Control Theory's clan controls (Maruping et al. 2009; McHugh 2011; Persson et al. 2012). The synergy between agile self-managing teams and concerted control is "inescapable" (Hoda et al. 2012). Structure is provided by a common set of values formally encoded in the Agile Manifesto. These values are instilled and reinforced through rituals and ceremonies in the form of sprint planning meetings, daily standup meetings, story-board walls, and retrospectives that nurture and support a collective culture (Whitworth 2008). Agile projects are not uncontrolled, but subject to control through "'peer pressure' and 'control by love'" (Takeuchi and Nonaka 1986). These concertive control mechanisms grounded in consensual values reinforced in self-managing or self-directed teams with social rules that sanction modes of conduct are both empowering and coercive (Proença 2010) and have been described as more powerful with "a greater ability to control than the bureaucratic system it replaces" (Barker 1993).

2.2.3 A Framework for IT Project Management Styles

Bureaucratic and concertive controls are often combined within organizations (Bierly and Spender 1995). In the context of IT projects, scholars recognize the control methods to be complementary (Choudhury and Sabherwal 2003, Harris et al 2009a) with practices from both plan-driven and agile approaches frequently combined for a target project (Batra et al. 2010; Jiang and Eberlein 2009; Vinekar et al. 2006; West et al. 2011). Juxtaposing

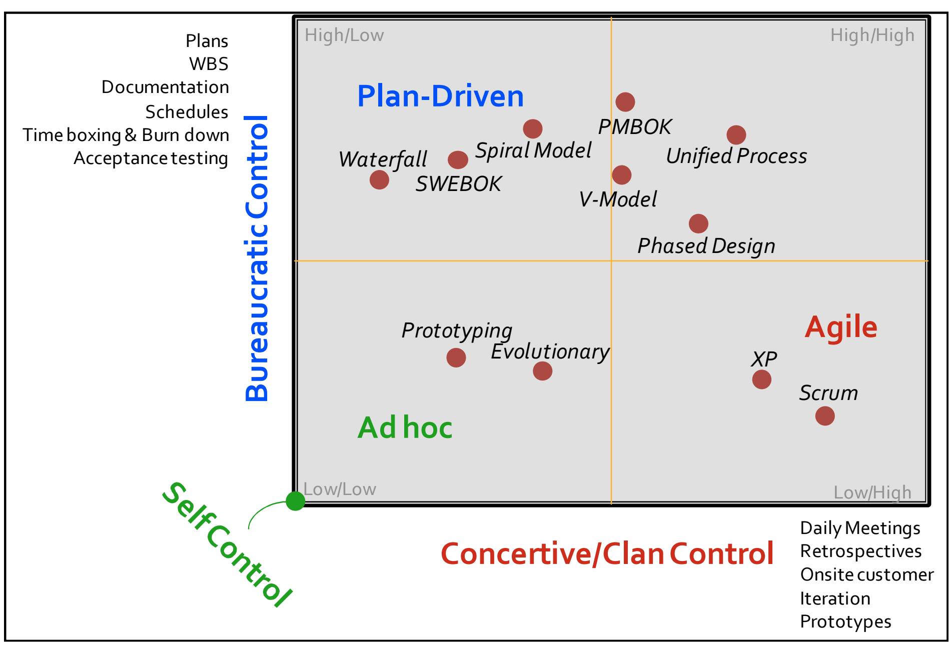


Figure 2-1 Project Management style framework

bureaucratic control and concertive control on a two dimensional grid (Figure 2-1) provides a framework for classifying Project Management styles[1]. The coexistence of practices is most evident in the Unified Process (UP), a commercial methodology that combines practices of both. The result is a balance where individual projects operate under a mixed control regimen as predicted by Ouchi (1979).

2.3 IS literature on Unexpected Challenges

2.3.1 Defining Risk and Unexpected Challenges

The phenomenon of interest for this study is unexpected challenges that emerge during the course of a project. The Information Systems (IS) and Project Management (PM) literatures have extensively covered the topic of risk and risk management as a lens to view and understand disruptive events. A temporal view distinguishes between initial risk, emergent issues, and unexpected challenges. Labels from existing literature that define these concepts are summarized in Table 2-3.

Project managers often focus on the identification and assessment of negative risks so that mitigation strategies can be applied. The starting point is “intial risk” involving potential problems or disruptions that a project may face. Once a project or activity is underway some risks become manifest in the form of actual events. “Emergent

[1] PM methodologies have been positioned within the Framework based on the judgment of the author. They serve as examples of how styles in practice may be located within the Framework. A methodical exercise to position formal PM methods has not been attempted in this study.

issues" have been examined from the standpoint of project scheduling (Zhu et al. 2005) and project success (de Bakker et al. 2010), as well as reactions at the team level (Pavlak 2004) and organizational level (Hallgren and Lillieskold 2006). Risk management planning will at some point suffer from diminishing returns as infinite imagination and cost are needed to foresee and mitigate all possible challenges (Wearne 2006).

Table 2-3 Risk, issues and unexpected challenges

Risk type	label	Definition
Initial risk	Risk	A *risk* is an unplanned future event that may positively or negatively affect your project. While risks are unplanned, they are not necessarily unanticipated. (Chin 2004)
	Risk	A *risk* is a known, yet unrealized situation. (Hallgren and Maaninen-Olsson 2005, p18)
Emergent issues	Changes	Changes, sometimes called "variances", refer to realized situations with a significant divergence to the project plan. In contrast to risks, changes are not addressed in advance, meaning that changes are managed when a situation has materialized, being reactive in nature. (Hallgren and Maaninen-Olsson 2005, p18)
	Discontinuity	A project management *discontinuity* is an occurrence that causes a project crisis; otherwise normal project management is practiced. (Gareis, 2006, p216)
	Disruptions	*Disruptions* are externally imposed on the project and take the form of deviations from the original plan.(Zhu et al.2005, p368)
	Disruptions	*Disruptions* are conditions or events that interrupt or impede normal operations by creating discontinuity, confusion, disorder or displacement. (Madni and Jackson 2009, p181)
	Issue	Once the risk actually happens, it becomes an *issue* and should be addressed appropriately. (Chin 2004)
Unexpected challenge	Deviations	*Deviations* are defined as unexpected events that require attention from the project team because they interfere with cost, time or scope goals. (Hällgren & Soderholm 2010, p353)
	Discrepancy	*Discrepancy:* an unexpected failure, a significant difference between expectations and reality. (Watson-Manheim et al. 2012, p38)
	Unexpected event	*Unexpected* in the sense that the precipitating events demand action not envisaged when planning the provision of resources for new projects. (Wearne 2006, p98)

A key frustration among software developers is the concept of expanding or changing requirements and scope. This challenge did not start with software, but it is particularly acute for IS projects as software is inherently malleable. Many responses to the scope creep challenge have been proposed over the years, including: increased user involvement in requirements gathering; formal signoff of requirements to force users to actually read and understand requirements; and planned system prototypes that allow users to see, touch and interact with the features that will be part of their work process. Iterative and incremental strategies have been used to allow teams to more easily react to scope change during the project. The Spiral Development methodology (Boehm 1986, 1988), for instance, plans a series of risk-based prototypes with the express purpose of capturing an evolving view of requirements from users. The emergence of agile methods combine the use of iterative and incremental practices with a philosophy that encourages requirements change throughout the project. Requirements change has transitioned from an unexpected challenge to an expected reality that the project team is intentionally working to expose and accommodate.

Scope and requirements are not the only type of unexpected challenge. Issues that jeopardize projects have been categorized as both technical (e.g., technology issues, task issues and project issues) and social (e.g., organization issues, process issues and people issues) (Aladwani 2002). Because projects do not progress naturally of their own accord they must be managed as circumstances diverge from expectation. Understanding what is happening at the intersection of IT project methods and unexpected events is the focal point of interest of this study. Throughout this manuscript the term *fluctuations* or *Flux* is used to broadly represent the ideas of project deviations, unexpected challenges and emergent issues that have a disruptive influence on projects outcomes.

2.3.2 Risk Management Literature

Coincident with the emergence of formalized project management practices for IS projects is a progressive examination of project risk by IS and PM scholars. Early investigations attempted to identify success factors (Kappelman et al. 2006; Sabherwal and Robey 1995; Wixom and Watson 2001; Yoon et al. 1995; Zmud 1980) and sources of risk (Barki et al. 1993; Chittister and Haimes 1994; Wallace et al. 2004a). This literature thread provides a contingency view of project success which by implication offer prescriptive recommendations (factors or situations to avoid or include) that are expected to predispose a project to successfully manage risk. Having identified risk management as a recurring success factor, the literature recommends a litany of best practices (Austin and Devin 2009; Boehm 2002, MacCormack et al. 2001; Pich et al. 2002; Soderholm 2008) that too often fail to produce desired results (de Bakker et al. 2010; Keil et al. 1994; Kutsch and Hall 2005; Pender 2001). The paradox of the contingency approach is an endless supply of contingent factors leading to increasingly complex multi-dimensional webs of cascading cross-relationships among risk and success factors (Aladwani 2002; Barki et al. 2001; Gemino et al. 2008; Lyytinen et al. 1998; Thamhain 2013) that offer little explanation of what is taking place within a project.

Contingency investigations have been supplemented by various meta-theories to justify certain prescriptive best practices. For example, Systems Theory approaches classify IS projects as Complex Adaptive Systems (Maaninen-Olsson and Mullern 2009). Literature in this domain provides analytical support for certain prescriptive practices (Meso and Jain 2006). While this area of study has yet to offer a theory that explains project team responses to unexpected challenges, it does provide important insights to conceptualize the emergent mechanisms and capabilities at work within a project faced with unexpected challenges (Maruping et al. 2009; Vidgen and Wang 2009).

2.3.3 Project Management Literature

Drawing back from the specific domain of risk management, to the general realm of IS project management provides little in the way of theoretical guidance for the focal questions of this study. The IS and PM literature has frequently examined project management methodologies in an exploratory manner. It is common to find studies that examine particular settings to demonstrate the efficacy of specific practices in certain situations. While this approach is pursued to obtain a better "understanding of the current situation" (Rumpe and Schroder 2002), we are left without a theoretical foundation (Glass 2004; Lee and Xia 2010) that guides practitioners and scholars. In this landscape scholars have repeatedly called for new theories (Koskela and Howell 2002; Shenhar and Dvir 1996; Turner 2006).

In the absence of theories specific to project management, some scholars have borrowed meta-theories from other domains. Lundin and Soderholm (1995) introduced the concept that projects can be generalized as a "temporary organization", a maneuver seeking to apply the entire spectrum of organizational theory and behavioral theories of the firm to project management. Other examples include the use of Control Theory (Harris et al. 2009b) and Economic Game Theory (Kautz 2010) to recommend agile software development methods, Critical Chain and Theory of Constraints to recommend PMBOK style risk assessment (Steyn 2002), and project scheduling (Leach 1999), and Agency theory to recommend more rigorous monitoring techniques (Mahaney and Lederer 2003). With the introduction of established meta-theoretic lenses, scholars provide more advanced analytical justification for certain prescriptive tactics, yet leave unaddressed the question of what happens when these methods are subjected to unexpected challenges. The empirical support provided characterize the degree to which certain practices are successful or not, but lacking a theory for deviation response, they fall short of explaining why. Without a handle on *why* certain methods are working, our ability to generalize across situation specific contexts remains constrained.

2.4 Examining IT Projects in the Field

In order to build a theory applicable to the target question of unexpected challenges in the domain of project management, this study pursues an exploratory case study using group cognition theory as a foundation. Theories of individual cognitions have spread to the domain of group cognition. The response mechanics of interruptions at the individual level are being applied at the group level and suggest patterns that may provide theory-building insight for IT projects.

2.4.1 Case Study Design

Using literature and theory from group cognition, propositions have been identified to guide a case study investigation (see Appendix A for a detailed report of the case study and its associated propositions). Following the recommendation of Yin (2009), the propositions represent contrasting theoretical explanations. Shared memory structures in the form of external artifacts are proposed as beneficial aids to IT project outcomes, or alternately they may serve as performance anchors with a negative influence on IT project outcomes. The factor of abundant or sparse external artifacts is believed to be associated with the concept of project management style, whereby plan-driven projects have a bias in favor of creating external artifacts in the form of plans and documentation, while agile projects have a bias to minimize documentation in favor of routines that encourage face-to-face interaction.

2.4.2 Case Study Setting

Four IT projects from teams within the health sciences industry are examined in this comparative case study. Two projects from a self-described agile organization and two projects from a self-described plan-driven organization. The study design seeks to triangulate opinions from multiple perspectives represented by four roles as presented in Figure 2-2: (1) business user {U}; (2) IT project manager {PM}; (3) IT manager {ITM}; and (4) IT individual contributor/Developer {D}.

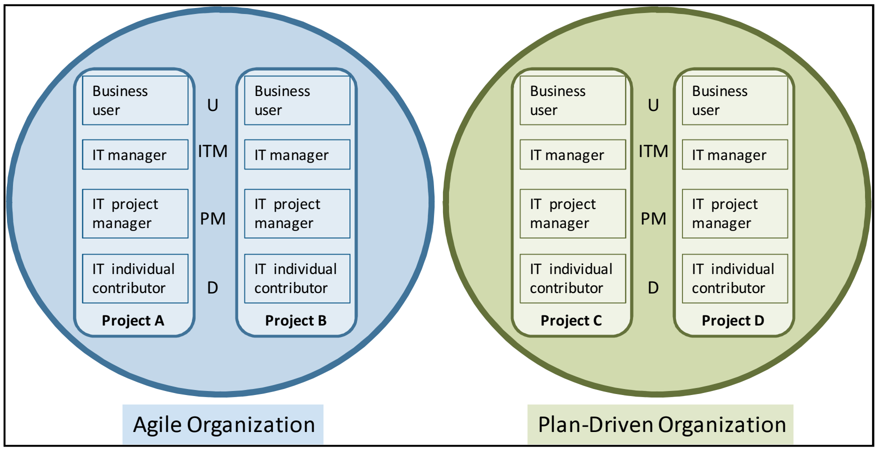


Figure 2-2 Study setting

The professionals participating in this study are based in Europe and the United States. Each individual participated in one of four actual projects lasting between 8 and 30 months and completed a milestone delivery to production during the 6-month period prior to being interviewed. Projects were selected in conjunction with a representative of the IT project office from their respective organizations. Four participants from each of the four projects were identified based on their role, for a total of sixteen interviews.

Table 2-4 characteristics of four projects

	Project A	Project B	Project C	Project D
Organization	ResCo.	ResCo.	OphCo.	OphCo.
PM Method	Agile (Scrum)	Agile (Scrum)	Plan Driven (PMBOK)	Plan Driven (PMBOK)
Team Size	7	4	8	12
Duration	30 months	8 months	18 months	12 months
Project Type	Develop & implement custom software system with integrations to other systems	Develop & implement custom software system	Implement a configured packaged system with custom integrations	Implement a configured packaged system with custom integrations
System Function	Computerized system supporting pharmaceutical research	Computerized system supporting pharmaceutical research	Computerized system supporting financial transaction processing	Computerized system supporting warehousing and distribution

Individuals interviewed had no contact across organizations while the projects were underway. A manipulation check during interviews confirmed the project management style distinction within each organization's cultural setting. A summary of Project characteristics is provided in Table 2-4.

2.4.3 Case Study Methodology

Semi-structured interviews were conducted with members of each project team. The interviews were conducted individually, lasted between forty and sixty minutes and were openly recorded for transcription[2] and analysis. The topic areas covered are summarized in Table 2-5. A pilot-interview was conducted to refine the topic areas, practice the mechanics of recording to assure a smooth flow, and maximize the opportunity to collect relevant information from the 16 interviews.

Following transcription, each interview was organized into a case study database following the procedure recommended by Yin (2009) and Pare (2004). The narrative database is analyzed using both Selective Coding and Open Coding technique to identify and organize concepts (Strauss and Corbin 1998). Selective Coding allows alignment of concepts and narratives with the a priori propositions. Open Coding is used to expose additional properties, dimensions and interrelated concepts not explicitly detailed prior to

[2] Transcription performed by a freelance journalist with a laypersons understanding of Information Systems Development and Project Management.

the interviews. Emergent ideas are used in post hoc analysis to provide a more mature theoretical explanation of events.

Table 2-5 Interview topics

Topics
Project success criteria and project outcome
Project characteristics (size, duration, type of project, project management methodology, constraints)
Planned risk mitigation strategies (including role of documentation, prototypes, iteration, and user interaction)
Unexpected events and challenges that jeopardized project success
Strategies and tactics used to manage unexpected events/challenges

Secondary data has also been collected in the form of project status reports. These status reports were collected across the entire corporation (not limited to the four projects or the two organizations) to provide a cross-sectional view of project challenges for hundreds of projects across all divisions of the parent companies.

2.4.4 Case Study Findings

Analysis of the case study proceeded in two stages. The first stage assessed a priori propositions inspired by the ideas of group cognition using positivist and interpretivist methods with a confirmatory intent. Results from this stage were equivocal, and suggested an alternate theoretical perspective is needed (see A.3.2). A second stage assessment applied a Critical Realist approach to expose underlying truths and drew from the associated mechanisms a descriptive theory.

The Critical Realist approach is particularly relevant when the view and intention held by actors may be incorrect with regard to the true effect and value of various behaviors. Just this situation appears to exist for IT project management where contradictory actions are prescribed to achieve the same target outcome (e.g., emphasizing plans and documentation or delaying plans and minimizing documentation). In such situations social scientists are justifiably skeptical of methods that rely on the interpretation of first person actors.

Critical Realism asserts the existence of a reality independent of our knowing of it; there are underlying truths that exist in the "ways of acting of things" (Bhaskar 1975; Mutch 2010). Through the efforts of social science it may be possible to describe the mechanisms of that underlying reality as a descriptive theory. This underlying reality includes the entities, structures and causal powers that exist independent of our understanding, and have their effect whether observed or not. Causal powers generate outcomes in the form of actual events. These events may be observed, and when experienced and perceived they enter the realm of the empirical. The interviews summarized in section A.2 capture a subset of empirical events from four IT projects. These events and the settings in which they took place are analyzed here in the stepwise fashion detailed by Wynn and Williams (2012).

The methodological principles of Critical Realist case studies begin with identifying and explicating events occurring in the natural setting. The components of social structure and situational context are then identified along with the relationships that exist among them. Through a process of retroduction (employing abductive reasoning, as opposed to inductive or deductive) the sources of power, linkages between structures, and thematic tendencies that form narratives ("first this happened, then that happened, and then the other happened, and it ended up like this") are analyzed to suggest candidates for generative and causal mechanisms (Becker 1998, Mingers 2012, Saldana 2011, Thomas 2010). Logical and analytic support provides a means to refine the ideas into plausible theory. In the course of applying this Critical Realist methodology to the information gathered during the case study, a plausible theory of unexpected project challenges takes shape in this chapter.

2.4.4.1 Events

Two aspects of events are detailed in this section. First is the nature and type of unexpected challenges that serve as the instigating event of interest. Second is the variety and type of response by project participants. The project as a social collaboration is the central unit of investigation for this analysis. The dynamics of group problem solving and the emergence of a shared collective cognition are most centrally related to communication (Cooke et al. 2007; Roloff and Van Swol 2007; Warner and Letsky 2008). Therefore this investigation is drawn to the events, responses and patterns involving communication.

2.4.4.1.1 Unexpected challenges as events. The narrative detail for unexpected challenge events is provided in Appendix A (section A.2.2). Some issues were anticipated by the project participants with specific mitigation strategies identified and implemented. Even so, some events were encountered in a way that surprised respondents. An illustrative example comes in the form of requirements exposed during acceptance testing after the developers thought they were done with all implementation tasks. Project A assumed their incremental development method with periodic demonstrations would expose all user requirements. Participants expressed genuine surprise that important requirements remained hidden. Similarly, Project C anticipated technical challenges as revealed by the inclusion of a senior technical analyst. When the global standardization strategy yielded to local customization, the quantity of change exceeded the resource capacity allocated for the tasks of testing and debugging, and these resources became a bottleneck for progress in the project. These are examples of anticipated risks, yet when and how they occurred was unexpected. Unexpected challenges from the case study are summarized in Table 2-6.

An important characteristic of unexpected challenges is their place of origin. The "In/Out" designation categorizes the event as originating within the scope of focal activity for the project, or outside the project's primary domain of focus. Issues that emerge within the project focal scope are often considered to be a natural part of the project. Managing issues that emerge from the internal actions of active contributors may be viewed as a natural continuation of the task or process. Managing issues that emerge from outside the project's focus are different, as contributors may view these not as something the project *did*, but rather as something that is *happening to* the project. The contributors view themselves as instigators and responsible for internal project issues, but view the project as a victim of issues originating from outside. Because external fluctuations emerge from the periphery of attention, they are inherently more difficult to recognize and understand. While contributors are likely to have some existing momentum and mental frame for internal deviations, those imposed from outside require participants to begin assessing and problem solving with a much less mature cognitive model.

Table 2-6 Unexpected events

		Agile		Plan-Driven	
Unexpected Event / Issue	**In/Out**	**Project A**	**Project B**	**Project C**	**Project D**
Staffing fluctuation (project contributors choose to leave)	In		✓	✓	
Staffing fluctuations (organization removes contributors)	Out	✓	✓		✓
Staffing fluctuations (new contributors join the project)	In	✓	✓	✓	✓
Staffing fluctuations (contributors split time between projects and non-project duties)	Out		✓	✓	✓
Technology fluctuations (emerge from actions of contributors on focal tasks)	In	Expected	Expected	Expected	Expected
Technology fluctuations (imposed by external entities)	Out	✓	✓	✓	✓
Scope fluctuations (users change their mind)	In	Expected	Expected	✓	
Scope fluctuations (hidden during implementation, exposed at end)	Out	✓	✓	✓	✓
Scope fluctuations (change in organization strategy)	Out	✓	✓		
Alignment (stakeholders with conflicting goals)	Out			✓	
Alignment (contributors with conflicting goals)	In		✓		
Intervening project	Out				✓
User training failed	Out				✓

2.4.4.1.2 Responses as events. As the projects encountered unexpected challenges, participants responded in a variety of ways. The experiences remembered by participants form a record of what happened that may be different than what individuals wanted to happen. Emergent tactics are organized into groups that reflect common traits as follows:

Delay schedule and extend project duration

- Project A resumed development sprints instead of proceeding with global rollout when acceptance testing exposed significant new requirements.
- Project B experienced schedule delay imbedded in management decision to expand scope.
- Project D experienced an intervening project that diverted IT contributors and forced schedule delay of many weeks for the original project.

Reducing scope

- Project A negotiated with the steering committee to remove optional functionality and extend deadlines.
- Project B abandoned "nice to have" functionality and code reuse objectives in order to deliver required performance.
- Project C resorted to manual processes that worked outside the new system when features were deferred and delayed. This allowed the project to deliver on time, but in an incomplete form that subsequently received functionality upgrades.

Expanding Scope

- Project A expanded scope to additional customer/user sites, with additional requirements.
- Project C expanded scope with steering committee approval to address local requirements that could not be accommodated by global standards.

Local innovation

- Project B created its own work-around solutions to the technical problems presented by unstable third party software libraries.
- Project C created new set of logic to repair data corrupted by the country/language specific character set issue.

Alter resource/staff level or mixture

- Project A reformed around a new project manager.
- Project A integrated a replacement technical lead and additional staff.
- Project A instituted extra discussion, code-reviews and informal dialog to accomplish knowledge transfer to new staff.
- Project B fought for and received additional staff.
- Project C reported very long workdays for several months.
- Project D assimilated a new senior analyst (technology expert replaced project manager).
- Project D added extra resources for several weeks to work through training problems and data problems.

Networking and knowledge mining from “outsiders”

- Project A had a chance encounter with a remote user that revealed hidden requirements (Accidental non-routine communication with extended user community).
- Project A engaged the infrastructure department to reverse a disruptive upgrade (Reactive non-routine communication to outside group with domain responsibility).
- Project A engaged web-services library developers to report bugs (Reactive non-routine communication to an outside group with domain responsibility).
- Project B had an accidental encounter that helped resolve technical problems with a third party software library (Accidental non-routine communication with technical experts).

- Project C engaged infrastructure department on performance problems (Reactive non-routine communication to outside group with domain responsibility).
- Project C engaged Business Intelligence department seeking an existing solution to the language specific character set issue (Reactive non-routine exploratory collaboration with outside group facing similar challenge).
- Project D engaged professional trainers to re-train users (Reactive unplanned knowledge transfer from domain expert).

Specialty sub-groups

- Task force: Project C task force to address payables use-cases not handled by global standard.
- Brainstorming: Project C Networking problems were addressed in cross functional "war room" conference involving project contributors, service providers and outside experts.
- Vendor: Project C engaged the software vendor with contract change orders to perform unplanned development.
- Task force: Project D task force to deal with Inventory and Data Integrity cleanup.

Remove Distractions

- Technical death sprints: Project B conducted several "technical death sprints" limited to integration work and refactoring with no new features, reduced interaction with outsiders and no demonstrations.
- Isolated participant workspace: the project manager of Project B fought for a project space and got it. The contributors sat together, and away from other project groups.

The last two types of responses reveal contrasting behaviors among the approaches. In the face of acute stress and uncertainty from unexpected challenges the Planners organize new sub-groups with special assignments and increase their non-routine communication. Agile project leaders formed special sub-teams during normal circumstances (Project A formed a technical committee around a specific user community working group; Project A formed three work stream teams: database schema, service layer and front end). However, in response to uncertainty Agilists took steps to remove distractions and focus on innovation from within. Where problem-solving communication continues to take place, participants described it to be "accidental."

2.4.4.2 Sources of Structure

Several elements of structure are common to many IT projects. The first element of structure involves the project task or goals (what the project is trying to do) as well as its constraints (what the project cannot do). The goals and objectives have a direct bearing on the duration of the project and staffing (both in terms of knowledge, skills and ability as well as size). This leads to a second source of structure, the project management methodology used for the project. This defines many expectations for how contributors go about tasks in the form of documentation templates, plans, and work-breakdown-structures for plan-driven projects or daily standup meetings, sprints and retrospectives for agile projects. A third element of structure is the core project team. The most visible manifestations of this structure are the membership and their respective roles. In some situations these roles are formal and enduring (e.g., the project leader) while others are temporary (e.g., note taker for requirements gathering workshops).

Upon these pillars of structure other secondary structures are formed, such as the selection of technologies and the duration of the project. As the project unfolds through a series of activities, routine events emerge and decisions are made. Many of these unique structures are transient to the project in progress, while others become enduring in the form of habits and routines.

2.4.4.2.1 Task and objectives as structure. The primary source of structure involves the task, which is an IT implementation project. Contributors direct their focused attention to activities, decisions and sub-tasks of the project to a far greater extent than other peripheral concerns such as long term career objectives, sales of the firm, or even certain shorter term decisions such as where to have lunch (Shenhar et al. 2001). While peripheral concerns may garner some attention from individuals, the objectives, task assignments and intellective puzzles of a project provide the guiding structure to focus attention of participants. Additional structural properties exist in the form constraints identified by the situation and organization.

The primary task for each of the IT projects studied was to deliver a computer system that supported a specific business function. Project A replaced an existing system that was incompatible with a reengineered business process. Project B and Project D were replacing an existing computer system that had reached end of life. Project C implemented a computer system to automate a previously manual business process. While the underlying motives for each project differed, the general objective to implement a new computer system that collaborated with users to accomplish a business process serves as a common theme. Each project involved data exchange interfaces with other computer systems, necessitating some degree of coordination with outsiders.

A relevant aspect of the structure provided by task, goals and objectives, is the propensity for these to change over the course of an IT project. The environment of change itself is part of the structural landscape for these projects. While somewhat persistent, each case study project experienced requirements and scope changes:

- Business needs shift during a long project. The business processes of the user community continued to evolve during the lengthy 30 months involved with Project A's implementation. Requirements in this situation are a moving target.
- A strategic decision made by management changed the scope of Project B to cover multiple sites. With a single decision the requirements of the target system expanded dramatically. Multiple existing business processes that evolved independently needed to be supported. New opinions and preferences for interaction patterns entered the picture. In such situations projects take a different trajectory, responding to needs from multiple directions instead of a single target context.
- A different sort of goal shift occurred in Project C, where the original strategy involved adapting local business processes to a single global model. As the project moved to accommodate new requirements driven by local cultural and statutory obligations, the staffing level and skills of contributors were out of alignment with the demand for regression testing and bug fixing required.

While objectives and requirements represent what the project should do, constraints serve a similar structural influence role by declaring what the project cannot do. Examples of project constraints in this case study are as follows:

- Project C had a schedule constraint. Many users were losing their jobs as the business function moved to a regional service center in a different country. While retention incentives were in place to mitigate the risk of early departure, when the regional service center opened (a firm date) new employees needed to be trained and the new system needed to be operational.

- Project A and B were constrained to develop new software as no acceptable off-the-shelf commercial packages met the business needs. In addition, the legacy systems were deemed to be no longer viable.
- Project C and D were constrained to enhance and implement a configurable commercial off the shelf software package selected by an earlier project.
- Project D had a regulatory constraint. Due to the nature of the products being shipped, the system participated in enforcing regulations related to quarantine, chain of custody, and environmental controls. The project has no flexibility and must follow a rigorous validation test protocol based on matching explicitly agreed requirements with test cases objectively demonstrating expected results.
- All projects were constrained to use computing platforms provided and operated by the IT infrastructure department within the company. This included some standards for operating systems, databases, security, and location of the computing environment. Similarly the client side computing environment was constrained to certain web browsers and technology sets on the user's workstation.
- None of the projects had fixed cost/resource constraints (though Project C in particular was expected to provide a return on investment that would be at risk with significant cost escalation). All projects were able to negotiate increased resources from management and steering committees.

2.4.4.2.2 Project management processes as structure. The project participants identified their projects as plan-driven or agile. The structure of the associated process is reinforced and guided by the decisions and actions that follow. Examples can be seen in the process technologies used for each project, as well as communication habits and planning behaviors.

Process Technologies are toolsets employed by the project team to perform their tasks. Organizations encourage certain toolsets in a variety of ways, including paying for software licenses, arranging training for groups and individuals, and selecting staff with experience using certain tools. While the availability of tools is often strongly influenced by the organization at large, the degree to which tools are actually employed is controlled by the team. The following process tool choices were exposed during this case study.

- Tools common to all projects include a word-processor for documenting user requirements and user stories, as well as a team web site of some form.
- Project A and B adopted the organization's standard project toolset called JIRA for issue and task prioritization and tracking as well as defining the teams workflow.
- Project A and B adopted the JIRA add-on tool Greenhopper to schedule sprints and manage backlog.
- Project A and B adopted the Confluence tool to build wiki pages to store and manage requirements and other documentation.
- Projects A, C and D reported using a graphic diagramming tool for documenting business processes. This included both as-is models and to-be models.
- Project B and C reported using spreadsheets to catalog requirements as well as track issues.
- Project B and C used MSProject to schedule task assignments, identify the critical path for activities and allocate resources.

The various tools are cataloged in Table 2-7. Alignment of these choices supports the idea that projects migrate to and use process tools that fit the project management style of the project. This is consistent with earlier research showing that style effects project management tool choices (Fox 1997), and that project management tools and techniques are used in groups as "toolsets" (Besner and Hobbs 2012).

Table 2-7 Process tools

	Agile		**Plan-Driven**	
	Project A	**Project B**	**Project C**	**Project D**
Project web-site	Confluence	Confluence & Sharepoint	eRoom	eRoom
Requirements list	MSWord & JIRA	MSWord & JIRA	MSWord	MSWord & Excel
Task Planning	Greenhopper	Greenhopper	MSProject	MSProject
Status tracking/reporting	Greenhopper	Greenhopper	MSWord	MSWord & Excel
Graphic diagramming tool(s)	✓		✓	✓
Issue Tracking	JIRA	JIRA	Spreadsheet	Spreadsheet

Meetings and face-to-face interactions were apparent in both plan-driven and agile projects. These represent intentional and planned interactions with a purpose or agenda characterized as *proactive communication* and *coordination*. These behaviors reveal a duality of structure described by Structuration Theory (DeSanctis and Poole 1994; Orlikowski 1992), in that the communication actions both emerge from the structural influence of the project management methodology and also through habit forming repetition, creating new structures with recurrent influence. Examples of proactive communication interactions are as follows:

Interaction among primary contributors

- Project A and Project B held daily standup meetings. These provided a daily forum for recognizing progress and challenges. Often these meetings triggered subsequent working level interactions to address specific issues.
- Project A and Project B held sprint retrospective meetings. These provided a periodic forum for the participants to recognize what is working well and critique behaviors that are not. This forum also allowed important information to be communicated through dialog to mature shared and transactive memory.
- Project A and Project B held sprint planning meetings. During periodic gathering the project participants identified the tasks and functionality (user stories) targeted in the next sprint.
- All projects involved coordinated action among multiple individuals engaged in decision making and collectively working on tasks, both mundane (e.g., applying configuration settings) and creative (e.g., design and problem solving).

User Interaction

- Projects A, B and C held requirements gathering workshops. These served the role of a kickoff for the project that introduced the users to project contributors and attempted to gather a complete picture of the business process and functional requirements of the system.
- Project A employed an IT business analyst as a proxy to represent users. Actual users were assembled for the acceptance task after development was complete.
- Project A built ad hoc prototypes to demonstrate specific user stories and presented these to user proxies for feedback but only occasionally confirmed with actual users.
- Project B user interaction and communication was channeled through focal point representatives. While these users were from a single site, IT participants assumed

these individuals would represent the full range requirements across all sites. A broader collection of users performed acceptance testing after development was complete.

- Project C engaged multiple users directly as collaborators with daily interaction. These business users split their time between regular duties and the project.
- Project D engaged a subset of users, including representatives from all functional groups at the target site on a day-to-day basis.
- Project D engaged the full user community in day-in-the-life testing and training exercises as a planned and scheduled series of events over seven weekends.
- All projects scheduled and coordinated a user acceptance-testing event at the end of the implementation effort. This involved an expanded set of users, many who had not been involved in the implementation project.

Peer-to-Peer interaction with individuals not accountable to the project

- Project B conducted regular demos approximately every two weeks for two audiences. The first was developers of other IT projects delivering complementary subsystems that will eventually be combined into a larger integrated system. The second audience was user representatives who were presented a scripted presentation of the current user interface.
- All projects interacted with an established infrastructure department. At a minimum this involved coordination of hardware provisioning. More involved interactions occurred when the project participants initiated communications with this department to address specific problems.
- Project B had bi-weekly demonstrations among distinct teams who were separately building subsystems planned for assembly or integration into an overall system.

- Project C interacted with the third party software vendor with formal statement-of-work contracts and specifications for expanded functional capabilities needed to address scope creep.
- Project D sourced programming for functional changes to a development department with responsibility across multiple projects.

Hierarchical communication relates to communication that occurs with management and organization decision makers. For many IT projects this involves key user management who participate in a steering committee.

- Projects A, C, and D held scheduled periodic meetings with a steering committee to report status and receive updated guidance. Project A's steering committee was directly involved in defining functional requirements and acceptance testing. This project also had a special technical steering committee focused on one particular area of functionality. Meetings and interaction with the steering committees took place through conference calls with participants spread across multiple locations. Project C's steering committee held authority for approving all functionality, in particular any local requirements that did not fit the global standard had to be reviewed and approved by the steering committee. The steering committee also had authority over cost and schedule changes. Project D's steering committee held authority over readiness of both the system and training, including the decision of when to move to production. Project representatives met with this committee to report status on a scheduled basis. Interviews revealed the information flow was bi-directional as the committee participated in vetting issues.
- Projects B and C reported status to IT and business management beyond the steering committees on a periodic basis (approximately monthly). This form of communication appears to have taken place among all the projects and served as a communication supplement where steering committees were involved.

Table 2-8 captures the frequency of the various proactive communication patterns and habits with intensity shading. The pattern reveals agile behaviors that emphasize communication within direct contributors while plan-driven behaviors emphasize communication with outsiders (both hierarchically with sponsors and management, as well as peer developers, projects, departments and external vendors).

Table 2-8 Communication and coordination patterns and habits

		Agile		Plan-Driven	
		Project A	**Project B**	**Project C**	**Project D**
Core Team	Daily Standup meetings	D	D		
	Sprint Retrospectives	P	P		
	Sprint Planning meetings	P	P		
	Coordinated activities	D	D	D	D
	Specialty sub-group formation	P	P	R	R
	Brainstorming			R	
User Interaction	User Proxy (IT business analyst)	F			
	Focal point users	R	F	D	D
	Broad user community	W, Te	W, Te	W, Te, Tr	Te, Tr
Peer group interaction	Other developers & projects	R	P	R	R
	External subject matter experts	R	R	F	F
	Infrastructure Department	R	R	R	R
	3rd Party Vendors			R	
Hierarchical	Steering Committee	P	R	P	P
	Management representatives	R	R	P	P

Frequency in decreasing order of intensity: (D) Daily = continuous, imbedded as continuous contributor; (P) Periodic = specific time increment such as bi-weekly or monthly; (F) Frequent = but no recurring pattern noted by participants; (Te) planned acceptance testing near the end of the project; (Tr) planned training; (W) planned workshop; (R) Reactive = in response to specific issues.

External communication may be particularly valuable when dealing with unexpected challenges imposed from outside the focal attention of project collaborators. An important purpose of documentation artifacts recognized in the IS literature is their role as boundary spanning objects that facilitate communication with outsiders and circumvent barriers (Gopal and Gosain 2010; Pawlowski and Robey 2004). The structural influence of the project management bias extends to the ability of documentation artifacts to fulfill a role as a communication aid. Both plan-driven and agile projects create documentation, but with important differences. First is the formality of plan-driven documentation that includes templates to assure full content coverage and approval with signatures to enforce careful review by project principals, users and stakeholders. This formality does not exist with agile projects which operate with an assumption that requirements are necessarily incomplete and are only revealed through repeated exposure to an evolving system. Instead agile projects capture intentionally brief information artifacts. This philosophical point of view is tied to a second difference in how and where documentation artifacts are used. The Planners use documentation both among technical contributors and as an object to facilitate information exchange with outsiders. The Agilists focus use of documentation among direct technical contributors, and treat documentation artifacts as counterproductive interacting with users, stakeholders and other outsiders.

2.4.4.2.3 Project team provides weak structure. A team is a “bounded and stable set of individuals interdependent for a common purpose” (Wageman et al. 2012). The “teams” working on IT projects have a common purpose in the previously described project goal. Even where this goal evolves and shifts with changing requirements, there remains a focal objective throughout. Stable membership, however is more elusive. The participants of the case study projects were representative of IT projects common in contemporary organizations. At different times the teams were virtual, distributed, dispersed and global. In addition, many participants were not exclusive to a single team, but worked simultaneously on more than one project, often with additional production responsibilities. These projects were performed by “porous teams” sustained by collaborative behaviors that “would, by the traditional definition, not be teams at all” (Wageman et al. 2012). Nevertheless, a degree of structure emerges from repeated interactions. Participants settle into recurring roles and individuals gain confidence in the abilities of other collaborators. Collaboration among virtual and matrix teams have been shown to develop performance-enhancing transactive memory systems, even when face-to-face interactions are limited (Maynard et al. 2012). The longer a project is active the more repeated interactions that occur and the more likely a group of collaborators will bond and form an interdependent entity with a distinct identity.

Some scholars propose role stability as a substitute for membership stability (Higgins et al. 2012). The case study projects did provide examples of various distinct roles. The plan-driven projects had a leader designated by the organization to serve as Project Manager. The agile projects had a leader designated by the organization to serve as ScrumMaster[3]. The role of Technical Lead emerged for Projects A, B and C. The concept of elite status existed for these individuals with certain key decisions either assigned to these individuals, or some form of review for key decisions passing through these individuals. The role of Business Analyst was specifically identified for Projects B, C and D. These individuals typically have responsibility for business domain knowledge

[3] Many of the project participants identified this role using the label “project manager”

that supplements knowledge of the computer and software technologies used to implement the target system. As noted previously, all projects experienced staff fluctuations, which included replacement of individuals in elite roles such as the project manager and technical lead. The project leader serves a unique role around which much of the tempo of a team revolves. In agile projects, where the leader may not have elevated decision-making authority, there is an increased role as communication and coordination "hub" for the team (Vigden and Wang 2009). Both the roles of lead decision maker and communication hub require mature shared memory models to be effective. As team membership changes, this necessitates revised interaction patterns, restoring shared memory models, integrating new distributed memory and rebuilding transactive memory.

The stability of roles in IT projects appears to be less reliable than membership. Plan-driven projects that progress through a predictable sequence of stages will engage the business analyst role heavily early in a project during requirements gathering, and lightly or not at all later in a project. Similarly, the role of tester is transient and most prominent as a project nears its conclusion. Participants with specialty skills, such as a database administrator, may provide crucial contributions during some portions of a project, yet be a non-factor for challenges that arise in other areas, such as network performance. Agile projects that reassign roles and duties at the end of every two or four week sprint, plausibly provide even less role stability.

A final characteristic candidate for stable team structure is the team size. The reality of the case study projects suggests this too is dynamic. All projects reported adding collaborators during certain time periods. Participants in Project A, for example, provided highly variable characterization of team size ranging from 3 to 10. Interviews for all case study projects provided similar variance, revealing that team membership and size was not stable from the perspective of participants.

2.5 Summary of Events and Structure

This assessment of contrasting cases reveals three prominent sources of unexpected events: staff fluctuations, technology fluctuations, and scope or requirements fluctuations. These are supplemented by a variety of less frequent challenges in the form of alignment, intervening projects and ineffective training. Responses introduced by IT project teams are themselves events within the project lifecycle. Several responses were common across project management styles, including: altering schedules, increasing or decreasing scope delivered, innovation and prescriptive changes in staffing, and boundary spanning communications. Two response tactics appear to be associated with a particular PM style: the establishment of buffers to reduce distractions is particular to agile teams, whereas the formation of specialty sub-teams was a change in behavior evident for plan-driven teams. In the case of specialty sub-teams, the change may be notable as a deviation from standard behavior for plan-driven teams. Agile teams exhibit this type of requisite variety on a routine basis as they reform sub-teams around new requirements at the end of each sprint.

Additional insight into the different communication patterns is revealed in the structural habits and routines. Agile style emphasizes interactions within the team and plan-driven style emphasizes interactions outside the team. These distinctions appear to align with the control mechanisms for each PM style. Plan-driven style emphasizes predictable events and communications that are observable by outsiders and supporting bureaucratic controls. Agile style values autonomy and accountability within the team to devise creative solutions that bolsters concertive and cultural controls. These areas of emphasis serve as a springboard for new theory development in the next chapter.

Chapter 3

Proposed Theory of Reactive Decoupling

Two themes are explored here in an effort to devise a theory of unexpected project challenges. The first is the baseline difference in communication patterns among participants working on focal project activities, and outsiders not dedicated to or focused on the project. A project management bias appears to exist where Agilists use documentation artifacts among direct collaborators, and Planners are observed using these objects to facilitate communication with outsiders. A second theme is reactions to unexpected challenges where the response involves communication and coordination. Agilists remove distractions in an effort to maximize a high energy level on creative solutions generated by direct contributors. The intentionally established buffers decrease interaction with outsiders. In contrast, planners increase their networking with outsiders, often recruiting new transient participants for specialty sub-groups to address unplanned activities. The concept of a social boundary appears in organizational behavior literature in the form of boundary spanning and loosely coupled systems.

3.1 Loosely Coupled Systems: Seed of a New Theory

Existing theories in the form of loose coupling and boundary spanning describe mechanisms central to the tendencies of IT project teams faced with unexpected challenges. Information flows readily within and across teams that develop communication shortcuts as their common experiences build transactive memories. These transactive memories are not as well developed across the boundary of the team. As a result, communications across boundaries represents an important challenge that must be addressed in the course of dealing with both routine and non-routine problem solving. The theory of loosely coupled systems describes multiple forms of boundaries that are relevant to communications for IT project teams. In the process of cataloging the types of boundaries prevalent in the IT projects studied here, a new form of boundary, cognitive inertia, is introduced. At the intersection of these ideas emerge a new theory of unexpected project challenges.

3.1.1 Loose Coupling

IT project teams are systems that may be tightly coupled or loosely coupled. Loose coupling is a powerful multi-faceted concept within organizational theory. Coupling relates to the ties that connect entities and sub-systems (Orton and Weick 1990; Weick 1976). When those ties are strong, two entities will be *tightly coupled*, with information, meaning, control and influence linkages that are strong and direct. *Decoupled* entities are utterly disassociated and unresponsive to linkages with counterpart entities. *Loosely coupled* entities exhibit a degree of autonomy, modularity, requisite variety and discretion, yet maintain a persistent element of common identity and purpose with other entities. Loosely coupled project teams are characterized by the dialectic of both responsiveness and distinctiveness (Hallgren and Soderholm 2010; Orton and Weick 1990). Responsiveness provides linkages that couple the team with the task, the plan, and the organization. Distinctiveness separates the team from prescribed courses of action; it allows novel tactics and creative sub-team organization with solutions unique to the project. While these concepts appear to be mutually exclusive, their dialectic properties are possessed and exhibited by degree and sometimes simultaneously for a single project.

Orton and Weick (1990) surveyed literature on loose coupling and identified many types, some of which resonate with challenges identified in Table 2-6 (page 45). Coupling between *subunits* of an organization would relate to the ties which may exist between distinct project teams, between a project team and a service organization, or between functional units of an organization (e.g., sales and IT). Coupling between hierarchical *levels* of an organization involves the form and strength of ties between management and subordinates, between company executives and functional units, or between executive sponsors and project teams.

A very different type is the coupling between *intentions and actions*. Intentions equate to a priori plans or policies and procedures. Tight coupling in this sense can exist across hierarchy levels where management prescribes a process with bureaucratic behavioral controls. Alternately, tight coupling can exist within a subunit that defines a plan of action conforming to a standard, and then executes that plan in lockstep fashion. A strict plan-driven project team would be an example of a tightly coupled system. Contrast this with agile project teams that maintain a commitment to objectives through concertive control mechanisms, while at the same time possess the autonomy to adjust designs, re-sequence tasks, and the discretion to embrace new requirements on the fly. Project teams adopting agile methods embody the spirit of a loosely coupled system.

3.1.2 Boundary Spanning

Organizational subunits are more than administrative labels that identify a functional department or project team. The boundaries of these subunits distinguish one domain from another. Through the process of interaction and self-identification, individuals bond with their group, forming a sense of membership, satisfaction and group cohesiveness (Hackman 1987). Groups develop communication shortcuts involving shared transactive memory allowing tacit coordination that increases team efficiency (Wittenbaum et al. 1998). These shortcuts of local jargon, gestures and idiosyncrasies are covert in nature and may confuse outsiders and delay knowledge integration (Carroll et al. 2008). The resulting informal social structure provides genuine performance advantages for insiders, while at the same time isolating the team from outsiders.

The effects of these boundaries are highly relevant for project teams. Key information from outside the team must navigate the boundary to become visible to the project team. Not only must the message arrive, but the team must recognize the importance of information that may not conform to group jargon. Timing can also be relevant as some project teams follow an entrained process that efficiently processes new information at certain times and places (e.g., during requirements gathering workshops, or during sprint retrospective meetings). Situational awareness critical for team decision-making and performance, particularly in complex environments (Foltz et al. 2008), is diminished when boundaries delay perceiving and comprehending information.

A second effect of boundaries is to delay the propagation of information and knowledge from inside the subunit to outside entities. This has a direct implication for organizational learning, as awareness to breakdown, challenge, change and innovation is localized and does not spread to other portions of the organization (Weick 1976). Increasing structure (more bureaucracy, specialization, hierarchy and formalization) has been shown to foster the conditions that promote learning and continuous improvement across boundaries (Bunderson and Boumgarden 2010).

A third effect of boundaries is to guide participants where to focus their attention (Watson-Manheim et al. 2012). As perceived objectives are internalized into a shared memory model for the subunit, the problem solving activities and creative energies of the team become focused on a task of defined scope that is understood by the collective as its raison d'être. Responsibilities that fall outside the understood scope receive comparatively little attention and effort as the team ascribes these to outsiders with a "not-my-job" justification. Decoupling takes place as boundaries emerge where entities have goals that are misaligned, incompatible or inconsistent (Orton and Weick 1990).

3.1.3 Unexpected Challenges

IT project teams typically exist in an environment of many other groups and teams with which they need to interact. In addition to operating in parallel with other project teams, IT projects are often dependent upon ongoing service provider teams for

infrastructure (e.g., servers and networks). Other established groups and teams exist beyond the boundary of a single IT project team in the form of a general user population (which often is much larger than the user representatives or proxies participating on the core team) as well as management (both IT management and business function management) that represent the interests of the organization as a whole.
Understanding the subunits and organizational layers in a project team's environment is important to appreciating where challenges originate and which challenges can be addressed within the confines of the team, or in concert with outsiders. Fluctuations often originate outside work unit boundaries (Batra et al. 2010; Chong and Siino 2006). Table 3-1 catalogs the case study examples of many interactions with teams, groups, departments and vendors that operate outside the boundary of the core project team.

The secondary data collection summarized in Appendix A (A.2.2.5) indicates that as many as half of all project challenges originate from outside the domain of the core team. This suggests that dealing with unexpected challenges often require the team to share information and interact across the boundaries of the project team. Project teams tend to interact with representatives of the organization management and steering committees. While all projects studied here exhibited some form of interaction across hierarchical boundaries, Project D provides an exemplary illustration. The organization introduced an intervening project. The project team responded by recommending resources be diverted from the existing project and proposed a new schedule that accounted for holidays and end of year activities. The steering committee endorsed this plan and facilitated coordination across the full user community and other functional units.

Table 3-1 Organization layers

Organization Levels & subunits	Project A	Project B	Project C	Project D
Distinct project teams	• Web-Services development team • Geographically dispersed technical sub-teams.	• 3rd party vendors for Web-Services and control libraries • Parallel project teams at the program level	• Offsite/ offshore vendor developers	• Offsite/ offshore development team shared at program level • Parallel business rollout project team with distinct responsibilities
Extended user organization	• Globally dispersed user community at four primary sites.	• Four distinct globally distributed user teams	• Country specific purchasing department and payables dept. • New shared service center (SSC) purchasing dept and payables depts.	• Each warehouse represented a distinct user organization. • Over 100 users in extended user community at target site
External service provider	• Infrastructure (computers and networks)	• Infrastructure (computers and networks)	• 3rd party vendor for image scanning • Infrastructure (computers and networks)	• Infrastructure (computers and networks)
Organization Management	• Steering Committee	• IT management and business management	• Steering Committee	• Steering committee

The cases also exposed a risk of limited boundary spanning between subunits. The technical architecture choices revealed in Project B serve as an example. This team functioned as a dedicated work-team within the context of a larger program. Peer projects were chartered to create system modules that could be interconnected as complementary components of a larger system at some point in the future. Project B exercised their autonomy to select the WCF AJAX technology stack. Many months after the project was underway it became known that a peer team had chosen ASP.NET and Visual C, while a third team chose Java J2EE. Eventually the components from all three projects will be integrated into a single system. At some point a decision will be required to undertake a sizable refactoring project or absorb the support inefficiencies of maintaining the aggregate beast in production.

Administrative and hierarchical boundaries that form the layers of an organization are not the only barriers that isolate a project team. A subtly different boundary involves *idea/goal incongruence* (Orton and Weick 1990). This boundary emerges when goals of participating entities are incompatible or inconsistent. Several examples in the current case study are illustrative.

- The core team of Project A included individuals who had participated in the development of the predecessor system. The project team inherited with these members a mental model of purpose, goals and requirements for the target system. These members were placed on the team partially because they possessed this mental model and were expected to take advantage of it. However, the new system was intended to serve the global organization, which included user groups not serviced by the incumbent system. Users at different sites follow distinct processes and procedures representing a different purpose, goal and requirement set. The core team and the new user communities are separated by a boundary of ideas in the form of incongruous mental models that they may not appreciate to be different. Assuming full access and availability to these users and information, the core team must still cross the boundary created by incongruous mental models. It is not unusual for developers and users to "talk past each other" and fail to absorb the ideas and meaning expressed by counterparts. The effects of this boundary are most pronounced when one or both parties possess an a priori mental model they presume to be correct and they believe obvious to others acquainted with their task domain.

- As part of an organization-wide IT effort to increase software reuse, both Project A and Project B were encouraged to use a web-services library being developed elsewhere in the organization. That project team was responding to requirements and timelines from a variety of constituent projects in addition to those studied here. Many of those requirements were not relevant to Projects A and B, yet the relative priority with which the web-service developers pursued enhancements and bug fixes represented a goal incongruity. Managing the frequently changing web-services library requires boundary spanning to the peer project team responsible for it. Failure to navigate this boundary leaves the project team at the mercy of an uncontrolled stream of technology interruptions and what Project A members describe as a moving target.
- The extended duration of Project A prompted the administrators of the legacy system to continue making changes and enhancements to support evolving and maturing business practices. The goal of the project team was to deliver a system, while the goal of the user community was to continue to run their dynamic business. The result was that user requirements became a moving target. Each time prototype and requirements feedback was obtained, it had a limited shelf life. The dynamic pace of ongoing business progresses at a different pace than that of the project team, resulting

in a re-emergent boundary requiring continuous spanning effort. Iteration within Scrum development process offers a vehicle to repeatedly span this boundary, as long as participants remained vigilant throughout a long project.

- Project C experienced quality problems from an external service provider responsible for digitizing paper invoices. The vendor was slow to acknowledge the problem which became visible to the project team only after moving into production. The contractual relationship instilled in the vendor a profit-making objective to maximizing the volume of transactions, while the project team had a goal related to accomplishing the end-to-end business process that constrained volume until the quality issues were resolved. Incongruous goals created a boundary the project needed to cross in order to manage this challenge.
- Project C suffered poor quality and inefficiency during the test phase. New users were engaged for this activity and they were responding to different priorities. ("The system was not ready because the people that were supposed to be doing the testing were doing it improperly, again different priorities.*")*
- All of the projects examined in the case study were dependent upon an infrastructure department providing computer servers and networks. The engineering of the technical architecture is a service provided by this department external to the core project team. In the case of Project C, the infrastructure department had an objective and goal to virtualize computer servers, effectively moving all server based applications to an internally hosted "cloud computing" infrastructure. The dynamics of tuning memory, CPU allocation, network paths and storage hierarchies is an intricate technical challenge that can play the needs of one application against the needs of another on shared equipment. How this balance is managed has an implication for secondary resources such as power and air conditioning, which may become bottlenecks in an economy that lacks infinite resources. Failure to properly navigate this boundary creates a risk that the misaligned entity will react slowly to challenges specific to one project. "We tend to have the Citrix people saying it's not

their problem, and the software vendor saying it's not their problem and the VM people saying it's not their problem, the network people say it's not a network problem. OK, so it's nobody's problem". Project B also noted similar frustration with their infrastructure service organization. "Getting those people's attention and those resources, when they're so busy, it's very difficult."

- During Project B management made a strategic decision to change the scope of the project from replacement of a local system at a single site, to a global system that would service at least four sites spread across North America and Europe. The team was faced with resetting their own goals as well as those of their user proxies, extended user communities and the service departments they depend upon. In this situation the core team understood a goal reset was in order, but their partners beyond the boundary of the team may not. This creates the need for effective ambassadors to spread the message while at the same time broaden the scope of their requirements-gathering efforts.
- Project D experienced an idea boundary that forms a back-story to disruptions exposed as ineffective training and inventory related data quality problems. The legacy system allowed users to pick inventory that violated a "first-expire first-out" rule. When directed to pick a specific lot of inventory, if that inventory was not readily available users could pick convenient inventory then update the system post-hoc. To the users, one use case was picking convenient inventory that matched first-expire first-out (the preferred case), and another use case involved picking inventory that was convenient but violated first-expire first-out (the discouraged case). In the new system, the team chose to implement strict enforcement of the rule, with no post hoc work around available. Where the project team's mental model included a single use case, the user's mental model contained two. Testing and training progressed with the users only exercising the rule compliant use case. The discrepancy in mental models is a boundary between the team and the user community.

A new form of boundary revealed in this study involves *cognitive inertia*. An unengaged project participant will tend to stay unengaged unless met with sufficiently relevant motivation. Engagement of users at a cursory level is not the same as a committed partnership. A humorous vignette demonstrates the point. When preparing a breakfast of ham and eggs, the chicken is involved (she contributed then walked away), but the pig is committed (his skin in the game makes a difference that was meaningful to the pig). The importance of overcoming this boundary of cognitive inertia is apparent in the literatures of social psychology, organizational change and workplace training. Social psychology's Collective Effort Model matured to explain social loafing within collective tasks reveals that a task and its outcome must be important and meaningful to the individual to motivate their effortful participation (Karau and Williams 1993). Kurt Lewin's classic 3-step model for organizational change (unfreeze, change, refreeze) observes a quasi-stationary equilibrium that must be overcome before planned change can be successfully implemented (Burnes 2004). In the domain of training, motivation is a significant factor in learning and retention (Grossman and Salas 2011; Noe and Wilk 1993). If the complexity of a training scenario does not match (is either too elaborate or too simple), workplace training will not click for participants and commitment will remain low, serving as a barrier to learning (Dieckmann et al. 2012; Schraagen et al. 2008). Training in an artificial setting provides little opportunity for participants to perform decision making under the stress of uncertainty and event novelty, which limits learning (Crichton et al. 2008). Demonstrations and artificial scenarios are no substitute for actual interaction (Roloff and Van Swol 2007). The contextual distinction between artificial prototypes, simulated training and real-world use forms a barrier that must be crossed in order to displace cognitive inertia or to unfreeze the equilibrium. Cognitive inertia is reinforced by time constraints and distractions that transient participants and users face from the pressures of their regular job (Sloman and Borsattino 2007). The result is users who are involved but not committed; they are insufficiently engaged to internalize information as learning during training; they may go through the motions of providing cursory feedback while unintentionally passing over what later is revealed to be an important requirement. Examples of cognitive inertia from this study help to clarify the concept:

- Project B performed an up-front requirements gathering activity followed by aggressive use of prototypes and demos to elicit feedback throughout the development effort ("We had those bi-weekly global demos so that a lot of the stakeholders sit in and offer their feedback.") However the requirements were surprisingly incomplete when the product was delivered for acceptance testing. The prototypes and demonstrations were unrealistic in a way that failed to capture the attention of users who provided perfunctory feedback but failed to reveal significant requirements. ("We had interaction with the users quite often, but this was one of those unexpected changes. The user was being shown what developers were doing, but they weren't forced to work with it because it wasn't ready, which meant that what they were seeing was nice and they liked it, but they weren't really looking for what was missing." "We had live demos, and at that point they were still not providing the right feedback, because it was a guided demo. The big difference was, and actually made a big surprise to me, the guided demo was very different from the hands on, when you play with it. It wasn't until it was really in their hands that we discovered a lot of things." "Once the users had possession of it, they said 'oh wait now, I have to get this', they really went back to their core requirements and there were a significant set of requirements that hadn't been put into place.")

- Project C depended upon the finance users adjusting their business processes to accommodate a standard global system. However, much of this agreement came from user teams who would be losing their jobs due to the organizations initiative to relocate to a regional service center. These users will not have to follow the new processes. ("We were dealing with people who were actually losing their job." "Looking back at the workshop, the buy-in, the business probably wasn't where it should be." "It's the business in the end of the day whom are ultimately responsible for it to work, and in this case the business wasn't really interested") In addition many were distracted by non-project priorities. ("The people in the market were more preoccupied with closing their budgets for the year because it was the middle of budget time." "That was a problem from day one, people within the markets didn't have the same priorities as the people that were working at implementing the project.") In this situation users were providing perfunctory agreement with a strategy without reflective assessment of the implications for local requirements. These users were involved, but not committed.

  Project C had a high-profile objective to implement a standard global system with no local customizations. The strategy called for the business processes to be adapted to comply with out-of-the-box capabilities offered with the commercial software system. Each time a locally unique requirement was exposed, inertia drove the change to business team with instructions to adapt their process. Some of these local requirements were not conveniences, but constraints dictated by strongly imbedded cultural norms (e.g., country specific character sets) or statutory obligation (e.g., retaining a copy of invoices in-country). The participants must cross the cognitive inertia boundary imposed by the global strategy in order to recognize a new requirement as a constraint. Only after the boundary is crossed does the team accept responsibility for assessing feasibility and design for changes to the technical system instead of the business process.

- Project D suffered what was described as a training failure. Despite extensive walk through exercises with users involving production transactions simulated in the test environment, knowledge did not transfer in a usable way. After completing seven "day in the life" training exercises and demonstrating apparent competency with the new system, these domain experts were unable to perform their work task on the new system in production. The artificial nature of the activity presented a barrier to be crossed before users are sufficiently motivated to absorb and internalize new knowledge.

Table 3-2 summarizes various strategies and tactics employed (with varying efficacy) by projects in this case study to facilitate boundary spanning.

3.2 Reactive Decoupling: A Project Management Theory of Unexpected Challenges

Cowboy programmers, with their ad hoc practices, are largely decoupled from the organization. Their skills and role make them distinct within the organization, yet they lack ties that provide organizational control of key objectives, processes, and outcomes. Only the goodwill of developers guide these projects to deliver value to the organization. Such project teams exhibit tendencies of decoupled systems.

Planners claim that rigorous planning allows complex contingencies to be anticipated and architected into a cohesive system with predictable return on investment for the organization. By following a prescribed discovery, analysis, design and implementation process, the project provides objective and externally observable metrics

Table 3-2 Boundary spanning strategies and tactics

Tactics	Project A	Project B	Project C	Project D
Mitigation Strategies	• User interaction through proxy • Incremental development with frequent demos • Escalation: Management interaction and steering committee	• User interaction through proxy • Incremental development with frequent demos • Management interaction • Broad technology search	• Kickoff workshop for strategy alignment • Documented requirements and designs shared and approved • Iterative delivery, site by site • Escalation: Management interaction and steering committee	• Kickoff workshop for requirements gathering • Frequent (daily) user interaction • Extensive "day-in-the-life" testing and training with users • Escalation: Management interaction and steering committee
Emergent Tactics	• Collaboration: Informal networking with individuals outside the core team • Negotiation: infrastructure change was withdrawn	• Collaboration : Informal networking with individuals outside the core team • Expanding the team size added new viewpoints and knowledge	• Collaboration : Informal networking with individuals outside the core team • Negotiation: with vendor implement core product changes	• Expanding staff, including technology experts

for progress and conformance. Participants understand their position relative to others with interactions expected in a predictable and orderly manner. The intention of tightly coupled plan-driven projects is for actions of the project team to closely follow planned intentions and adhere to standards with limited discretion.

By contrast, loosely coupled systems exhibit decentralization, autonomy and delegation of discretion with activities that are not regularly inspected and leave room for self-determination (Weick 1976). “Loose coupling is a situation whereby organizations exhibit properties of both decoupled and tightly coupled systems, which are the real extremes of the organizational continuum”(Brusoni and Prencipe 2002). Agilists claim flexibility and quick response to emerging situations as a hallmark of their values and approach. In the absence of hierarchical control, self-forming teams have the ability to deconstruct and remake themselves. With each two- to four-week development iteration or sprint, the IT project team identifies and chooses a new set of objectives, then reorganizes to address those tasks. By making their own assignments, individuals and sub-teams bond and self-associate with an objective, thereby achieving high motivation and ownership for accomplishing the task.

The Project Management style framework introduced in Figure 2-1 (page 26) recognizes that plan-driven and agile project management methods possess elements of both bureaucratic and concertive control. Organizations regularly mix and match practices from both plan-driven and agile traditions to establish locally unique methodologies (Batra et al. 2010; Jiang and Eberlein 2009; Salo and Abrahamsson 2008; Vinekar et al. 2006). Furthermore, each individual project team demonstrates elements of both loosely coupled and tightly coupled systems, where plan-driven projects tend to behave as tightly coupled systems, agile projects tend toward loosely coupled systems, and ad hoc projects tend to behave as decoupled systems. Customizing practices for an individual project determines its positioning within the framework. Figure 3-1 (page 82) overlays the continuum of loose coupling as a cure or "flexure" onto the project management style framework. This places tightly coupled systems at one end, decoupled systems[4] at the other and loosely coupled systems in the middle. The four-quadrant control framework is recast into three regions reflecting the degree of loose coupling associated with project management style. The resulting conceptualization anticipates each project team is capable of both tightly coupled and decoupled behaviors, yet has tendencies associated with a certain combination of distinctiveness and responsiveness.

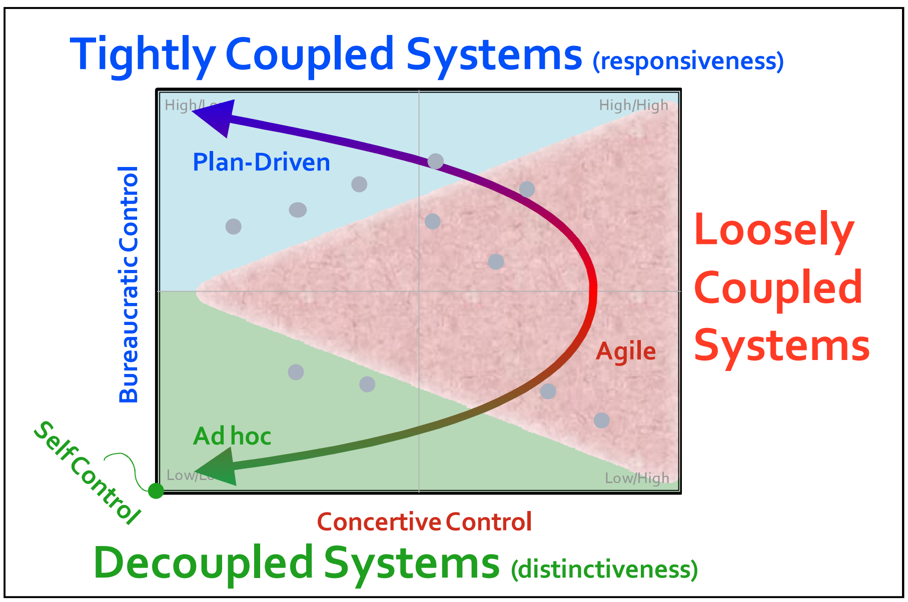


Figure 3-1 PM style framework with loose coupling flexure

[4] The theory of loosely coupled systems includes the concepts of *decoupled systems* and *uncoupled systems*. In the context of project management teams and the theoretical ideas presented here, these two forms of systems function similarly, so the conceptual distinction is not required.

3.2.1 Baseline Effect of PM Style

Both tightly coupled and loosely coupled systems have characteristics helpful for teams managing fluctuations. The self-organizing activities of loosely coupled systems enable them to acquire knowledge through an intensive, focused search (Romme 1999). The freedom to pursue actions that are neither prescribed nor directed by bureaucratic control allows these entities to pursue exploratory problem solving and experimentation. Loosely coupled systems are better able to sense their environment due to the relative absence of internal constraints and the independence of its members (Weick 1976). They interact with their target problem space more thoroughly and, by untangling causality, gain more accurate understanding. The freedom to pursue and retain a greater number of mutations allows loosely coupled systems to more easily generate novel solutions. An outcome of loose coupling is adaptability that encompasses assimilation and accommodation of change (Orton and Weick 1990, Weick 1976). "By localizing subunit responses (including behaviors, beliefs, and interpretations), and allowing inconsistencies to persist, loose coupling provides local havens for deviance and change. Indeed, loose coupling may provide the psychological safety that, according to Schein, is necessary to induce change" (Meyerson and Martin 1987).

Plan-Driven teams exhibit less requisite variety, but that does not equate to an absence of variety or creative capacity. Where innovation does emerge in a tightly coupled system, the strength of its ties to the larger organization facilitate cross boundary visibility and more rapid diffusion. Tight coupling of a subunit to other subunits and the organization as a whole allows emergent change and orchestrated change to more readily propagate through the organization as a whole (Orton and Weick 1990). Several studies have found empirical support that tightly coupled systems are more conducive to system-wide innovation and change (Firestone 1984; Mitchell and Zmud 1999; Wilson and Corbett 1983). This has implications for solutions generated outside the project team, where tight coupling helps propagate these *into* a team facing fluctuations. Where environmental contingencies and emergent issues trigger learning and local innovation, the lessons from such responses are more likely to ripple into teams tightly coupled with the organizations than those with loose or no coupling (Weick 1976).

3.2.2 Fluctuations and Reactive Decoupling

Project plans, particularly those that are shared with other teams to facilitate coordination of schedule or with management and steering committees to commit resources, provide a somewhat rigid structure that leads plan-driven projects to behave like a tightly coupled system. However, projects do not progress naturally of their own accord; they must be managed as circumstances diverge from expectation.

As a baseline, tightly coupled systems have a tendency to use their ties to reach out beyond the core team through boundary spanning to integrate knowledge drawn from other sources (Maaninen-Olsson and Mullern 2009). Managers at some organizations insist on formal documentation to facilitate predicable levels of cross-boundary interaction as situations evolve (Ramesh et al. 2006). When faced with fluctuations, the inclination is to seek assistance, either hierarchically to management or horizontally to other project teams, customers and suppliers. Something even more dramatic can happen to plan-driven project teams faced with uncertainty – they may stop behaving as tightly coupled systems and transform into loosely coupled systems.

Recognition of discrepancies, where there is a significant difference between expectations and reality, triggers a reexamination of the work situation that motivates new patterns of behavior and adapted processes (Watson-Manheim et al. 2012). A shift occurs such that the conditions of tight coupling are replaced by a new set of conditions. A new dynamic originates from an acute objective to which the team must respond. The team exerts autonomy with new actions and tactics that diverge from those prescribed in the plan. As members respond to deviations, recombination of distinctiveness and responsiveness recast a plan driven project as a loosely-coupled system (Hällgren and Soderholm 2010). In the face of fluctuations, plan-driven teams supplement boundary spanning with off-plan responses such as brainstorming and assembly of tactical sub-teams with specific objectives related to the deviation. PM literature is flush with examples of firefighting (Barber and Warn 2005), fast moving tiger teams (Pavlak 2004), cheetah teams (Engwall and Svensson 2004) and flash teams (Tannenbaum et al. 2012). Brainstorming, whether individually, separately, collectively or electronically, has been shown to be an effective idea-generating device (DeRosa et al. 2007; Lamm and Trommsdorff 1973) that increases the variety of candidate solutions. As teams depart from the predefined script with situated action they supplement their membership with unplanned expert consultation (McChesney and Gallagher 2004). Either by fine-tuning or transformation, revising plans is a fundamental response to unexpected challenges with project managers "empowered to make decisions necessary to keep the project on track, even though these decisions may require complete re-shuffling of resources within the project" (Soderholm 2008). In various forms, these tactics represents loose coupling of the project from the predictable path of a preexisting plan. When enacting these behaviors in response to uncertainty and unexpected challenges, plan-driven projects are demonstrating classic self-organizing characteristics whereby people empower themselves and form groups spontaneously around issues (Stacey 1993). These teams are colluding to operate a "mock bureaucracy" that pays lip service to the rules but tacitly agrees not to enforce them (Stacey 1993). The plan-driven projects in this case study embraced behaviors and tactics of this sort. Project C engaged in brainstorming with a cross-functional group assembled to address performance related infrastructure problems, as well as forming a special task force to address a use case that stubbornly resisted standardization. Project D formed a tiger team to resolve the data integrity and inventory problems. Project D also augmented the on-site project team with a return visit of the training expert that was not in the original plan. Some practitioners contend that we will never see a software project that proceeds in a fully rational top-down manner (Parnas and Celments 1986). Instead, real world projects operating under the control of plan-driven methods engage in "improvisation, opportunism, interruption and mutual negotiation as much as by progress milestones, planning and management control" (Nandhakumar and Avison 1999). While most projects do not plan for this flexibility, they frequently use flexible approaches anyway (Olsson 2006). This response which occurs when existing plans fail to address novel conditions has been labeled “improvisation” (Leybourne 2010; Pavlou and El Sawy 2010).

Agilists react differently to challenges. When faced with fluctuations, the natural response of a loosely coupled system is to reorganize around the problem and experiment with options until a satisfactory solution emerges. The self-organizing and requisite variety characteristics of loosely coupled systems are evident in agile philosophy and practice (Vidgen and Wang 2009). In addition, agile teams attempt to remove distractions and accelerate their creative activities with experimentation – to innovate their way to a solution.

A consequence of agile’s concertive controls and preference for autonomy is a tendency to become isolated. Scholars note it is the ScrumMaster who “protects the developers from interference” (Batra et al. 2010). In what may seem to be an extreme practice of isolation, one study described a ScrumMaster’s efforts to isolate the team in response to challenges and time pressure – to the point “the team was not even allowed to receive phone calls” (Moe et al. 2009). These are not singular occurrences. Agilists describing an ideal team workspace recommend actions such as “isolate the team from off-topic noise,” and “minimize distractions” by allowing “no phones,” “no email or IM” while “executives stay muted” (Pietri 2009). The casual nature with which advocates prescribe isolation suggests the preference is strong and tied to the aforementioned plan and documentation bias:

> "Scrum organizes developers (the "pigs") into tightly knit teams who communicate daily with one another on a face-to-face basis. Communication with non-team members (the "chickens") tends to be limited. Indeed, one of the Scrum roles is tasked with shielding the team members from outside distractions -- like, for example, technical writers jumping up and down outside the team room door trying to get information."
> (Hughes 2009)

Project B reported less extreme, but similar tendencies. "*I would say that one of the biggest problems running projects like this is that often developers like to work alone, and they don't necessarily want to reach out to others and go discuss their problems. They want to solve it on their own and they're not going to necessarily consider everything.*" The project manager in this project arranged physical isolation to go along with behavioral isolation: "*One of the things I did was secure us a new workspace where the entire team would sit together. We used to be spread out all over the place, so that ironically, one of the things I did later to get our team to bond and learn from each other and work together closely, if we had done it earlier we wouldn't have had that interaction with the developer who helped us. So we now have this team space on a new floor where the entire team sits together. They're all within ear-shot, they can all talk to each other and have stand-ups in our team space*." According to the IT manager, multiple mechanisms were employed to buffer the team. "*We clarify the reporting line significantly and we isolated some, dedicated some of the team so they could actually work and concentrate on what we're doing and have 100% bandwidth to what they were trying to do… We isolated a number of developers from many of the meetings since they didn't have to be there*." Agile team isolation has been attributed to emphasis on communication rituals within the team at the expense of inter-team interaction (Chau and

Maurer 2004; Elshamy and Elssamadisy 2006; Whitworth 2008) that also isolates the team from stakeholders (Pikkarainen et al. 2008; Pikkarainen et al. 2012). Studies contrasting plan-driven and agile teams support this finding that agile teams have a tendency to become more isolated (Karlstrom and Runeson 2005, 2006).

Agile teams operate at the edge of chaos that is difficult to maintain. Sustaining agility is contingent upon maintaining a balance of interdependent principles and behaviors (Maruping et al. 2009; Vidgen and Wang 2009). Groups that over respond and loose elements of structural influence provided by boundary spanning behaviors such as customer collaboration, risk crossing the tipping point into chaos (Kautz and Zumpe 2008; Wang and Vidgen 2007). “Agile methods are more effective when paired with the exercise of outcome, rather than self control” (Maruping et al. 2009). Communication rituals and jargon that reinforce distinctiveness without supporting ties to the organization may weaken concertive controls to the point the team enters the region of ad hoc behaviors where outcomes are dependent on self control.

This shift in PM style is an important insight. Not only do organizations regularly mix and match practices from both the plan-driven and agile traditions to establish a locally dominant paradigm, but project teams faced with unexpected events can and do improvise and transform their PM style during a project. In so doing these project teams demonstrate requisite variety far in excess of that associated with their initial PM style.

Project management style that drives planning, documentation and communication has a moderating effect on a team's response to fluctuations. PM biases are associated with patterns of responsiveness and distinctiveness that reflect the fundamental levers of a team's reaction to deviations. While baseline style and routine processes provide the structure for teams managing general fluctuations, high intensity fluctuations may unfreeze routine behaviors leading to improvisation and transformation of PM style. The resulting shift depicted in Figure 3-2 allows project team behaviors to slide in the direction of decoupled systems, with an associated loss of bureaucratic controls. This may be good news where tightly coupled team behaviors avail themselves

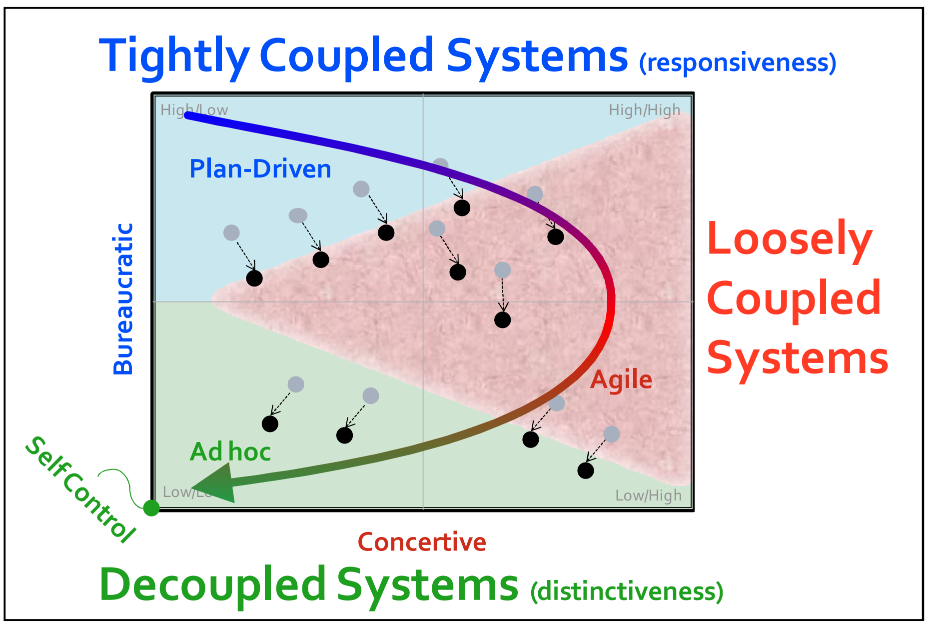


Figure 3-2 Reactive Decoupling

to the adaptable advantages of loosely coupled systems. When the shift is particularly dramatic, teams may become fully decoupled, with the loss of both bureaucratic and concertive controls. In environments of highly turbulent fluctuations an IT project team can become an adhocracy, dependent upon self-control to deliver organizational value.

The concept of Reactive Decoupling is new and introduced in this study to explain the response of IT project teams faced with unexpected disruptions. The ideas depicted in Figure 3-2 summarize a proposed theory of Reactive Decoupling. These ideas are formalized as a series of specific hypotheses suitable for empirical testing in the next chapter.

Chapter 4

Constructs, Hypotheses and Research Models

4.1 Constructs and Hypotheses

Unexpected challenges may emerge from all elements of a project, represented by the Work Systems Framework (Alter 2006). These events have a direct impact on project outcomes. The case study suggests the area of origin, or *locus of flux*, may amplify this effect. Issues emerging from outside the core project team will have a greater impact than those originating within the team. Project Management style will also moderate the impact of certain challenges as the baseline practices engage with and attempt to manage unexpected challenges. Plan-driven projects have a tendency to behave more as tightly coupled systems, emphasizing the bureaucratic controls of planning and leveraging documentation artifacts as boundary-spanning objects. Agile teams have a tendency to behave more as loosely coupled systems, emphasizing self-organizing autonomy and informal communication guided by concertive controls. The different behavior patterns provide a baseline moderation influence on unexpected challenges. In addition, some forms of fluctuations, perhaps the more intense and salient disruptions, induce improvisation and a shift in PM style. What follows is a description of each construct and the hypothesized relationships.

4.1.1 Project Outcomes

Project success is widely proclaimed as a multidimensional idea in both IS literature (Aladwani 2002; Petter et al. 2008) and PM literature (DeWitt 1988; Muller and Turner 2007; Pinto and Slevin 1988; Wateridge 1998). An assessment of PM literature reveals a maturation of project success criteria converging on several distinct dimensions: (1) the iron triangle of time, cost, and quality; (2) satisfaction across end-users, stakeholder and project personnel; and (3) strategic organization benefit (Ika 2009). This resonates with organizational and human resource literature that presents a multi-dimensional balanced scorecard to assess outcomes across four domains: Financial, Internal Business, Customer and Learning (Kaplan and Norton 1992; Milis and Mercken 2004). These dimensions capture the success of the project management process in terms of its efficiency, as well as the subjective and subtle concerns that encompass both short and long term effectiveness and value of the project from multiple perspectives.
Figure 4-1 details a balanced scorecard tuned for IT project teams.

The Financial Scorecard and the Internal Business Scorecard relate to the performance of the process of executing the project. This assessment of project management performance often revolves around of the “iron triangle” of Time (schedule), Cost (resources) and Scope (functionality, quality) (Jugdev and Muller 2005). These objective measures relate to standards of performance identified in the early stages of a project.

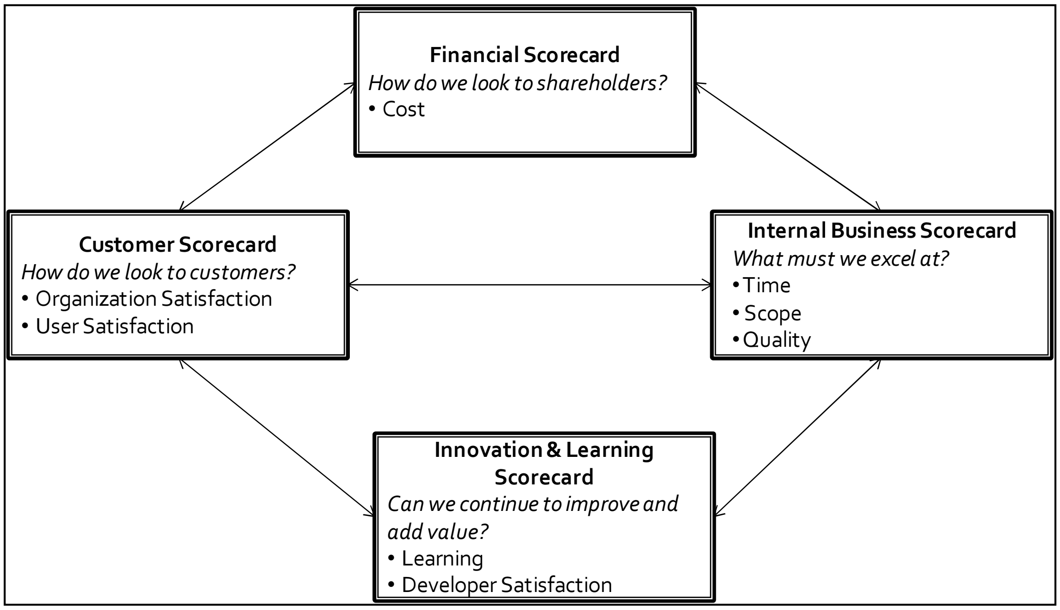


Figure 4-1 Balanced scorecard of IT project outcomes

Qualitative aspects of project success are captured in measures of satisfaction. Three distinct communities (users, management and developers) may harbor unique interests and dimensions of satisfaction (Shenhar et al. 2001). The users are focused on the immediate work system that the project output will be applied to. Their satisfaction is colored by perceptions of ease of use, utility and the degree to which the IT will help them perform their role. In some circles *User Satisfaction* is considered the most important criterion for measuring the success of IS, and perhaps the most prevalent as well (Zviran and Erlich 2003). *Sponsors Satisfaction* captures the perspective of management from IT and other functional group management with an interest in the project. This vantage point reflects the overall organization's interest and combines ideas such as return on investment, expected future value, and the opportunity cost associated with performing the project. *Member Satisfaction* represents the opinion of core team members who apply their talents and efforts to create the project output. These opinions reflect not only elements of project success, but also influence the project team as members perceive and anticipate these opinions.

Beyond the direct value of an individual project, there is a long term strategic value of expanding organization capabilities. Learning occurs around the technologies used within the computer system, as well as the process of performing the project. Improved knowledge, skills and abilities accumulate among the individual participants and the broader organization that matures local conventions, spreads best practices and replicates successful strategies and tactics. Team coordination capabilities mature, allowing an in-tact team to perform a subsequent project with greater efficiency, which may in turn be further enhanced by stronger collective efficacy. Growing organizational capabilities create long-term value that goes beyond the computer system created by a single project.

4.1.2 Fluctuations (Flux)

Different types of interruptions have been shown to impact processes and outcomes differently. For example, technology related interruptions have a negative influence on knowledge transfer, whereas changes in team structure can positively influence knowledge acquisition (Zellmer-Bruhn 2003). Three forms of flux were identified in the case study as particularly prevalent and therefore more relevant for study. Based on the frequency and relative importance this study uses staff, technology and requirements fluctuations as prominent indicators of the general concept of Flux that impacts IT project performance.

4.1.2.1 Staff/Resource Flux (SF)

Staff and resource fluctuations involve both changes to the project team membership as well as fluctuation in the attention of participants. The most obvious manifestation is the departure of a team member. New members may also be added, either in combination with a departure, or as the team expands in size. A less apparent form of staff flux occurs when one or more members are temporarily absent. This may be a transient absence for personal reasons, such as illness or planned vacation, or recurring, such as a member splitting time between multiple project assignments. The duration and frequency of an absence magnifies the impact on affected projects, along with the individual's centrality and the criticality of specialty knowledge, skills and abilities. Staff fluctuations necessitate rebuilding team mental models related to project goals, objectives and strategies as well as the intricate details of system design, data models and technologies. While the general category of staff fluctuation includes the possibility that a less capable team member is replaced with a more capable team member, on the aggregate this would be balanced by the opposite. While the change in staff capability may net to no effect, the disruptive impact of having to reestablish mental models, coordination schemes, and team self-efficacy present significant challenges to group processes (van der Vegt et al. 2010; Wageman et al. 2012).

4.1.2.2 Technology Flux (TF)

Technology instability in organizational literature is generally tied to the rate of change of technology for an industry at the firm level. For a project team, the rate of change associated with industry trends is relevant, but not acutely so. Short term tactical problems with technology central to the specific system being implemented are more salient. This includes software bugs (either emergent from team tasks, or imposed by providers outside the team), infrastructure service lapses, data issues, connectivity challenges and compatibility problems that the project must resolve before completing an implementation. Technology fluctuations impose uncertainty on the project team. The solution may depend upon the team acquiring new information, revisiting previous decisions, and applying increased effort to creating new solutions. Increasing the effort required to achieve an objective has a direct negative impact on process performance. Tatikonda and Rosenthal (2000) showed that technology novelty and complexity are negatively associated with new product development project success, suggesting the same can be expected for IS implementation projects.

4.1.2.3 Requirements Flux (RF)

Shared team mental models about goals and timing motivate the entire collective problem solving, decision making and intellective process (McComb 2008). Changes to requirements and scope are therefore central to IT project outcomes. Requirements fluctuations must be recognized by the team in a timely manner and accommodated. Accommodation requires the team to apply new effort to understand and adjust the system being implemented. New mental models are needed across members that will participate in addressing new requirements. Effective response involves altered designs, different decisions and new solutions. The direct effect on the project management process efficiency is expected to be negative.

Overall these project fluctuations, labeled Flux in this study, represent disruptions and a challenge to a project that would otherwise proceed with a routine project cadence. The disruptive consequence of fluctuations is a negative influence on performance. Therefore:

H1: Flux has a negative relationship with Performance (PERF)

4.1.3 Project Management Style

Teams mix and match practices from both plan-driven and agile methods to form a baseline process during project launch. This baseline style is characterized by its location within the Project Management Style Framework (Figure 2-1, page 33). As unexpected changes emerge and begin to have their impact on a project's outcomes, the baseline PM style engages to mitigate that impact. Normal behaviors position project teams to be relatively well equipped or poorly equipped to handle certain types of unexpected challenges. In all its forms, project management has been used to plan for, implement and meet change for centuries (Cleland and Ireland 2006). This relationship is formalized in the following hypotheses:

H2: Plan-driven style (PDS) has a positive relationship with Performance (PERF)

H3: Agile style (AS) has a positive relationship with Performance (PERF)

The tactics employed by plan-driven project teams reveal a tendency toward tightly-coupled systems behavior. They focus on executing according to defined plans and interacting across the team boundary using formal documentation at scheduled meetings with set agendas. Behaviors associated with this bias include boundary spanning that serves as an intervening variable between environmental characteristics and the processes and functioning of work groups (Leifer and Delbecq 1978). These characteristics of *responsiveness* are a hallmark of plan-driven projects faced with fluctuations. Plan-driven teams employ several tactics that are helpful in dealing with fluctuations. The most common mitigation is to plan for challenges by including contingency reserves for time, money and resources (PMBOK 2008). This risk management tactic involves adding buffers or slack in the schedule and planning for increased resources late in the project to accommodate rework (De Meyer et al. 2002). Other actions employed by teams adhering to a structured schedule and plan include reducing scope, conducting more tests and using documentation artifacts such as risk registers to learn from other projects. Many plan-driven teams add customers to the team or establish cross-functional teams to include members with limited but specialized skills and knowledge (Graham 2000), tactics that facilitate boundary spanning communications when fluctuations emerge. These tactics introduced through risk management planning enable a plan-driven project to manage fluctuations better than had no mitigations been arranged. Therefore:

H4: Plan-driven style (PDS) has a moderating effect on Flux that attenuates the negative relationship to Performance (PERF).

Agile teams have a tendency to function as loosely-coupled systems, emphasizing informal face-to-face communication and reorganizing around a new set of tactical objectives on a regular basis (typically at the end of each two- to four-week sprint). *Distinctiveness* provides the requisite variety and autonomy that facilitates innovation needed when addressing dynamic challenges (Hansen 1999). Loose coupling has been associated with adaptive capacity at the firm level (Boons and Berends 2001), among collocated product development teams (Hinds and McGrath 2006), as well as individuals addressing problem-solving contexts (Chen et al. 2011). Therefore:

H5: Agile style (AS) has a moderating effect on Flux that attenuates the negative relationship to Performance (PERF).

The combination of distinctiveness and responsiveness provides a potential advantage for loosely coupled systems over tightly coupled systems that lack distinctiveness, or decoupled systems that lack responsiveness (Orton and Weick 1990). In addition to maintaining boundary-spanning behaviors associated with responsiveness, project teams operating as loosely coupled systems with the self directed autonomy associated with distinctiveness exhibit greater requisite variety than tightly-coupled systems following a predefined plan. As a result, the effect size of agile style is expected to be larger than the effect size of plan-driven style.

The previous two hypotheses conceptualize a broad concept of flux that captures both multiple sources of fluctuations and a range of intensity from moderate to extreme. The intensity of disruption to a social system may also be associated with a different type of response. For example, at the level of a firm, moderately turbulent environments engage baseline dynamic capabilities whereas highly turbulent environments are associated with improvisational responses (Pavlou and El Sawy 2010). While both plan-driven projects and agile projects may mitigate the negative effects of moderate intensity fluctuations, the concurrent distinctiveness and responsiveness characteristic of loosely-coupled systems provides an advantage for agile projects faced with high intensity flux. Therefore

H6: Plan-driven style (PDS) has a moderating effect on high intensity Flux (HIF) that amplifies the negative relationship to Performance (PERF).

H7: Agile style (AS) has a moderating effect on high intensity Flux (HIF) that attenuates the negative relationship to Performance (PERF).

The relationship between unexpected challenges and project outcomes is an interaction effect contingent on both Flux Intensity and Project Management style. Project teams operating as loosely coupled systems can exhibit greater requisite variety than tightly coupled systems following a predefined plan. “Rare and relatively less fit competencies, which would normally be weeded out in tightly coupled systems, can survive and contribute to the pool of possible solutions. Loose coupling thus raises the likelihood that the system possesses the competencies needed to adapt to new conditions, albeit at the cost of relative system inefficiency when these conditions to not exist (Staber and Sydow 2002)."

4.1.4 Locus of Flux (LoF)

Unexpected challenges may emerge from within the project team, or from outside the project team. This holds for each of the dimensions of Flux this study examines. Requirements and scope may fluctuate as users and their representatives collaborating with the core team alter their guidance about features and functionality as the project unfolds. Scope change also may be triggered by market conditions that necessitate a reaction from the project. Additionally, requirement fluctuations may be instigated by the organization as management expands the domain of users and/or business processes. This occurred in Project B when management expanded the scope from a single site to four sites. A similar dynamic exists for staff fluctuations that may be instigated from outside the project team by management. The locus of initiation (internal or external) of team membership change has been shown to have significant impact on teams (Arrow and McGrath 1993). This occurred in both Project A and Project D as the project manager was removed and replaced. Staff fluctuations alternately originate from within a project team, as occurred in Project B when one of the team members was not compatible with the agile project processes and a mutual decision was made to remove that member. Finally, technology fluctuations are common with computer system projects in the form of logic bugs and data complications created by actions of the project team. This type of challenge is largely responsible for the development of mature IT departments with experienced professionals to staff IT projects. Alternately, technology challenges may originate outside the project team, such as infrastructure surprises in Project A and Project C, and the instability of externally supplied software libraries in Project A and Project B.

The locus of Flux is relevant to project teams focused on a task with an agenda. The focus of attention, data gathering, discussion, problem solving and creative exploration are centered on the primary task. When issues arise within the focal activities of the team, these disruptions can be interpreted and understood in the context of the active memory models. The team must adapt their activities to account for the fluctuation. When challenges arise from beyond the arena of focused attention, the team is faced with multiple challenges that start with recognizing the issue. Because attention is focused elsewhere there may be a delay in appreciating the implications of disruptive information. The boundary of cognitive inertia is plausibly greater for issues arising outside the core team than for those arising within. Once the unexpected challenge is recognized, the team must shift a portion of its energies to building a new memory model that assimilates deviations into the solution. The implications of cognitive inertia lead to the following:

H8: External Locus of Flux (LoF) has a moderating effect on Flux that amplifies the negative relationship to Performance (PERF).

4.1.5 Reactive Decoupling (RD)

Team responses to fluctuations observed in the case study fall into three general categories: (1) more of the same, where the team continues to apply normal methods, plans, procedures and techniques; (2) task transformation, where the team lengthens the project by extending timelines, alters objectives or changes the design (e.g., adding or removing functionality); and (3) team transformation, where the team alters its makeup or changes processes and behaviors in an important way. Disruptions can trigger transformation of either the team's makeup (e.g., its membership or role and responsibility assignments) or its behaviors (e.g., deviation from plans or altering processes and routines).

A particularly interesting response involves improvisation that propels a shift in project management style across the loose coupling landscape in the direction of distinctiveness, greater autonomy and loosening of ties with those outside the project team. Tightly coupled plan-driven teams react to unexpected challenges with a combination of increased boundary spanning, a classic behavior of using strong inter-entity ties, but also spontaneously forming special sub-teams to perform actions that deviate from plan-driven intentions. In response to uncertainty, tightly coupled projects become loosely coupled systems exhibiting both responsiveness and distinctiveness (Hallgren and Soderholm 2010). Where IT project teams react with increased loose coupling behaviors, the capacity to handle unexpected challenges also improves. The team will improvise and reform itself to address the challenge with unplanned actions.

While pursuing tactical actions the project team may continue to pay lip service to the intentions of the established plan to sustain an illusion of bureaucratic control (Stacey 1993). This shift toward loosely coupled systems behavior brings with it a freedom to explore more and different creative solutions than is possible when strictly following a predefined plan. The result, whether conscious and intended or not, is a team with increased autonomy (Kreiner 1992) and adaptive capacity to resolve unexpected challenges more effectively.

Concertive control that helps create distinctiveness for agile teams simultaneously encourages buffers and boundaries between entities of a system. Workers in a concertive environment reinforced by interaction rituals often identify more closely with their team than with the organization, and the immersion deepens over time (Barker and Tompkins 1994). Identification with values and attitudes of the group are strengthened by emotional satisfaction and growing self efficacy tied to both the perceived successes of the team and participation in reinforcing social symbols (Ashforth and Mael 1989). As team cohesion grows, member interaction patterns mature with the introduction of communication short cuts that facilitate tacit coordination (Wittenbaum et al. 1998). These communication semantics are private to the group and facilitate in-group communications but amplify the social distance to out-group individuals, making these boundaries more difficult to traverse (Tushman and Scanlan 1981).

In times of uncertainty these inward facing behaviors become more extreme as the team focuses its energy on exploring creative and novel ideas in a trial and error approach to problem solving. The loosely coupled agile team may become overly homogenous, highly differentiated and "isolated from the rest of an organization" (Whitworth 2008). The identity and value reinforcing rituals of agile practices intensify collaborative work "at the cost of a somewhat closed relation to outside the team" (Mackenzie and Monk 2004). Studies contrasting project management approaches confirm agile teams become more isolated with reduced communication outside the team compared to plan-driven teams (Karlstrom and Runeson 2005, 2006; Pikkarainen et al. 2008). Furthermore, agile teams sometimes over-rely on informal communication, causing miscommunication (Ramesh et al. 2006). When faced with sufficiently great uncertainty from unexpected challenges, agile bias can intensify the distinctiveness and isolation of agile teams enough to shift them into the decoupled systems region. The difficulty of boundary spanning will be intensified at the very time when such behaviors are most needed. As suggested by the case study experiences, both plan-driven and agile project teams respond to fluctuations by embracing new practices and behaviors; they shift down the loose coupling flexure as depicted in Figure 3-2 (page 89).

The baseline project management style may often be a culturally imposed and reinforced set of behaviors. Some sort of impetus is necessary to disrupt the cognitive inertia associated with baseline PM style. Fluctuations that are unexpected and represent a shock to the project provide the trigger needed to unfreeze established routines and habits. Fluctuations generate pressure on projects regardless of PM style. Evidence for this is expected in a negative association between Flux and the baseline PM style. Therefore:

H9: Flux has a negative relationship with plan-driven style (PDS).

H10: Flux has a negative relationship with agile style (AS).

The proposed theory of Reactive Decoupling predicts that shifts tend to move along the loose coupling flexure in the direction of decoupled systems. The pressures associated with this shift materialize in the degree to which project teams deemphasize responsiveness behaviors and emphasize distinctiveness behaviors. While projects try to manage moderate fluctuations with normal behaviors, high intensity fluctuations are more likely to trigger a shift. This relationship is formalized as follows:

H11: High intensity Flux has a negative relationship with responsiveness (RDresp).

H12: High intensity Flux has a positive relationship with distinctiveness (RDdist).

Plan-driven style provides certain advantages managing change, such as boundary spanning and maintained alignment with management. Reactive Decoupling suggests that shifts will tend to follow the loose coupling flexure in the direction of decoupled systems behavior. Forces that continue to emphasize the beneficial behaviors of distinctiveness help limit the distance of that shift, and will help preserve associated performance rewards. Agile style provides certain advantages managing change, such as creative problem solving and autonomous teams that rapidly reform around a new challenge. Forces such as Reactive Decoupling distinctiveness that shift plan-driven projects into the region of loosely coupled systems could create fluctuation handling advantages for those projects. These same forces applied to an Agile project team operating at the "edge of chaos" may be driven from the loosely coupled systems region into the decoupled systems region of ad hoc project behaviors. Forces that are advantageous for plan-driven projects may have negative consequences for agile projects. Therefore:

H13: Reactive Decoupling - responsiveness (RDresp) has a moderating effect on plan-driven PM style (PDS) that amplifies the positive relationship to Performance (PERF).

H14: Reactive Decoupling - distinctiveness (RDdist) has a moderating effect on plan-driven PM style (PDS) that amplifies the positive relationship to Performance (PERF).

H15: Reactive Decoupling - responsiveness (RDresp) has a moderating effect on agile PM style (AS) that amplifies the positive relationship to Performance (PERF).

H16: Reactive Decoupling - distinctiveness (RDdist) has a moderating effect on agile PM style (AS) that attenuates the positive relationship to Performance (PERF).

4.2 Research Models

Collectively, these hypotheses represent a holistic conceptualization of the proposed theory of Reactive Decoupling depicted in Figure 3-2 (page 89). These propositions are re-characterized in terms of two research models. The first model uses the direct effects of Flux on Performance to explore the contingent effect of PM style and locus of Flux. This research model characterizes the value of PM style associated with its positioning on the PM Style Framework (Figure 2-1, page 33). A generalized depiction of this model is provided in Figure 4-2.

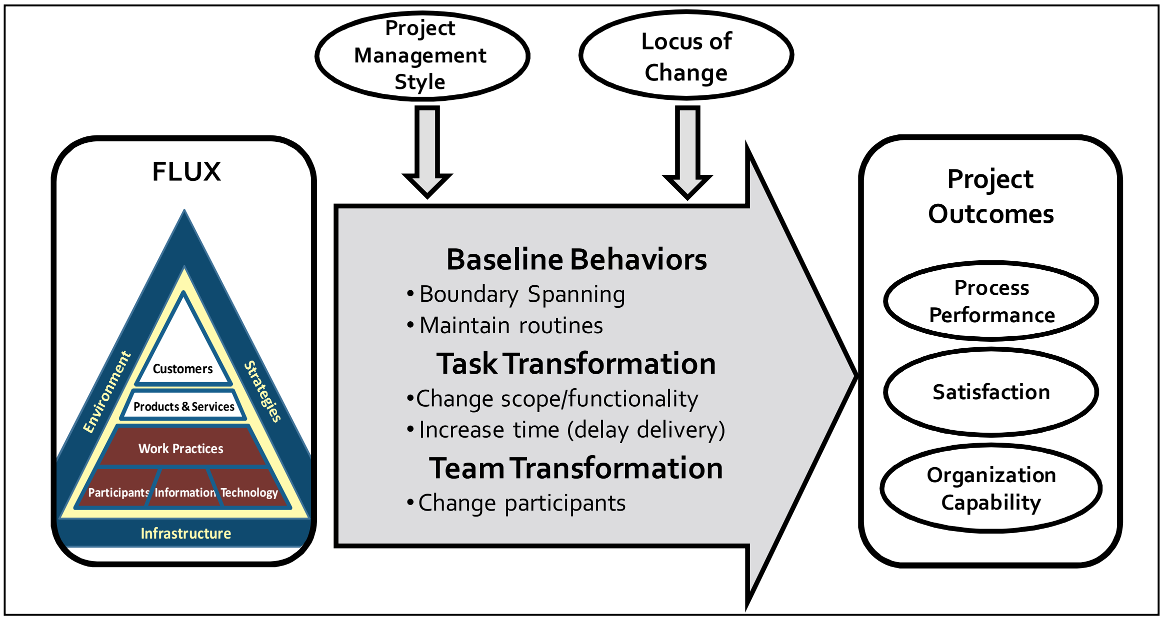


Figure 4-2 Contingent effects and PM style

The hypotheses that involve contingent effects of PM style are captured in the variance style research model shown in Figure 4-3. The research model simplifies the theoretical model in two important ways. First is a focus on three forms of fluctuations: Staff Flux, Technology Flux and Requirements Flux. Instead of measuring fluctuations along all nine components of the Work Systems Framework, these areas of Flux suggested by the case study to be most frequent for IT projects have been selected. Flux intensity in this model is a function of these forms of Flux that emphasizes fluctuations perceived to be most significant. The interaction effect of Flux Intensity foreshadows the phenomenon of Reactive Decoupling explicitly operationalized in the second model.

Locus of Flux adds texture to the phenomenon of Flux by examining the amplifying effect of challenges originating outside the core project team. The second simplification involves *process performance* as the focal project outcome in this study. This abbreviated dimension of performance defers *organization capabilities* and *satisfaction* for future research. These steps remove conceptual clutter and provide a more concise model to introduce the proposed theory.

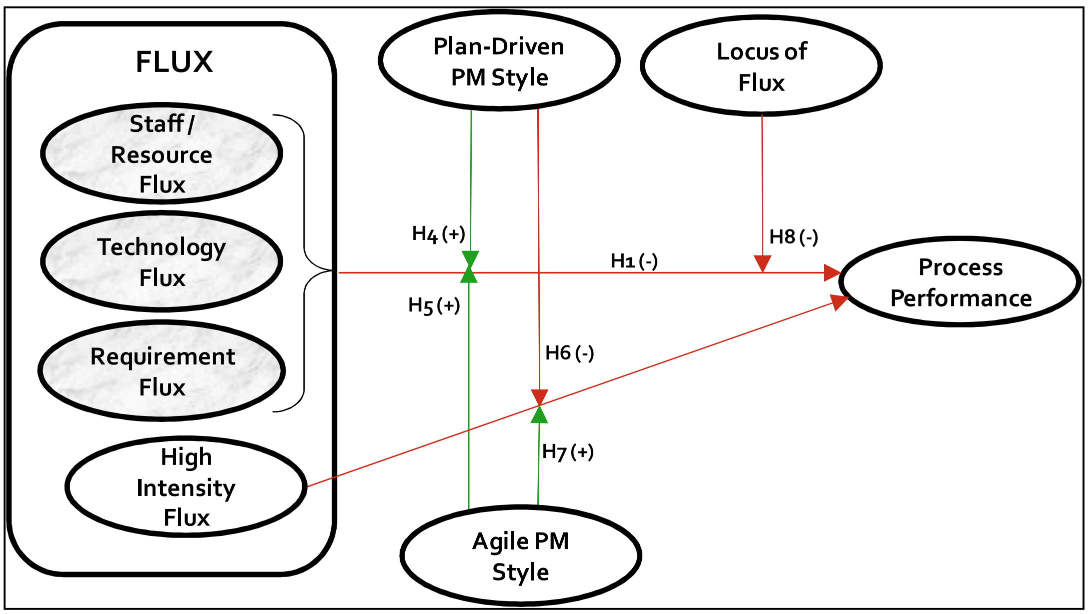


Figure 4-3 PM style research model

Independent measures of plan-driven style and agile style are consistent with the reality that project teams demonstrate ambidexterity as they mix and match practices from both traditions. This conceptualization allows varying degrees of plan-driven and agile style to be accommodated without presupposing rigid adherence to any particular methodology.

The second research model conceptualizes Reactive Decoupling directly to illustrate its capacity to shift PM style in an environment of Flux. This model, generalized in Figure 4-4, predicts performance implications of this shift.

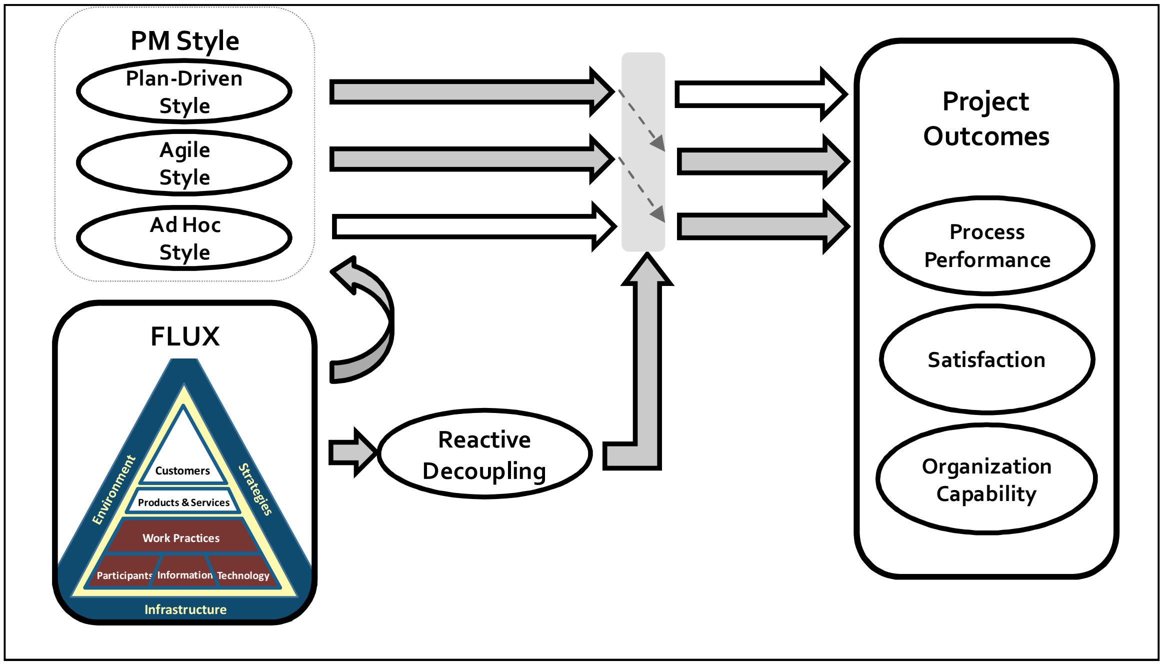


Figure 4-4 Flux and Reactive Decoupling

The hypotheses that apply to directly conceptualized Reactive Decoupling are captured in the variance style research model shown in Figure 4-5. As with the PM Style contingency model there is again simplification to three forms of fluctuations emphasized by the case study. Similarly, Process Performance is the focal Project Outcome. Instead of examining the contingent effects of PM style, this model casts PM style as a baseline direct effect available to the project at its inception. The ability of Flux to dislodge the baseline PM style and trigger a shift along the loose coupling flexure are revealed through the direct effect of Flux on PM Style and the compounding effect of Reactive

Decoupling. The dialectic properties of loosely coupled systems require distinct conceptualization of responsiveness and distinctiveness.

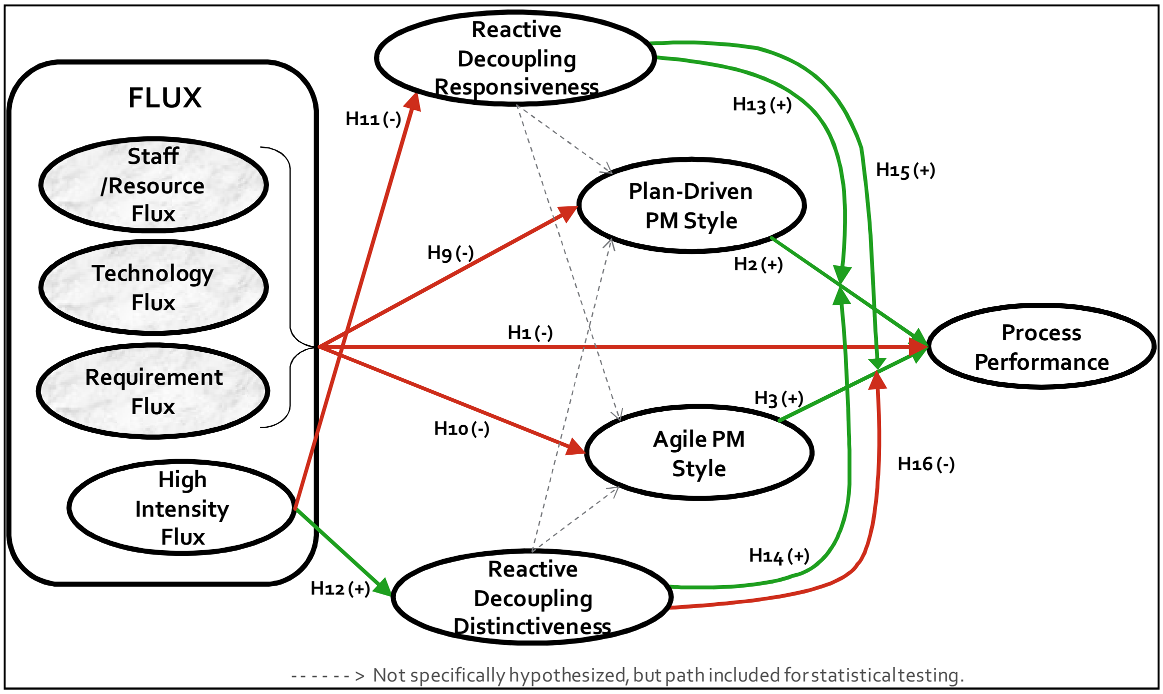


Figure 4-5 Direct observation research model

While the constructs suggest a balance of style and Reactive Decoupling, the hypothesized polarity of the paths is not. High intensity flux decreases *responsiveness* while at the same time encouraging *distinctiveness*. The unbalanced consequences nudge plan-driven Style to loosely coupled behavior with positive performance interaction effects. In contrast, emboldened *distinctiveness* can prod agile style into the decoupled region with negative performance interaction effects.

Chapter 5

Survey Study Design

This chapter describes a study design to provide confirmatory support for the PM Style framework and the theory of Reactive Decoupling. Motivated by a desire to complement the initial case study with a more generally representative population, the confirmatory phase of this study uses survey techniques and collects data from a broad set of IT projects. Survey designs are common in the literature examining IT projects (Caniels and Bakens 2012; Doke 1990;Faraj and Sproull 2000; Kirsch et al. 2010; MacCormack et al. 2001; Majchrzak et al. 2005; Moløkken-Østvoldand and Jørgensen 2005; Murphy and Ledwidth 2007; Stewart and Gosain 2006), and play a prominent role in rigorous multi-phase multi-method studies (Bresman 2010; Lee and Xia 2010; Yin 2009).

5.1 Study Design

5.1.1 Unit of Analysis

As in the case study, the unit of analysis remains the IT project team. During the case analysis a triangulation approach was used with similar information collected from multiple participants representing different roles in an effort to expose ideas with validity across perspectives. Moving to a much larger cross-sectional data collection effort, collecting multiple responses for each project is a significant challenge undermined by the desire to ensure anonymity and thereby obtain a larger sample with greater statistical power.

Research targeting team level emergent phenomenon such as group efficacy, collective cognition or group empowerment, benefit from aggregation of opinion across multiple members to form a composite. An implicit assumption of multi-respondent compositional strategies is membership stability (Higgins et al. 2012; Mathieu and Chen 2011). The case study suggests that team composition is a common source of fluctuation that serves to disrupt IT project teams. In an environment of inherent team instability, aggregating the responses of multiple members is not justified (Wagemen et al. 2012). However, phenomenon and events that are externally verifiable, such as the observable practices and behaviors in this study's research models, can be effectively reported by strategically positioned single key informants. As noted by Homburg et al. (2012), "Key informants are significantly more reliable for constructs that refer to the present, that point to observable verifiable referents, and that address salient events." There is a precedent to use project managers as key informants in IS literature (Chou and He 2011; Mahaney and Lederer 2003; Pavlou and ElSawy 2010; Stewart and Gosain 2006; Wang et al. 2008), PM literature (Bartsch et al. 2012; Hsu et al 2011), Operations Management literature (Chandrasekaran and Mishra 2012; Kirsch et al 2010; Saeed et al. 2005), Marketing literature (Sethi and Iqbal 2008) and Management/Psychology literature (Rijsdijk and van den Ende 2011; Rogelberg et al. 2006; Yang and Guy 2011). This is justified based on the unique position that project leaders have as the project's communication and coordination hub (Vidgen and Wang 2009; Wagner et al. 2010).

The decision to focus on the key informants from the single role of project manager helps preserve measurement equivalence across respondents (Rungtusanatham et al. 2008). Studies of situational issues that pool responses from multiple informants with varying demographics relative to the unit of analysis ignore the inherent differences among informant roles and may incorrectly measure the phenomenon. Efforts to assure homogenous demographics for key informants help preserve equivalence and remove error that could otherwise dilute measures of substantive phenomenon.

5.1.2 Scale Development and Pilot Survey

Survey items were selected from previous research where items coincided with the theoretical concepts involved in this study. In a few situations (described in more detail below), suitable existing scales were not available and new items were devised. In all cases questions were adapted through a multi-step process to refine the wording of questions for the final survey.

In order to minimize method bias a referent-shift approach (van Mierlo et al. 2009) is taken such that questions were worded to solicit a response on behalf of the target project. Respondents were not asked what they personally did, but rather to report on the overall project. This approach leverages the expert key informant aspect of the project manager role.

A draft version of the survey was administered to a small group of professional project managers identified at their place of work[5]. All pilot participants were over 35 years of age and experienced in the role of IT project manager (one participant self-identified with 1-2 years experience and one with over 10 years experience, all others between 3 and 4 years). In addition to completing the survey all pilot participants provided feedback on the interpretation of questions, wording that was potentially confusing, redundancy that was distracting and survey fatigue. Based on this feedback several questions were reworded to simplify vocabulary and reduce conceptual complexity; techniques that also help mitigate the effects of method bias (MacKenzie and Podsakoff 2012). In addition the total number of questions was reduced to decrease the time needed to complete the survey and thereby improve the response rate (Frohlich, 2002). Finally the sequence of questions was altered to start with the simple project outcome concepts, and build up through the more complex concepts of project management style and fluctuations. Once the terminology and context of PM style and fluctuations were established, the most complex questions involving Reactive Decoupling and Locus of Change were administered. Finally at the end of the survey, when fatigue presents its greatest risk, the survey concludes with the least cognitively demanding demographic questions. Questions designed to capture control variables (such as project size and geographic distribution) were distributed throughout the survey to allow a

---

[5] These project managers worked at "OphCo" from the case study, but were not involved in any of the projects examined in the case study.

mental break even as the overall complexity builds. The cadence that builds in complexity and then trails off at the end has been shown to facilitate retrospective recall and information retrieval to mitigate the confounding effects of survey method bias (MacKenzie and Podsakoff 2012).

Responses to all questions (unless otherwise noted) were measured on a six point continuous scale ranging from (1) Strongly Disagree to (6) Strongly Agree. Final question wording and descriptive statistics are provided in appendix B (Table B.1). As noted in the appendix, several questions were reverse coded to minimize repetitiveness that reduces respondent motivation and cognitive engagement (MacKenzie and Podsakoff 2012).

5.1.2.1 Performance (PERF)

The primary dependent variable for the research model is Process Performance (PERF). Questions were adapted from Chandrasekaran and Mishra (2012) to capture objective project performance criteria of time, cost, functionality and quality. The constituent criteria objectively measure the degree to which the project process is successful. Variants of these process performance items have been employed in other IS and PM studies to measure performance (Agarwal and Rathod 2006; Lee and Xia 2010). PERF is a reflective scale of four items. During assessment of constructs, item PERF3 related to functionality lacked measurement validity (loading on the latent construct was statistically insignificant at the $\alpha=0.05$ level[6]) and was removed from further analysis.

There are other dimensions of performance, such as organizational value and satisfaction that are not directly captured by this conceptualization of PERF. While organizational capability is not operationalized in this survey, additional questions were included to capture three categories of satisfaction (User Satisfaction, Stakeholder Satisfaction and Member Satisfaction). The satisfaction responses were not used in the primary hypothesis testing, but are available for post hoc examination. User Satisfaction is a three item adaptation of the reflective scale from Seddon and Yip (1992), which itself was built upon the work of Ives et al. (1983). Sponsor Satisfaction is a three-item adaptation of the reflective scale from Ferreira and Cohen (2008). Member Satisfaction is a three item adaptation of the reflective scale from Aladwani (2002). All satisfactions measures demonstrated good convergent and discriminant validity.

5.1.2.2 Fluctuations (FLUX)

Unexpected deviations, issues and challenges in a project represent fluctuations that prevent a project from progressing normally through its intended steps and processes. Such fluctuations are common in most all non-trivial projects. Three forms of fluctuations were identified in the case study as particularly frequent, and these are operationalized as Staff Flux (SF), Technology Flux (TF) and Requirements Flux (RF).

---

[6] Convergent Validity for the measurement model is tested using PLS concurrently with calculation of item weight values, path coefficients and model statistics. Item loadings revealed to be non-significant are removed and the model re-run with the remaining items. P-values for items dropped from analysis are reported in Appendix B. See section 5.2.1 for additional details.

Staff Flux is a four item reflective scale that combines a turnover question introduced by Gopal and Gosain (2010), a partial membership question introduced by Carbonell and Rodriguez (2006), a new question related to dynamic membership intensity inspired by Tannenbaum et al. (2012), and a new ombudsman question. Conceptually this scale captures staff fluctuations associated with the temporary or partial distraction and unavailability of members as well as the outright removal, replacement or addition of members. During assessment of constructs, item SF1 related to multiple work assignments lacked convergent validity (loading on the latent construct was statistically insignificant at the $\alpha=0.05$ level) and was removed from further analysis.

Technology Flux has typically been conceptualized as an organization level phenomenon involving emerging and evolving technologies across an industry as a whole. This study conceptualizes technology flux not at the organization level, but at the tactical project level. No suitable existing scales were identified in the literature. As a result, a new scale inspired by insights from the case study was developed to reflectively capture the effect of technology flux in terms of frequency, design changes and functionality compromises, along with an ombudsman question. All items demonstrated acceptable convergent and discriminant validity.

Requirements Flux is a four-item scale that combines two questions introduced by Wallace et al. (2004a) with a new reverse coded item and a new ombudsman question. Conceptually this scale captures the instability of requirements. IS projects have long been concerned about requirements change, so this concept is believed to be highly salient for the target respondents. All items demonstrated acceptable convergent and discriminant validity.

Each form of fluctuation is independent and may exist or not exist irrespective of the others. Due to this independence, a general concept of Flux is operationalized as a formative second order construct aggregating the first order factors of Staff Flux, Technology Flux and Requirements Flux. This construct serves as a proxy for the broad concept of Flux that includes fluctuations originating from any part of the project work systems.

5.1.2.3 High Intensity Flux (HIF)

Each of the Flux constructs includes an ombudsman item intended to capture the intensity of fluctuations. Where respondents "Agree" and "Strongly Agree" fluctuations are *highly significant*, a new factor of High Intensity Flux (HIF) can be calculated. This coded factor is used to assess the interaction effect of High Intensity Flux on PM style. This split group analysis approach follows a similar technique applied to survey research on team stability (Akgun and Lynn 2002a).

5.1.2.4 Locus of Flux (LoF)

Locus of Flux (LoF) is a construct that attempts to capture the within group or outside group origin of Flux. When fluctuations originate at the hands of the project team, members may perceive the disruption is something they have done to themselves. When fluctuations originate from outside the team, members may perceive the disruption is something imposed upon the team. Three reflective questions adapted from items introduced by Imamoglu and Gozlu (2008) serve as the core of this scale that is supplemented by a new item. A unique scale is applied to these questions capturing the "extent" to which the origin was internal or external: 1 through 6 starting with "Not At All" and progressing to "a Very Great Extent".

This scale suffered as these questions were perhaps the most complex in the survey. Two questions (LOF3 and LOF4) did not converge with the others (statistically insignificant loading at the α=0.05 level) and were removed from further analysis. A final two-item construct failed to demonstrate significant relationships where they were hypothesized, and in the end provided limited insight for this study. While the concept of Locus of Flux is analytically relevant and supported by the case study, additional investigation is appropriate to gain a better understanding of this phenomenon.

5.1.2.5 Project Management Style

Organizations mix and match practices from multiple methodologies to establish a baseline project management processes. Where these practices are drawn from plan-driven and agile style, the result will include associated biases that can be measured.

Plan-driven style (PDS) measures the presence of documentation and scheduling practices which are highly characteristic of plan-driven style. This scale is composed of six reflective items, two adapted from documentation structure questions introduced by Ceschi et al. (2005) and four adapted from planning structure questions introduced by Dvir et al. (1998). One question (PDS5) did not converge with the others (statistically insignificant loading at the α=0.05 level) and was removed from further analysis.

Agile style (AS) measures the presence of informal communication rituals and autonomous teams that reform around evolving self-determined tasks. This concept is operationalized with eight reflective questions, three adapted from Chandrasekaran and Mishra (2012), one adapted from Shenhar and Dvir (1996), and four substantially new items inspired by Lee and Xia (2010). One question (AB1) did not converge with the others (statistically insignificant loading at the α=0.05 level) and was removed from further analysis.

5.1.2.6 Reactive Decoupling (RD)

Operationalizing the concept of Reactive Decoupling draws upon the nature of loosely coupled systems and the dialectic involving both *responsiveness* and *distinctiveness*. Responsiveness and distinctiveness are analytically different concepts (Hallgren and Soderholm 2010; Orton and Weick 1990) and are indicative of the extremes on the loose coupling flexure in the PM Style framework (see Figure 2-3, page 89). Previous scales of loose coupling draw upon the boundary spanning ideas of interunit ties and formal vs. informal knowledge and documentation (Hansen 1999). Some scholars operationalized the phenomenon of loosely coupled systems with a single dimension, such as organizational linkages (Taylor and Helfat 2009), coordination mechanisms (McChesney and Gallagher 2004), strategy coupling through plan coordination (Mitchell and Zmud 1999) and Improvisation (Akgun and Lynn 2002b). An important insight from High et al. (2008) involves the difference between coupling which exists as a baseline phenomenon, and cohesion which measures the adaptive response and change in coupling that occurs in the face of stressor events. Another important advance in operationalizing comes from Akgun et al. (2012) who conceptualized loose coupling across with three constituent concepts: Autonomy, Management style and characteristics of Information Travel. The operationalization developed for this survey combines the influences from previous research that (1) establishes separate constructs for responsiveness and distinctiveness, (2) captures the domains of automony, management and contrasting information exchange biases, and (3) addresses the adaptive response to change induced by Flux.

Reactive Decoupling – responsiveness in the final survey is composed of five questions. After removing items with poor convergent validity[7], a reflective scale with two measures remains: one addressing the domain information exchange and boundary spanning (RD2), and another dealing with management style associated with conforming to plans (RD4). These questions emphasize a response expected to strengthen objectively verifiable bureaucratic controls common within plan-driven behaviors. Reactive Decoupling – distinctiveness is composed of four questions. After removing two items with poor convergent validity, a reflective scale with two measures remains: one addressing autonomy (RD7) and one addressing the domain of information exchange buffers (RD3). These questions emphasize a response expected to strengthen independence favored within agile behaviors. These questions are framed to capture the

[7] Three items have been dropped to correct Average Variance Extracted below 0.5 (RD1, RD6 and RD8). Two items have been dropped due to poor indicator reliability (RD5 and RD9). See section 5.2.1 for additional details.

adaptive behavior occurring in response to Flux. The result is a new survey measure for Reactive Decoupling.

5.1.2.7 Control Variables

Additional single indicator control variables are included in this study to capture the covariance associated with relevant factors that are not directly substantive to the proposed theory. Identification of these control variables was guided by existing studies of IT project management suggesting these constructs should be accounted for. The first is *team size* captured in question C1, has been included frequently in IT project studies (Bresman 2010; Kirsch et al. 2010; Majchrzak et al. 2005; Rai et al. 2009; Robert et al. 2008). Geographical dispersion of teams has become increasingly common and relevant to team behaviors and performance. Both *team member dispersion* (question C2) and *user dispersion* (question C3) have frequently appeared in IS literature (Majchrzak et al. 2005; Wakefield et al. 2008). The *duration* of a project (question C6) has been operationalized in existing studies as both team longevity (Bresman 2010; Deeter-Schmelz and Ramsey 2003) and project age (Stewart and Gosain 2006; Rai et al. 2009). Experience from multiple perspectives has commonly appeared in IT project studies, including operationalization as general experience (Lin et al. 2012), project leader experience (Rai et al. 2009) and tenure (Wakefield et al. 2008). *Experience* is operationalized in this study as years in the role of project manager (question Q4). Additionally, *age* (question D1) appears as a control in several IT project studies (Wakefield et al. 2008; Lin et al. 2012).

This study also included controls related to:

- technologies new to the organization (question C4)
- time since the project ended (question C6)
- percent of project duration the respondent was heavily involved (question Q1)
- the type of system implementation project (question C7)
- project teams within the IT industry (question D2 coded as 1 for computer or telecommunications hardware/software products and services, 0 for all other)

Only two control variables demonstrated a statistically significant relationship with performance: Duration (β = -0.067, f^2=0.01, p-value = 0.048) and IT Industry (β = 0.101, f^2=0.02, p-value = 0.021). Each control variable contributed 1% or less to the R^2, suggesting these factors do not justify significant attention. Only these two significant control variables remained (others removed) when testing research model hypothesis.

5.1.3 Survey Setting

The theoretical ideals presented in the PM Style Framework and the theory of Reactive Decoupling has applicability to many project types. The focus of this current study is IT projects, which are well represented by members of the Information Systems Community of Practice (IS-COP) within the Project Management Institute (PMI). PMI has over 400,000 members worldwide (Dickson 2013), of which over 9000 have self affiliated with the IS-COP. This is a convenient community to locate subjects from the target population of IS project managers. This community has the advantage of worldwide membership representing diverse levels of experience, cultural backgrounds, many industries and organizations of all types and sizes.

5.1.4 Data Collection

Several design mechanisms are used to secure responses from appropriate key informants. First is the choice of target audience from the PMI IS-COP. The membership was randomized and a subset of members was directly solicited through email to participate in this study. Second, respondents were asked to consider a single specific project in their response. Third, the survey includes filter questions to disqualify respondents who did not serve as the project manager of their target project, and those that lacked a deep understanding of their target project.

Email requests were delivered to target participants with an introduction to the study and a request they participate anonymously by responding to an online computer based survey. Reminders and multiple mailings have been demonstrated to improve the response rate (Frohlich 2002). In this study a reminder email was sent to non-respondents 10 days after the initial mailing and a final reminder after 25 days. From 3000 solicitations, 643 responses were received yielding a 21.4% response rate.

Respondents were asked to focus on a single project that had completed during the previous 6 months as the basis for their answers. This was done so that the project performance could be assessed from the vantage point of a completed project, but not so long ago that memories have faded. It is likely this criterion disclosed in the survey introduction may have deterred many potential respondents. Two filter questions further narrowed the qualifying responses to individuals serving in the role of project manager and individuals with a deep understanding of the target project. While the members of the IS COP may serve in various roles across multiple projects, these qualification questions remove responses that did not adequately represent expert key informants for the projects being addressed. The result is a sample of 268 qualified responses.

A power analysis for path modeling suggests that 271 respondents provide 80% power at the $\alpha=0.05$ level of significance to detect medium effect sizes for the number of latent factors and manifest measures involved in the target research models. This sample of 268 provides much less power for small effect sizes where the study risks not finding statistical support for relationships that actually exist (Type II error of false negatives).

5.1.5 Methods for Data Analysis

The survey instrument collects a multivariate dataset in which the substantive constructs are each represented by multiple questions. Three methods are commonly used to aggregate and assess survey data sets of this type. The first method is to generate a linear composite of items associated with each substantive construct followed by linear regression. The second method is a covariance based structural equation modeling (CB-SEM) approach that simultaneously calculates latent construct values, accounts for measurement error and solves the path relationships between constructs using an iterative estimation approach to minimize the residual variance of all observed variables. The third method is Partial Least Squares structural equation modeling (PLS-SEM). This method calculates weighted composites of latent factors then employs an iterative linear regression to minimize the residual variance of all dependent variables in the model.

A comparison of these candidate techniques for complex models performed by Reinartz et al. (2009) concludes that "PLS should be the method of choice for all situations in which the number of observations is lower than 250 (400 observations in the case of less reliable measurement models, i.e., low loadings and/or few indicators)" (p342). In addition, IS literature recommends PLS-SEM as a better choice when "complex models" are involved (Ringle et al. 2012). This study involves several characteristics that align with this recommendation. First, latent constructs in this study were measured with relatively few questions. Some surveys use scales with eight to twelve questions per latent factor, whereas the constructs in this study involve far fewer items. While sufficient to calculate latent constructs, this is appropriately characterized as "few indicators." Second, some items have standardized factor loadings between 0.4 to 0.6, which is acceptable by common guidance in the organizational behavior and IS fields, but appropriately characterized as "low." Finally, the model involves multiple moderation (interaction) relationships, which by all accounts classifies it as complex. As a result, the overall hypotheses and research models are evaluated using PLS-SEM. Where other techniques and tools have been used to assess specific aspects of the data this is noted along with the results.

5.2 Validity Assessment

After collecting data, various tests are applied to assess its appropriate use in evaluating the proposed research models. These assessments are primarily concerned with validating the measurement instrument and establishing assumptions relevant for

PLS-SEM analysis. With PLS methods the factor analysis of measures is calculated concurrently with the path model of hypothesized relationships (Geffen et al. 2000). After corrections have been made (such as removing items with insignificant indicator reliability or with excessive cross-loading) the model is rerun in order to obtain new results (Urbach and Ahlemann 2010). Statistics reported in Appendix B are from an intermediate model used to disqualify certain measures or from a final model for items included in hypothesis testing.

5.2.1 Construct Validity

Construct validity is a concern that the operationalized measures accurately capture the intended construct. The logical validity of survey questions were assessed during the pilot study where professional project managers provided feedback on their interpretation of constructs, and identified specific questions and wording that undermined purported meaning. Additional insight of validity is offered in statistical tests for convergent validity and discriminant validity common in the IS field.

Convergent validity assesses the degree to which measures of constructs that theoretically should be related are in fact related. Several statistics are applicable with thresholds for IS research using PLS provided by Urbach and Ahlemann (2010).

- Indicator reliability is validated by the item loadings (identified as “Load” in Appendix B). Loadings above 0.7 are considered good, with a minimal acceptable threshold of 0.4. In all cases the values should be significant at the α=0.05 level.

Some survey items failed to meet these criteria and were excluded from further analysis (AS1, LOC3, LOC4, MK1, PERF3, PDS5, RD5, RD9, SF1).

- Unidimensionality is verified where there are no (zero) cross-loading items. For this dataset, all items had greatest correlation with the construct they were intended to measure.
- A measure of internal consistency and reliability is provided by the Composite Reliability metric for the weighted composite latent factor. This value should be greater than 0.7. Each latent factor demonstrated this criterion[8].
- Average Variance Extracted (AVE) represents the amount of variance due to the target construct as compared to the amount due to measurement error. AVE greater than 0.5 is recommended. After removing items that compromise convergent validity (RD1, RD6 and RD8), remaining latent factors met this criterion.

Discriminant validity assesses the degree to which measures that theoretically should be unrelated are in fact unrelated. While the theoretical model predicts certain correlations between distinct constructs (for example, Staff Flux is expected to be negatively correlated with Process Performance), these correlations between distinct factors should not be overly strong. In particular they should be distinctly lower than the items intended to measure the latent construct itself.

---

[8] Chronback's Alpha is a common reliability indicator for regression analysis. However the assumptions of this method are violated by the unequal weighting applied to measures by PLS where the Composite Reliability indicator is preferred. Composite Reliability calculations in SmartPLS suffer from two programming bugs. First the calculation is based on observed factor loading, not standardized factor loadings. Second, SmartPLS includes a minus sign for reverse coded measures, which is not accounted for in the calculation of CR, generating in some wildly deflated values. Results reported in Appendix B were hand-calculated using the absolute value of standardized factor loading and standard errors as determined by bootstrap re-sampling.

- Cross Loadings reveal where individual measures correlate too strongly with factors they are not intended to measure. No items for an off-variable item should load higher than the latent variable's weakest measure. All reflective latent factors meet this criterion.
- AVE must be larger than inter-construct correlations. An additional rule of thumb is that the square root of AVE for each latent variable should be greater than the latent variables highest correlation with any other latent variable. All reflective latent factors meet this criterion.

5.2.2 Self-select and Non-Response Bias

Another threat to survey studies is response bias. Many Internet surveys are posted on web sites and depend upon respondents happening across the survey and choosing to participate – they self-select. The concern is that a certain type of person who visits the site with a survey link and chooses to participate may not be representative of the target population (perhaps these are people with an axe to grind or a personal agenda unrelated to the study). This study mitigated this bias through random selection of target respondents and directly soliciting their participation through email. These individuals did not choose themselves, but were selected at random by the study design. Response rates of 20% to 30% (Frohlich 2002; Lee and Xia 2010; Murphy and Ledwith 2007) are desirable for this type of technique. In this study 3000 individuals in the IS COP received direct email solicitation to participate, with 643 responses yielding a 21.4% response rate, markedly better than the 5-14% reported by other studies targeting PMI members (Mathur et al. 2013; Wallace et al. 2004b). Filter questions used to qualify experienced project managers with a deep understanding of their project further reduced the sample to a final analysis size of 268 responses.

A related concern is non-response bias. Most survey studies receive responses from a subset of targeted participants. If those who respond are unusual and provide information that is not representative of the target population, then the results can be biased. The study design employed multi-item scales to measure substantive latent constructs and thereby mitigates the impact of the isolated non-response bias. Furthermore, it has been shown that perceptions of anonymity and confidentially are strongly correlated with self-report accuracy (Bates and Cox 2008). Solicitation emails and the survey introduction offered assurances of anonymity for participants of this study. The survey did not ask information that would identify their person, their organization, or the project they were addressing.

A *time trends* method is commonly applied to measure the non-response bias (Lee and Xia 2010; Wakefield et al. 2008). In this study a reminder was sent to non-respondents after 10 days. The responses before the reminder and those after the reminder were compared under the presumption that the difference between groups is an indicator that non-respondents provide meaningfully different answers. In total there were 120 qualified responses before reminder and 148 after reminder. A t-test comparison of means[9] for measures in the research model provided support at the α=0.05 level that no significant differences exist with the exception of AS5 (p=0.038), RD4 (p=0.001) and RD5 (p=0.041). Among control variables, two differences emerge: responses after the reminder involved more geographically dispersed teams (C2, p=0 .032) and more geographically dispersed users (C3, p=0 .024). Neither of these control variables was significantly associated with outcomes and were dropped further analysis. A similar test of demographic items revealed no significant differences. In total the mean response for 5 of 68 questions (7%) were statistically different after the reminder. Assuming response patterns are randomly distributed across the time period, this quantity of questions with a statistical difference before and after reminders are within the range expected by random chance for 268 respondents. It is reasonable to conclude that the non-response bias is unimportant in this study.

5.2.3 Common Method Variance

This study uses a multi-method multi-phase approach in that the original theoretical ideas and models were developed based on information obtained in a case study of four projects, with supporting evidence obtained in cross-sectional survey spanning hundreds of projects. Studies based on survey instruments alone may have biased results stemming from the use of a single method to gather information. The behavioral science literature is concerned with "Common Method Variance" (CMV) and has developed several methods to assess the potential bias. Among the techniques

[9] This test performed using STATA 12.

employed, “Comprehensive CFA Marker” technique has demonstrated value for IS research (Malhotra et al. 2006) and is more accurate than correlation based marker variable method or the Harmon’s one factor test (MacKenzie and Podsakoff 2012).

The Comprehensive CFA Marker technique described by Williams et al. (2010) has been followed for the data collected in this study. This study included three questions asking respondents about the quality of the legacy system that was being replaced by the project they managed. The answers to these questions should not be related to the responses for any of the salient constructs in this study. If CMV is a problem, then response patterns for the marker variable should coincide with patterns in the rest of the survey. Alternately, independent and uncorrelated marker variables is a strong indicator that respondents were assessing each question on its own merits and providing accurate feedback for each theoretically distinct construct. Interestingly, marker question 1 (MK1 “the old system was well understood by users”) did not correlate with the others (MK2 “the old system provided poor quality information”; MK3 “users thought the old system was unreliable”) to be included in the CMV assessment. This supports the conclusion that nuances of seemingly related questions (in this case adjacent questions about the old system) were fully evaluated with meaningful distinctions made; evidence that CMV is not a problem for this dataset. The remaining two items converged with good validity and served for the following analysis.

The CFA Marker technique analyzes a set of four common factor models[10] with different combinations of correlations assumed. The first model (called the CFA Model – illustrated in Appendix C) assumes the marker items are not related to the substantive items, and takes an otherwise classic CFA approach of allowing all latent factors to correlate freely. This model establishes item coefficients and variances for the marker construct and has a Chi-square statistic of 700.052 (df=330). The second model (Baseline) assumes there is no correlation between marker and substantive latent factors, and fixes the marker item and error variances to the values obtained in the first CFA Model. This baseline model represents the measures and latent factor correlations as they would be if no CMV existed. This second model has a Chi-square statistic of 713.655 (df=338). The third model (Method-C) allows the marker latent factor to cross-load with all substantive measurement items. In this model the cross-loading correlation is constrained to have equal values on the presumption that method variance would have a consistent bias across all measures. This model has a Chi-square statistic of 711.975 (df=337). This model does not differ from the Baseline model to a statistically significant degree (p=0.195), supporting a conclusion of no common method bias for this dataset. The fourth model (Method-U) allows the marker latent factor to cross-load in an unconstrained manner with all substantive items. This model considers that the method effects may not be uniform, but may be meaningful for only certain questions (e.g., users concerned about their image may bias their reporting of project performance, while

[10] Comprehensive CFA Marker models were assessed with covariance based Structural Equation Modeling using STATA 12.

reporting objectively about the severity of fluctuations). This model has a Chi-square statistic of 688.211 (df=312). Again, this model does not differ from the Baseline model to a statistically significant degree (p=0.494), further supporting a conclusion of no common method bias for this dataset. This assessment provides strong evidence that the remediation provided in the study design (e.g., using referent-shift questions, assuring anonymity, reducing ambiguity in wording of questions) were effective in mitigating CMV.

While the Comprehensive CFA Marker technique has demonstrated strengths over other detection methods, there remains some question of its effectiveness (Richardson et al. 2009). With the possibility that undetected method bias may still exist, it is relevant to understand how this could bias hypothesis testing in this study. Several scholars observe that the inflationary effect of method variance is offset by the attenuating effect of measurement error (Lance et al. 2010; Malhotra et al. 2006), such that statistical tests remain accurate overall. Of particular interest to this study are the effects on testing interaction effects central to this study. It has been demonstrated that CMV deflates interaction calculations (Siemsen et al. 2010). Statistics of these non-linear effects cannot be inflated to reveal false positives (Type I error), but can be attenuated by CMV and thereby hide relationships that actually exist (Type II error). In the end, the primary risk to hypothesis testing for the constructs that address new theoretical insights is that this dataset will fail to find relationships that in reality do exist (false negatives).

Where statistically significant results are found, particularly those involving the insightful interaction effects, the bias of CMV is not a concern.

Chapter 6

Research Results

This chapter describes results from hypothesis testing using the SmartPLS (Ringle et al. 2005) implementation of PLS-SEM[11]. Conclusions for hypotheses are assessed at α=0.05 level of significance common for the behavioral sciences. A Bootstrap re-sampling technique that is robust where data is not normally distributed is used to calculate the standard error and determine probability levels for hypothesis testing.

Two research models were examined independently. The first emphasizes the contingent effects of project management style in an environment of fluctuations that disrupt the normal course of events. This model provides insight for the performance implications of initial PM style. The second model directly conceptualizes the phenomenon of Reactive Decoupling to capture its interaction effect on project management style. This model provides insight regarding shifts in PM style associated with flux.

6.1 Sample Characteristics

A total of 268 survey responses were qualified for inclusion in this analysis as described in the previous chapter. The demographic profile of respondents is provided in Table 6-1. Of particular note is the experience level of participants with 84% possessing 5 or more years of experience as a project manager. Additional details on industry

[11] SmartPLS settings for estimation are as follows: *Mean replacement* of missing data, *Centroid* weighting, *Original* data metric, *Individual* sign changes, and *200* bootstrap repetitions.

breakdown are provided in Appendix B. In total this dataset effectively sampled experienced project managers across diverse settings.

Table 6-1 Demographics

Characteristic	**Value**	**Count**	**%**	**Histogram**
Age	Under 18	0	0%	
	18-24	1	<1%	*
	25-34	43	18%	*******
	35-44	107	44%	******************
	45-54	76	31%	************
	55+	0	0%	
Experience in role of project manager	<1 year	3	1%	*
	1-2 years	9	3%	*
	3-4 years	28	10%	****
	5-10 years	81	30%	************
	10+ years	144	54%	*********************
Industry	Agriculture, farming or ranching	1	<1%	*
	Computer or Telecom hardware/software products or services	92	34%	**************
	Consumer products or services	17	6%	***
	Construction	3	1%	*
	Defense	14	5%	**
	Entertainment, sports and recreation	2	1%	*
	Financial products or services	28	10%	*****
	Government or public sector	20	8%	***
	Healthcare products or services	29	11%	*****
	Manufacturing or industrial	20	8%	***
	Not-for-profit organization	8	3%	*
	Transportation	8	3%	*
	Other	25	9%	****

6.2 Model 1 – Baseline PM Style and Performance

The first research model conceptualizes fluctuations as an independent variable and stimulus that triggers negative response in performance. This response may be moderated by several factors, including the project management style and the locus of fluctuations. Overall findings are depicted in Figure 6-1 and summarized in Table 6-2. In addition to reporting path coefficients (β), which are interpreted as the strength of the correlation, the effect sizes (f^2) are also provided. Effect sizes which are interpreted as the level of importance of the relationship, should be viewed with caution as the PLS method that has demonstrated power advantages for detecting statistically significant relationships (Reinartz et al. 2009), provides somewhat less accurate path estimates (Goodhue et al. 2012).

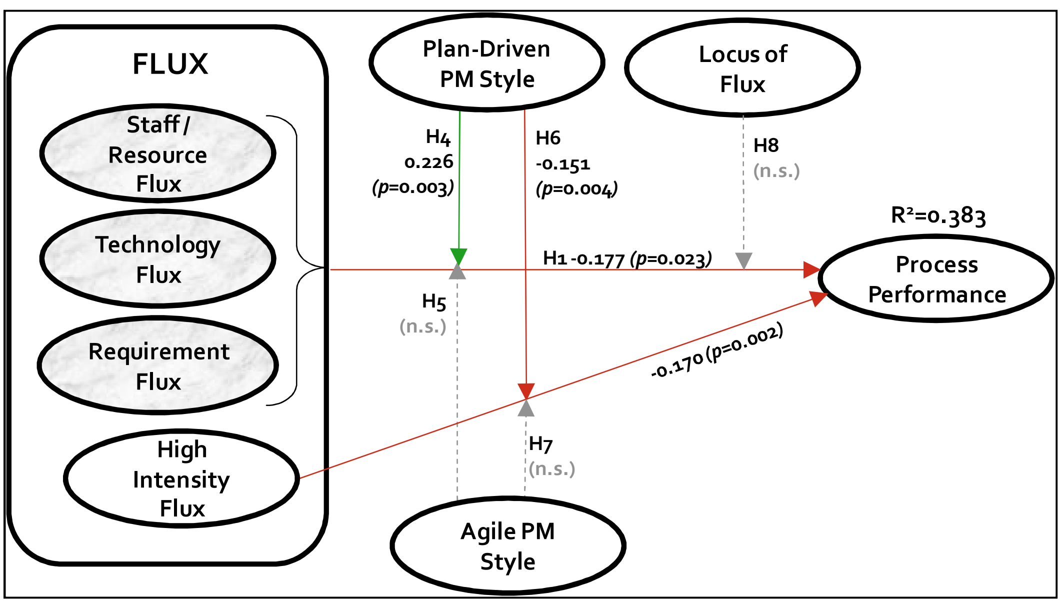


Figure 6-1 Contingent PM style results

6.2.1 Direct Effect of Fluctuations

This research model aggregates staff fluctuations, technology fluctuations and requirements fluctuations into a second order construct that serves as a proxy for general flux. This overall flux phenomenon has a statistically significant[12] negative effect on performance (β = -0.177, f^2=0.05, p-value = 0.023), providing support for Hypothesis 1. Post hoc analysis that examines each first order flux construct in an un-saturated model reveals that Requirements Flux (RF) is statistically significant (β = -0.311, f^2=0.12, p-value < 0.001), Staff Flux (SF) is statistically significant (β = -0.156, f^2=0.03, p-value = 0.011), but Technology Flux (TF) is not significant (β = -0.036, f^2=0.00, p-value = 0.213). These project managers may view the technology problems to be normal activities for their projects to manage, and relatively non-disruptive when also considering staff and requirements fluctuations.

The post hoc analysis also examined the relationship of these constituent fluctuations on three elements of perceived satisfaction. Staff Flux (SF) is negatively associated with perceived stakeholder satisfaction (SS) (β = -0.121, f^2=0.02, p-value = 0.037), member satisfaction (MS) (β = -0.317, f^2=0.14, p-value < 0.000), but not perceived user satisfaction (USI) (β = 0.053, f^2=0.00, p-value = 0.366). Requirements Flux (RF) is negatively associated with perceived stakeholder satisfaction (SS) (β = -0.207, f^2=0.05, p-value = 0.004), member satisfaction (MS) (β = -0.236, f^2=0.08, p-value

[12] Hypothesized path coefficients are evaluated for statistical significance using a one-tailed test. P-values for control variables and item loadings were evaluated using a two-tailed test.

< 0.001), and user satisfaction (USI) (β = -0.171, f^2=0.03, p-value = 0.021). Technology Flux (TF) is negatively associated with perceived stakeholder satisfaction (SS) (β = -0.163, f^2=0.03, p-value = 0.016), member satisfaction (MS) (β = -0.124, f^2=0.02, p-value = 0.048), and user satisfaction (USI) (β = -0.272, f^2=0.09, p-value = 0.003). These findings suggest that flux has a mostly consistent impact on project manager perceptions of satisfaction. To the extent that perceptions provide a motivating stimulus on project managers during in-project decision-making, it is plausible that these perceptions foreshadow changes and management manipulations the project manager attempts during the project.

Table 6-2 Summary of model 1 results

Model 1 hypotheses	path coefficient	p-value	Std. Error	f^2	Conclusion
H1: FLUX (-)→ PERF	-0.177	0.023	0.062	0.05	Supported
H2: PDS (+)→ PERF	0.106	0.005	0.042	0.02	Supported
H3: AS (+)→ PERF	0.277	0.000	0.053	0.12	Supported
H4: FLUX * PDS (+)→ PERF	0.226	0.003	0.082	0.08	Supported
H5: FLUX*AS (+)→ PERF	-0.016	**0.579**	0.076	0.00	***Not Supported***
H6: HIF * PDS (-)→ PERF	-0.151	0.004	0.045	0.04	Supported
H7: HIF * AS (+)→ PERF	0.057	**0.119**	0.049	0.01	***Not Supported***
H8: FLUX*LoF (-)→ PERF	-0.076	**0.078**	0.054	0.01	***Not Supported***

6.2.2 Interaction Effect of Project Management Style

This research model characterizes the role of project management style as a moderator of fluctuations. The idea starts with an assumption that plan-driven and agile project management processes have some built in resiliency and capability to manage fluctuations. The moderating influence of these styles is examined in the interaction effects of PM style on the direct effect of fluctuations. The data provides support for Hypothesis 4 and the positive interaction effect of plan-driven style on the relationship between fluctuations and performance (FLUX*PDS) (β = 0.226, f^2=0.08, p-value = 0.003). A nuance of this moderating effect is the increased challenge presented by increased flux intensity. The negative interaction effect between high intensity Flux (HIF) and plan-driven style (PDS) supports Hypothesis 6 (HIF*PDS) (β = -0.151, f^2=0.04, p-value = 0.004). The interaction effects of Plan-Driven style with Flux and HIF can be seen in the top two graphs in Figure 6-2. The top left graph shows the beneficial influence of strong plan-driven style as Flux increases. However, when flux is stratified to isolate the effects of High Intensity Flux, there is a decidedly negative effect associated with strong plan-driven bias. This suggest that the value of strong plan-driven bias may be a non-linear function that increases in value as flux increases in intensity until it reaches an inflection point and imposes negative marginal returns.

The situation is less conclusive for agile style. The data in this study failed to find support for Hypothesis 5 and its predicted positive interaction effect of agile style on the relationship between fluctuations and performance (FLUX*AS) (β = -0.016, f^2=0.00, p-value = 0.579). Similarly, the data does not support Hypothesis 7 and its predicted interaction effect of high intensity Flux and agile style (HIF*AS) (β = 0.057, f^2=0.01, p-value = 0.119). The interaction effects of Agile style with Flux and HIF as shown in the bottom two graphs of Figure 6-2 demonstrate the lack of interaction effect, and emphasize the positive direct effect of Agile style for all flux intensities.

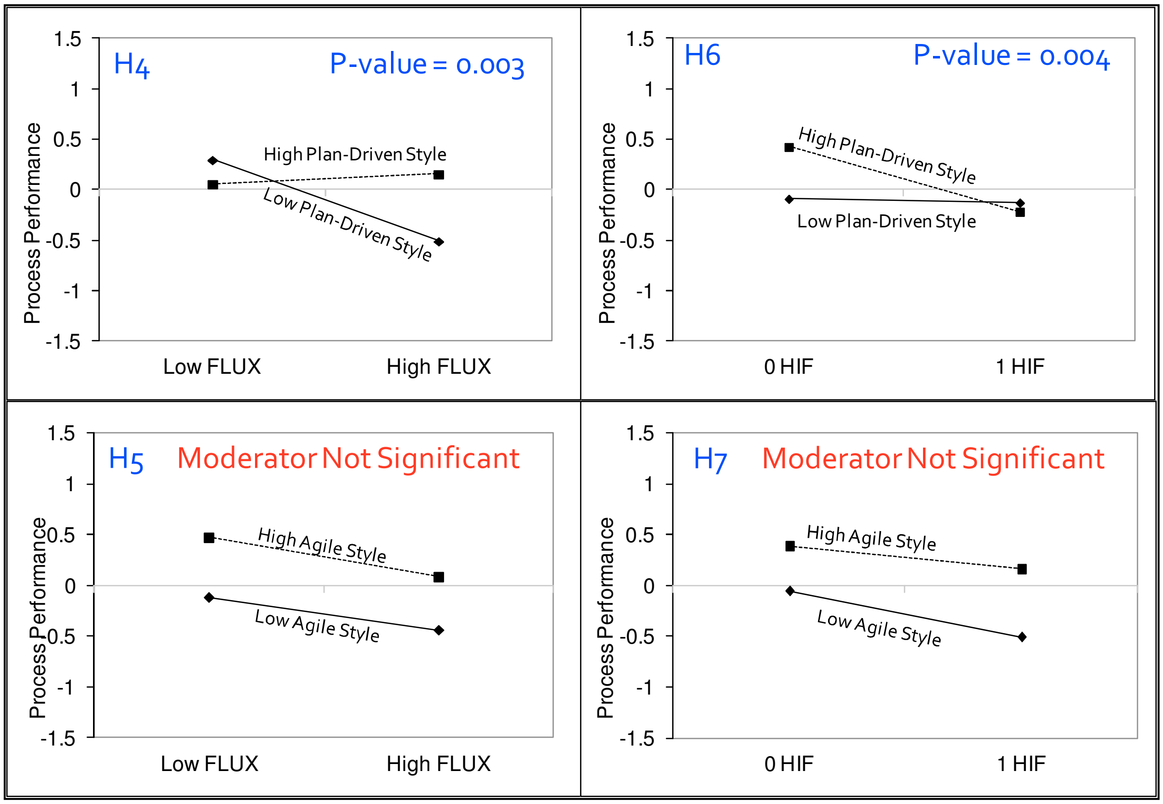


Figure 6-2 Moderation effects of PM style

6.2.3 Interaction Effect of Locus of Fluctuation

Another contingency effect is predicted for the interaction associated with External Locus of Flux (LoF). Hypothesis 8 predicts that fluctuations originating outside the project team will have a greater negative relationship to performance than those originating within the team. The data fails to find support for the interaction effect of External LoF on fluctuations (FLUX*LoF) (β = -0.076, f^2=0.01, p-value = 0.078). While the survey data does not support an influence from Locus of Flux, the findings from the case study suggest this is an area for additional study.

6.3 Model 2 – Directly Conceptualized Reactive Decoupling

This second research model conceptualizes project management style as an independent variable with an observable relationship with performance. Fluctuations have an influence that can unseat initial style and habits then induce a shift to different behaviors. This shift is predicted along the loose coupling flexure in the direction of greater decoupling. Overall findings are depicted in Figure 6-2 and summarized in Table 6-3. As with model 1, conclusions for hypotheses are assessed at $\alpha = 0.05$ level of significance common for the behavioral sciences.

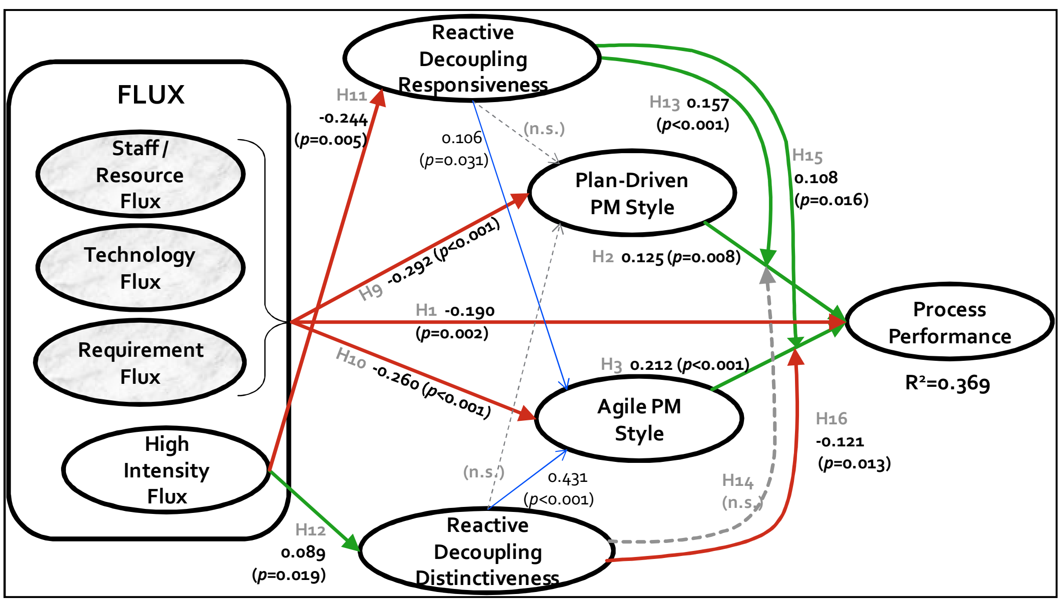


Figure 6-3 Reactive Decoupling results

6.3.1 Direct Effect of PM Style

The second order flux phenomenon has a statistically significant negative effect on performance (β = -0.190, f^2=0.06, p-value = 0.002), providing support for Hypothesis

1. The data also supports a direct positive relationship between PM style and performance. There is support for Hypothesis 2 and the positive relationship from plan-driven style (PDS) (β = 0.125, f^2=0.02, p-value = 0.008), as well as Hypothesis 3 and the positive performance relationship from agile style (AS) (β = 0.212, f^2=0.07, p-value < 0.001). The path coefficients and effects sizes suggest that agile style and its loose coupling behaviors have a potentially larger positive effect on project performance. Complex projects appear to be well suited for the advantages offered by loosely coupled systems.

Table 6-3 Summary of model 2 results

Model 2 hypotheses	path coefficient	p-value	Std. Error	f^2	Conclusion
H1: FLUX (-)→ PERF	-0.190	0.002	0.066	0.06	Supported
H2: PDS (+)→ PERF	0.125	0.008	0.052	0.02	Supported
H3: AS (+)→ PERF	0.212	<0.001	0.053	0.07	Supported
H9: FLUX (-)→ PDS	-0.292	<0.001	0.054	0.13	Supported
H10: FLUX (-)→ AS	-0.260	<0.001	0.070	0.11	Supported
H11: HIF (-)→ RDresp	-0.244	0.005	0.096	0.09	Supported
H12: HIF (+)→ RDdist	0.089	0.019	0.043	0.01	Supported
H13: PDS * RDresp (+)→ PP	0.157	<0.001	0.05	0.04	Supported
H14: PDS * RDdist (+)→PP	0.029	**0.325**	0.064	0.00	***Not Supported***
H15: AS * RDresp (+)→ PP	0.108	0.016	0.051	0.02	Supported
H16: AS * RDdisp (-)→ PP	-0.121	0.013	0.055	0.02	Supported

6.3.2 Direct effect of FLUX on PM Style

The theory of Reactive Decoupling suggests that projects do not maintain their baseline PM style, but rather shift along the loose coupling flexure in response to fluctuations. If change is to occur, we expect there to be an "unfreezing" of baseline PM style in response to fluctuations that is revealed by a negative association between fluctuations and PM style. The data provides support for Hypothesis 9 that fluctuations are negatively associated with plan-driven style (β = -0.292, f^2=0.13, p-value < 0.001). The data also supports Hypothesis 10 that fluctuations are negatively associated with agile style (β = -0.260, f^2=0.11, p-value < 0.001). This finding supports the proposition that project management style does not remain fixed throughout complex IT projects.

6.3.3 Reactive Decoupling

Once fluctuations have unfrozen the baseline PM style of a project, other influences associated with the mechanics of flux come to bear. The reactive influences of responsiveness and distinctiveness capture the dialectic effect of these pressures on the positioning of an individual project within the PM Style framework. *Responsiveness*, to the extent it exists, will tend to support maintaining the tight coupling properties of the baseline PM bias and help the team resist a shift toward decoupled systems behaviors. When projects react to fluctuations with increased responsiveness, this reflects pressure up the loose coupling flexure. *Distinctiveness*, to the extent it exists, will provide the influence to accelerate a shift toward decoupled systems behaviors. Reactions that increase distinctiveness take the form of communication buffers that reduce boundary spanning beyond the team. This can remove structure needed to maintain an Agile project on the edge of chaos and reflect pressure down the loose coupling flexure in the direction of decoupling. This influence may counterbalance tight coupling influences, or if strong enough, may shift the project down into the region of decoupled systems behavior.

The hypotheses predict these pressures will be most evident in an environment of severe fluctuations. The data reveal that High Intensity Flux (HIF) is *negatively* associated with responsiveness. Support for Hypothesis 11 (β = -0.244, f^2=0.09, p-value = 0.005) has an important implication within the theory of Reactive Decoupling. Irrespective of the potential value of balanced responsiveness and distinctiveness, high intensity fluctuations appear to disrupt responsiveness behaviors, providing support for the shift down the loose coupling flexure. The data also supports Hypothesis 12 and its prediction that High Intensity Fluctuations (HIF) are positively associated with *Reactive Decoupling - distinctiveness* (β = 0.089, f^2=0.01, p-value = 0.019). High intensity fluctuations create the conditions for increased *distinctiveness*.

6.3.4 Interaction Effect of Reactive Decoupling

The performance consequences of Reactive Decoupling are exposed in the interaction effects on PM style. The data supports the Hypothesis 13 prediction of a positive moderating relationship by *Reactive Decoupling responsiveness* on plan-driven style (PDS*RDresp) (β = 0.157, f^2=0.04, p-value < 0.001). This positive relationship suggests there is performance value maintaining responsiveness behaviors such as boundary spanning. The top left graph in Figure 6-4 demonstrates the positive interaction effect that Reactive Decoupling responsiveness has on plan-driven style. However, as revealed in H11, high intensity flux undermines responsiveness and deprives projects of this influence. Data does not support the Hypothesis 14 prediction of a positive moderating relationship by *Reactive Decoupling distinctiveness* on plan-driven style (PDS*RDdist) (β = 0.029, f^2=0.00, p-value = 0.325). The lack of interaction effects is depicted in the top right graph in Figure 6-4. While the conditions for distinctiveness exist, as revealed in H12, this study does not provide insight into the performance consequences for plan-driven projects. This is perhaps to be expected, as the study provides no means to detect how far along the loose coupling flexure a plan-driven project moves. A shift into the loosely coupled systems region would plausibly have positive performance consequences, while a shift all the way into the decoupled region would have negative consequences. The inability of the current study design to detect how far a projects shift, leaves both the positive and negative outcomes mixed in a non-descript aggregate.

Turning to agile projects, the data supports the Hypothesis 15 prediction of a positive moderating relationship by *Reactive Decoupling responsiveness* on agile style (AS*RDresp) (β = 0.108, f^2=0.02, p-value = 0.016). This positive relationship that suggests a performance advantage from maintaining responsiveness practices such as boundary spanning is depicted in the bottom left graph of Figure 6-5. As noted above, Hypothesis 11 reveals that high intensity flux is associated with loss of responsiveness, and therefore may deprive a project of this value. Data also supports the Hypothesis 16 prediction of a negative moderating relationship by *Reactive Decoupling distinctiveness* on agile style (AS*RDdist) (β = -0.121, f^2=0.02, p-value = 0.013). The effects of this interaction are depicted in the bottom right graph of Figure 6-4 where RD distinctiveness erodes the potential value of Agile style.

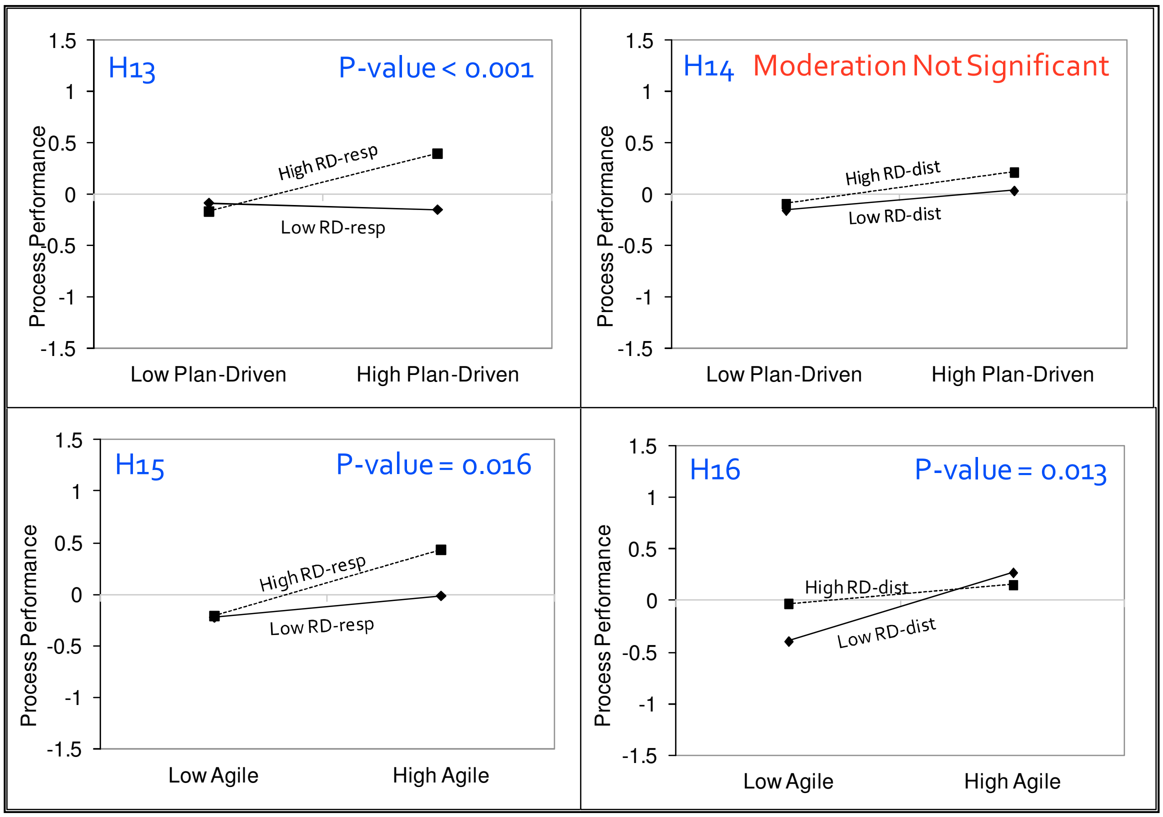


Figure 6-4 Moderation effects of Reactive Decoupling

The combination of forces exposed by this research model is telling. In the face of fluctuations, PM style habits and routines become unfrozen. These pressures encourage the project team to alter its behaviors within the project management style framework. High intensity flux is associated with reduced responsiveness while simultaneously increasing distinctiveness. In the face of high intensify fluctuations, these pressures conspire to encourage PM style shift along the loose coupling flexure in the direction of decoupled systems behavior. While fluctuations may induce positive effects for plan-driven projects, the implication for agile projects is decidedly negative. The dialectic of responsiveness and distinctiveness plays out as a struggle to maintain a loose coupling balance. Where the balance is upset, agile projects teetering on the edge of chaos may pursue practices with sub-optimal outcomes.

6.3.5 Post-Hoc Analysis of Reactive Decoupling

Post-hoc analysis have been used in the IS literature to provide additional insights related to constructs involved in interactions (Angst and Agarwal 2009). In this study Reactive Decoupling constructs drive important moderating effects on project outcomes. A post-hoc assessment of the direct effect of both Reactive Decoupling constructs did not find a significant association with Plan-Driven style[13]. Where fluctuations are negatively associated with both responsiveness behaviors and plan-driven style, it is not surprising that a weak Reactive Decoupling responsiveness is not supportive of plan-driven style. Furthermore, the plan-driven bias does not emphasize distinctiveness. A strong distinctiveness response is therefore not expected to be supportive of plan-driven style.

A Both Reactive Decoupling constructs do have a significant positive relationship with Agile style. These relationships provide support for the interpretation of loose coupling as a balance involving the tension between dialectic properties. Furthermore, the

[13] Neither RDresp ($p=0.187$) nor RDdist ($p=0.155$) are significantly associated with PDS. Both RDresp ($p=0.031$) and RDdist ($p<0.001$) are significantly associated with AS in a positive direction.

relative weakening of responsiveness and strengthening of distinctiveness in the face of fluctuations puts the balance in jeopardy. Irrespective of the effectiveness and suitability of agile methods, fluctuations generate pressures that may move Agile projects from the edge of chaos toward the domain of ad-hoc behaviors. At just the times when a project would most benefit from loosely-coupled systems behavior, the mechanisms of Reactive Decoupling drive projects toward greater decoupling. For some projects this may involve a shift into or within the loosely coupled systems region. However, where the shift is dramatic enough, project teams may move to the decoupled systems region and cross the tipping point into chaotic ad hoc behaviors that depend upon self-control of the team to produce outcomes aligned with organization interests.

These finding do not invalidate Agile methods. Agile practices were introduced to deal with a particular form of fluctuations – requirements fluctuations. However, IT projects operate in environments containing many types of disruptions, and the prescription of Agile behaviors at the onset of a project does not assure the project maintains those behaviors.

Chapter 7

Discussion and Conclusion

Responsiveness and Distinctiveness are valued traits of loosely coupled systems. High *responsiveness* will establish IT project teams as tightly coupled agents of the organization, aligned with plans, schedules, and approved specifications with an emphasis on objectively verifiable bureaucratic control outputs. A high propensity for *distinctiveness* will empower a team to behave as a decoupled system subject to the whims and adhockery of self-control. Where responsiveness and distinctiveness exist in balance, the associated system behaves with the ambidexterity of loosely coupled systems, and is perhaps optimally positioned to manage change and disruptive events.

The PM Style Framework positions project management methods in a two dimensional grid with the vertical axis associated with the degree of bureaucratic control, and the horizontal axis associated with the degree of concertive control. The control landscape is further segregated into three regions of tightly coupled, loosely coupled and decoupled systems around a curved flexure with extreme plan-driven style at one end, uncontrolled ad hoc style at the other. Across the center is the ambidextrous region of agile style project management. This PM Style Framework overlays Control Theory with the theory of Loosely Coupled Systems and provides insights that agile style embraced as baseline behavior at the start of a project may be advantageously suited to manage fluctuations in the form of disruptions and deviations.

The proposed theory of Reactive Decoupling posits that PM style is not a fixed decision. Rather, teams respond to fluctuations by adapting their project management behaviors. Analysis from a comparative case study provides the insight that project teams improvise changes to their PM style along the flexure of loose coupling in the direction of decoupling (Figure 3-2, page 89). A survey study has been conducted to complement the theory building case study in order to empirically test both the Framework's ability to predict performance benefits of baseline PM style and simultaneously the theory of Reactive Decoupling with its prediction of improvisational PM behaviors.

7.1 Model 1 – Baseline PM Style and Performance

The first model confirms that project managers report fluctuations are detrimental to project performance. This model also supports the view that formal project management methodologies and practices have a beneficial relationship with project performance. An important insight of overlaying the theory of loosely coupled systems on the PM Style Framework is the prediction that an initial agile style can have an advantage over plan-driven style. The concept is supported by strength of influence revealed in effect size differences. Plan-driven style is exposed to have a smaller effect size than agile style. The modest overall effects of plan-driven style are explained by the different interaction relationships it has with general Flux (positive) and High Intensity Flux (negative). Interestingly this study fails to find statistically significant interaction effects for agile style or for Locus of Flux. The divergent interaction effects of plan-driven style and the failure to find interaction effects of agile style in the face of fluctuations suggests that initial PM Style may not be the key determinant of project performance.

7.2 Model 2 – Reactive Decoupling and PM Style Shift

While initial PM Style may not determine project success, the theory of Reactive Decoupling posits that project teams may abandon initial PM style as they respond to fluctuations. The second research model confirms that fluctuations are negatively associated with initial PM style, an effect that can be understood as an unfreezing of project management routines and habits. The study further provides empirical support that High Intensity Fluctuations are negatively associated with responsiveness behaviors, a circumstance that weakens plan-driven style. Simultaneously, high intensity fluctuations are positively associated with distinctiveness behaviors, a condition that strengthens decoupling. This combination has affiliated interaction effects that are positive for projects that begin with a plan-driven style and negative for projects starting with agile style.

The overall performance key for IT projects faced with fluctuations may be the degree to which teams improvise and alter their PM style. As the baseline style is unlikely to remain, particularly in the face of high intensity flux, behaviors embraced after the shift can be far more important to IT project performance.

7.3 Implications for Theory

The primary contribution of this study is the introduction of a new theory describing mechanisms at work as IT projects encounter fluctuations in the form of deviations and disruptions. The project management area has lacked a guiding theory despite the elevated role it has within organizations. The theory of Reactive Decoupling is strategically positioned at the intersection of organizational control and change that is inevitable for complex projects. The empirical support provided by this multi-method study generates confidence in the applicability of this theory for IT projects dealing with malleable software.

A secondary contribution is the PM style Framework that overlays Control Theory with the theory of Loosely Coupled Systems. The framework provides a means to understand the relevant relationships between project management styles and has value for future project management theories addressing phenomenon beyond deviations.

The empirical support provided here substantiates the characterization of projects as temporary organizations (Lundin and Soderholm 1995), such that many ideas from the domain Organizational Theory apply to projects. In particular, Reactive Decoupling of project teams parallel the ideas of Dynamic Capabilities involving firm level reconfiguration in response to dynamic markets (Eisenhardt and Martin 2000). Improvisational Capabilities have been advanced as an extension of Dynamic Capabilities that exist for firms operating in highly turbulent environments (Pavlou and

ElSawy 2010). Project teams similarly exhibit more dramatic Reactive Decoupling responses when faced with high intensity Flux.

A third contribution is the introduction and explication of cognitive inertia. This phenomenon serves as an impediment for teams and individuals that requires boundary spanning. This type of boundary is particularly insidious as it lurks in unspoken beliefs, invisible memory models, unmotivated participants, invalid assumptions, and insufficiently realistic use cases. Preliminary insights from this study suggest that rigorous communication practices may be more effective than informal information exchange. Additional study of cognitive inertia is needed to fully expose this concept, develop better understanding of its influences and establish the efficacy of various remediation tactics.

7.4 Implications for Practice

Recognizing that teams improvise and alter their project management practices is an important insight for IT project teams and organizations alike. Project managers faced with fluctuations are often well served by adopting practices associated with loosely coupled systems. Efforts to ensure communications continue outside the team and resisting the tendency for isolation can often generate important performance returns. Leveraging tiger teams and other self-forming sub-groups to focus on emergent disruptions allow tightly coupled plan-driven teams to temporarily harness the innovative and variety advantages of loosely coupled systems. The ability of the project manager, or indeed members of the team, to recognize the PM style shifts taking place in response to fluctuations provide a degree of control over the situation that does not exist when events manipulate the situation unchecked.

7.5 Limitations of This Study

Operationalization of both Reactive Decoupling dimensions (responsiveness and distinctiveness) depends upon two measures in this survey study. While these factors demonstrated both convergent and discriminant validity, two is the minimum for measuring psychometric constructs. The small number of measures limits the statistical power of this study and may contribute to the lack of confirmatory support for some hypothesized relationships suggested in the case study.

Another measurement related concern involves Locus of Flux where half the measures failed to converge on their intended construct, which itself failed to have any significant relationships. In this case a unique scale was used (1: Not at all – 6: Very great extent) that may have challenged respondents and increased the cognitive complexity of these questions. A scale consistent with the remainder of the survey may be an appropriate remediation in future studies.

The Work Systems Framework identifies nine domains (Alter 2006) for projects from which fluctuations may originate. This survey study operationalized only three of the nine. It is plausible that fluctuations associated with other domains may exhibit different properties that are important to fully understand the associated pressures and responses. Introducing additional sources of fluctuations opens the possibility that influences involving a granular investigation of particular forms of flux can provide additional insights.

7.6 Suggestions for Future Research

The PM Style framework presented in this manuscript places PM methodologies within the framework based on the judgment of the author. They serve as examples to facilitate the discussion. The actual positioning of an individual project is not critical for the purposes of this study. A more rigorous cataloging of practices and their relationship with bureaucratic and concertive controls could provide value to practitioners or scholars seeking to understand the relative value of particular practices in certain situations.

This particular study pursued a dual purpose of validating the PM style framework and empirically supporting the theory of Reactive Decoupling. The multiple objectives are well served by a cross sectional survey investigation technique. However, the design also represents a compromise that does not assess the velocity and distance of PM style shifts motivated by fluctuations. A longitudinal study may be well suited to more effectively measure project management improvisations.

A related direction for future research involves the decomposition of Flux and capturing the unique dynamics between different origins of unexpected challenges and the associated relationships both initial PM style and Reactive Decoupling. It is highly plausible that agile style, which gained popularity based on its ability to manage evolving requirements, may serve that type of change particularly well. However agile style may be more vulnerable to fluctuations emerging from organization and management strategy changes that do not become visible to a highly autonomous team as readily.

The case study revealed a new form of boundary in cognitive inertia. This phenomenon is a candidate for further development and clarification through both qualitative and quantitative study. The largely hidden nature of this construct may be very important for exploring other aspects of IT projects such as individual and team learning, or requirements elicitation.

7.7 Conclusion

Plan-driven project management biases are observed in the strict alignment of IT project teams to structured plans with prescribed documentation practices for specifications, plans, schedules, tests, training and all manner of formal project deliverables. These objectively observable work products allow organizations to monitor and evaluate projects in accordance with prescribed bureaucratic controls. Agile project management biases are observed in values reinforcing communication rituals and autonomous flexibility. Highly effective agile teams demonstrate an ambidexterity that maintains alignment with the organization through value centric concertive and cultural controls while benefiting from the requisite variety afforded by empowering teams and sub teams to self-assign tasks and independent decision making related to designs and solutions.

Agile style is aligned with an ideal form of loosely coupled systems organization where the balance between distinctiveness and responsiveness allows the project team to respond quickly to change while simultaneously serving organizational goals and objectives. Project teams that operate with behaviors associated with a balance of responsiveness and distinctiveness are particularly well suited to manage fluctuations that would otherwise disrupt an IT project.

This multi-method multi-phase study empirically demonstrates the tendencies of plan-driven and agile project management biases that place them within a PM style framework that combines the two dimensions of bureaucratic controls and concertive controls. This framework is further refined with the overlay of a loose coupling flexure with tightly coupled systems associated with the region of highly bureaucratic plan-driven practices and decoupled systems associated with the region of ad hoc practices. The middle region of loosely coupled systems behavior involves a balance of bureaucratic and concertive controls. When projects operate in this middle region, they are well positioned to handle unexpected challenges.

The most significant finding in this study is the empirically supported shift in project management style triggered by fluctuations in the form of unexpected challenges. High intensity fluctuations in particular are associated with improvisational shifts in PM style in the direction of decoupling. These shifts allow plan-driven projects to behave, at least temporarily, as loosely coupled systems with performance advantages. If the shifts are dramatic enough they may drive project teams to behave as decoupled systems, with performance disadvantages. The insight from the theory of Reactive Decoupling is that the initial PM style may matter less to project success than the improvisations and changes in PM behaviors that take place during IT projects. Furthermore, these improvisations are not random, but occur along the loose coupling flexure in the direction of increased decoupling.

IT projects faced with fluctuations have a tendency to gravitate toward decoupling in a way that resonates with the concepts of entropy in the physical universe. Where structures are sufficiently resilient, teams may benefit from operating in the ambidextrous region of loosely coupled systems. Where structures and organizational controls are weak, teams may digress into an adhocracy where outcomes are dependent on the self-control of individuals and teams.

Appendix A

Comparative Case Study

This study pursues an investigation of project management bias and a presumption that documentation itself plays an active role in the project teams activities. The case analysis described here occurs in two stages. First, a theory based framework provides structure to guide case study design and data collection (Yin 2009). An Interpretive assessment of a priori propositions is followed in the second stage by a Critical Realist exploration of the data in order to build new theory. This dual approach has been identified as a model practice for rigorous case study research in the Information Systems literature (Pare 2004)

A.1 Conceptual Development

While many IT activities are fully performed by individuals, particularly sustaining lifecycle maintenance tasks such as bug fixes, this is not how most new system implementations are accomplished. Within organizations, IT projects are group undertakings (He et al. 2007; Tiwana and McLean 2005). The literature on group cognition and macro-cognition provides a theoretical vantage point for the case study design investigating the response of IT project teams to unexpected challenges.

A.1.1 IT Projects and Collective Cognition

Organizations convene teams where social processes interact with cognitive and motivational processes to accomplish complex technical work that exceeds the capacity of a single individual (Curtis et al. 1988; Haynes and Smith 2008). Teams form and use group cognitive capabilities while working in an integrated and complementary manner to accomplish intellective tasks such as IT projects. “Successful software development is viewed as a consequence not of a single programmer’s cognition, but of an interaction of programmers and development artifacts in a system of distributed cognition. These systems have cognitive properties distinct from those of individual agents.” (Flor and Hutchins 1992) When groups perform collective tasks in the context of a complex socio-technical environment, collective team cognition emerges that is more than the cumulative individual cognitions of its members (Haynes and Smith 2008). The dynamics of group cognition involve mechanisms uniquely relevant to groups, including: group cohesion, group self-efficacy, shared memory (common information and shared beliefs), and consistent memory (aligned but not identical), distributed memory (diverse information), transactive memory (who knows what allows effective pooling of unshared information), groupthink, social decision schemes (e.g., majority rule), and social loafing (Cooke et al. 2007, McComb 2008). Some scholars suggest that all cognition is macro-cognition, integrative of emotion, motivation and collective interaction (Flach 2008).

The cognitive process of a group involves the sequencing of activities among individuals with the collaboration of dyads, triads and sub-teams. As tasks are initiated, transitioned, paused, and restarted, groups leverage shared common memory on goals, objectives and strategies to synchronize their efforts. Transactive memory allows the team to direct specialty tasks toward members possessing unique knowledge and abilities. Ideas, decisions and processes considered and performed in one setting are retained as memory schemas that allow the team to quickly repeat activities in other similar situations (Haynes and Smith 2008; Klein and Hoffman 2008). External artifacts allow individuals to store information and complex ideas as learning aids to grow shared memory models or to transport that information temporally and spatially to a different setting (Zhang and Patel 2006). External memory is unique among the elements of collective cognition as these artifacts allow collaboration and communication, the engine of group activity, in the presence of but a single team member. In so doing they allow the individual to leverage the cognitive influence of the team at any time and any place.

A.1.2 Requirements Change

Requirements changes are frequently identified as a source of concern and a leading motive for characterizing IT projects as complex undertakings. Users often do not fully understand what they need at the onset of a project. Over the course of discussion, or as a result of revelation induced by seeing the system (or a prototype of the system), users often evolve their requirements. Scope and requirements changes have been an ever-present concern for software developers. Such changes are identified as a disruption to be controlled in plan-driven methodologies through approaches such as early user involvement and stage gate reviews to formalize agreement on stable requirements. Alternately, requirements changes have been identified by agile advocates as inevitable characteristics of projects that should be embraced in order to better achieve user satisfaction. In situations where these changes cannot be deferred or avoided, it necessarily makes the project goal a moving target. Suddenly the shared mental models of the team need to be realigned on a new target. A team headed down a path toward a particular objective, when faced with a moving target, must make an accommodation in order to meet the changing objective (Levardy and Browning 2009). Judgmental decisions regarding solution architectures may need to be revisited and different choices made. The impact of changing requirements on IT teams employing either plan-driven or agile methods are believed to be negative (Appan and Browne 2010; Lee and Xia 2010), leading to the first proposition:

Proposition 1: Interruption events in the form of requirements fluctuation will be associated with decreased performance within IT projects.

A.1.3 Environmental Disruptions

The causes of project failures have been classified as internal or external (Morcos 2005). "Environmental disruptions" is the label applied to these external factors in this inquiry. This excludes requirements and scope change that are plausibly a natural part of all IT projects and addressed explicitly in proposition 1. This conceptualization also excludes process and quality faults (such as technology bugs) that emerge from the efforts of the core team, as these are a product of the group cognitive process, not an interruption of that process (though plausibly technology bugs may often be a second order effect of environmental disruptions – these are addressed explicitly in the next section). Environmental disruptions would include technology faults imposed on the project by outside teams or service providers, as well as decisions by management or events from the marketplace that alter the organizational strategy, technical strategy, or resource availability. Examples of this type interruption include: vendors who fail to meet schedules, weather or natural disasters that restrict participation of one or more members, other projects that consume key resources, and systems infrastructure service lapses. A more detailed explication and classification of disruptions external to the core project team is offered later in section A.2.2.2. For the purposes of this case study, environmental disruptions are interruptions that originate from outside the unit of analysis, which is the core project team.

An examination of teams working on a command and control simulation revealed acute stress to negatively affect mental models and transactive memory with an associated decline in team performance (Ellis 2006; Pearsall et al. 2009). Among New Product Development (NPD) teams, environmental turbulence has a negative impact on transactive memory systems and team performance (Akgun et al. 2006). Laboratory experiments have also examined disruption on dyads (Ruscher & Hammer 1994) as well as small and large groups (Ren et al. 2006), with demonstrable effects on transactive memory and group performance. It is plausible that a similar relationship exists for IT project teams. Therefore:

> *Proposition 2: Interruption events in the form of environmental fluctuations will be associated with decreased performance within IT projects.*

A.1.4 Plan-Driven Documentation as an Asset

The concepts of distributed cognition recognize that information relevant to performing a task is sometimes maintained outside the memory of individual members. Team knowledge involves both information possessed by all individuals as well as that stored external to individuals (Roloff and Van Swol 2007). Formal paper documents, books, email messages, e-documents, files, organized folders, and references to web pages remain important for decision makers (Jones and Ross, 2010). Distributed team memory includes an emergent "team knowledge" concept (Cooke et al. 2000), complemented by external information artifacts (Dror & Harnad 2008; Hazlehurst et al. 2008; Nemeth et al. 2004; Scaife & Rogers 1996; Susi & Ziemke 2001). Distributed memory and collective knowing constructs can contain explicit application and task specific knowledge as well as implicit process knowledge such as informal rules and relationships among team members which are unconsciously incorporated into the team structure and behaviors (Rouse et al. 1992). Structuration Theory resonates with these concepts and ideas which exist in a social setting beyond a single individual yet have a direct participative impact on task performance and interaction of a group (Jones and Karsten 2008). Within IT projects the need to capture explicit team knowledge in a way that it can be dispersed or accessed at later stages in a project has led to broad use of external information artifacts. Implicit recognition of this is found in the IS literature involving both hard (physical) and soft (electronic) artifact modeling techniques (Choppella et al. 2007; Shanks et al. 2008). The motive is to deliver knowledge across space (geographically to multiple group participants) and time (from design to coding or from decision to implementation).

Plan-Driven methodologies advocate formal documentation deliverables starting early in the project. From the standpoint of group cognitive processes, external artifacts in the form of specifications and design documents represent components of distributed memory models. IS researchers examining distributed cognition among people and technologies have emphasized team interaction processes involving the role of IT artifacts in facilitating the exchange of knowledge and information across the members of a group (Espinosa et al. 2007; He et al. 2007; Heylighen et al. 2007; Keith et al. 2009). The act of creating even simple artifacts not only triggers interactions and social cognition during its creation but also influences the process through which past knowledge is accessed (Cacciatori 2008) and decisions made (Permanova et al. 2008). These group cognition mechanisms are particularly important when dealing with conceptually complex algorithms or learning within unfamiliar domains where individual memory capacity is stressed. Furthermore, documentation plays a direct role of carrying forward effortful and carefully considered ideas temporally through a project. Various methods and medium have been investigated to facilitate the storage and dissemination of both explicit and implicit knowledge across phases of IT projects (Leonardi & Bailey 2008). During implementation activities, the group cognitive process draws upon heuristics and memory cues to minimize the cognitive load (Hayne and Smith 2008). Such tactics have been linked with IT project success (Busby 2001; Flor & Hutchins 1992; Keskin 2009). In addition, allowing group members access to information records, rather than relying on memory, helps attenuate certain group decision making traps such as overlooking unshared information (Kerr and Tindale 2004). Therefore:

> *Proposition 3: Creating external documentation artifacts during IT analysis and design activities is associated with increased performance.*

In addition to the posited direct effect of external documentation, artifacts also play a role mitigating the adverse effects of interruption events. External artifacts can contain substantially more detail than mental images, and some representations facilitate decision making (Vessey 1991) through superior ability to cue inferences for problem solving (Larkin and Simon 1987) and learning (Cheng 2002). Organizing relevant paper documents, e-documents and correspondence is a cognitively difficult and error prone activity that is complicated by a person's limited ability to keep track of unstructured information. Activities that keep knowledge information organized with predictable structure are directly relevant when cognitive activities are interrupted so that efficient resumption may be facilitated at a later time (Jones and Ross 2010). Public representation of these chunks, templates and frame memory structures are used to store encoded information and share it across team members for rapid retrieval and future problem solving. Patterns recognized by teams cue retrieval of memory schemas in a manner that is relatively insensitive to interference tasks (Hayes and Smith 2008). The mechanisms of distributed cognition position external memory structures that are insensitive to decay as important aids to reset progress following task interruption.

The role of external artifacts to re-queue memory models detailed in the literature on distributed cognition suggests that an IT project setting that emphasizes structured and comprehensive documentation may experience better team performance when faced with unexpected challenges relative to a setting that minimizes or discourages documentation. Where interruptions in the form of environmental disruptions may upset certain cognitive assets including individual memory, transactive memory and shared memory, there is a different dynamic for distributed memory in the form of external artifacts. Detailed protocol analysis of IT projects reveals external documentation artifacts to be resilient participants in the group cognition process (Flor and Hutchins 1992). The physical documents do not decay while the team is distracted by the interruption, but rather retain their value as heuristic cues. After an interruption the group may reference the documentation to leverage the cognitive effort expended during a design activity to quickly reset a working memory model. Examinations of software reuse demonstrate that anchoring heuristics can help restore shared memory models and situational control (Chong and Siino 2006). In the face of interruptions in a social setting, teams instinctively seek "written down cues" to facilitate the collective activity of task resumption (Ritterskamp 2011). Such restorative power is particularly beneficial in situations involving environmental disruption where the target task has not changed. Therefore:

> *Proposition 4: External documentation will moderate the performance impact of environmental fluctuations, such that the negative impact is attenuated.*

A.1.5 Plan-Driven Documentation as a Handicap

When faced with uncertainty, humans have a tendency to apply heuristics, or cognitive shortcuts. One such heuristic is Anchoring-and-Adjustment, in which original information biases subsequent decision processes (Epley and Gilovich 2006; Tversky and Kahneman 1974). When presented with a new situation, the anchoring bias explains the preference humans have for answers and solutions that are similar to the original information. The effortful process of creating these artifacts instills in their creator(s) a mental model that may be referenced in subsequent cognitive activities. As noted in the previous section, the artifacts and associated ideas facilitate the delivery of complex ideas across phases of a project and iterations of development. These artifacts can also be used to spread the mental model to additional team members through training.

In situations where requirements change and the team is faced with a different problem than that involved in the careful and effortful cognitive process that created the artifact, there can be a negative impact on quality (Allen and Parsons 2010; Tang 2011). When requirements change, the design should be adjusted, requiring another effortful cognitive undertaking. However, by anchoring on the heuristic cue represented by a formal specification, the cognitive process becomes biased toward an effort saving approach that prefers reuse of a solution to the wrong problem. The result is poor quality decisions and outcomes. Since existing documentation artifacts represent partial solutions, or guideposts encouraging candidate solutions, they are expected to incorrectly bias the decision maker in that direction. The Anchoring-and-Adjustment bias has been shown to be among the most robust cognitive heuristics across multiple domains and problem types (Furnham and Boo 2011). This bias has been demonstrated for problem solving tasks (Tversky and Kahneman 1974) judgmental tasks (Epley and Gilovich 2006) with supplied anchors (such as when documentation is handed to a new team member) (Block and Harper 1991; George et al. 2000), in situations of stress and uncertainty (Mosier 2008) for groups (Sniezek 1992) and with design artifacts for software development tasks (Parsons and Saunders 2004; Tang et al. 2010). Therefore:

Proposition 5: External documentation will moderate the performance impact of requirements fluctuations, such that the negative impact is amplified.

A.2 Case Study

Section 2.4 describes the case study design, setting and methods. Preliminary findings using conventional interpetivist methods are provided in the appendix. A separate Critical Realist assessment for theory building is provided in the main text.

A.2.1 Interview Findings

Case findings are reported below for each project. This provides details in the areas of planned risk mitigation strategies, unexpected challenges that emerged as issues, efficacy of mitigation strategies and emergent tactics employed by each project. In addition, documentation practices within the project are summarized. This is followed by a report of forensic analysis of status reports that provides broad cataloging of project challenges. Finally, a cross-case analysis is provided to assess the propositions used to guide the study.

A.2.1.1 Project A

Project A involved the custom development of a new system to replace an existing custom developed toolset in use for many years. This project was triggered as the organization adopted a new paradigm for handling a core business process. Table A-1 summarizes key findings exposed during Project A interviews. All IT participants described the project philosophy to be agile, and in particular noted that the project used the Scrum methodology. Most agile projects take a hybrid approach to selecting and tailoring practices for a particular project (Cockburn and Highsmith 2001), and this project is no exception. The overall project started with a requirements gathering effort "*initially with selected users from the business… Then going about and asking users in* *various sites and talking to users in person for all of the research sites, or most of them, to collect information and make users aware of the fact that the project would be starting*. {D}[14]" The user representative elaborated on the in-depth discussions with the local business analysts: "*We discussed it, took it apart…How we are doing it, what is required for us, really going in depth into our workflow, how we schedule status, what are the dependencies.* {U}" However, during project execution "*users were not present* {PM}" and developers "*didn't have that much direct interaction with users.*{D}" "*A lot of communication would be emailing with the users.* {PM}" Then "*every few weeks* [demos] *would be shown back to users while we were building.* {ITM}" Their involvement picked up near the end with an end-user populated acceptance team.

Table A-1 Project A: key findings

<table>
<tr><th colspan="2">Success Criteria[15]</th><th colspan="2">Performance</th></tr>
<tr><td colspan="2">Functionality</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">User Acceptance</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">Sponsor / User satisfaction</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">User adoption / Usage</td><td colspan="2">Met (mandatory)</td></tr>
<tr><th>Risk management strategies</th><th colspan="2">Unexpected challenges</th><th>Emergent challenge management tactics</th></tr>
<tr><td>Agile Project Management</td><td colspan="2">Staffing
(attrition & replacement)</td><td>Resource Change
(replace project manager)</td></tr>
<tr><td>User interaction
(through a proxy)</td><td colspan="2">Technology surprises
(moving target)</td><td>Social networking
(collaboration, negotiation)</td></tr>
<tr><td>Acceptance testing phase
(quality & performance)</td><td colspan="2">Scope creep
(new requirements late in project)</td><td>Re-circulate documentation</td></tr>
<tr><td>Experienced technical lead</td><td colspan="2"></td><td>Delayed project delivery</td></tr>
<tr><td>Steering Committee</td><td colspan="2"></td><td></td></tr>
</table>

[14] Throughout the Case study report, direct quotes are italicized in parenthesis and trailed by a designation of who made the comment: {D} = Developer; {ITM} = IT Manager; {PM} = IT Project Manager; {U} = User representative.
[15] Success criteria noted are those identified during interviews by project participants. No specific categories were suggested.

"When we were getting close to rollout, the last two months, we had the acceptance team, the users, working and testing the system on our integration environment.{ITM}" Another variant affecting user feedback involved the use of incremental development. Scrum sprints in this project generated working code used during orchestrated demonstrations to solicited feedback from stakeholders and user representatives. The system was not available for production use until after acceptance testing triggered a global rollout phase.

A.2.1.1.1 Risk mitigation strategies. Interviews revealed that the team viewed the agile development practice to be a mitigating strategy intended to manage scope creep by developing the system incrementally. The team started with a series of prototype *"wire-frames"*, progressed to functioning code, then a working system. The Scrum methodology was employed to manage the development effort through a series of sprints. This strategy was also expected to facilitate efficient system adaptation as requirements changed and became better understood. *"We knew this was going to happen, so the fact that the requirements change, and we had to change technology, a framework, etc., it wasn't a big deal. {ITM}"*

The early project activities emphasized sharing information with users. The IT developer observed "*the biggest risk is usually that the project doesn't do what the users really need. So what we did, we spent a lot of time, about half of the project time was spent making sure that the users understood what the project was going to deliver.*{D}" After the initial requirements gathering activities, functional requirements were relayed during regular interaction and meetings by the core team with user proxies and a steering committee. A nuance of this communication was its focus on a specific subset of users. "*We had a very specific user base* [involved during the project], *even though the final user base was a lot bigger… the way to mitigate risk was by focusing on a small, defined user community.* {ITM}" In addition to the user proxies, an active steering committee participated in setting goals, guiding functional requirements and reviewing progress.

The overall project had geographically dispersed technical sub-teams, users, and steering committee members. In order to facilitate effective decision making the "*two technical leads had most of the design decisions.* {D}" This reduced the amount of interaction overhead needed to reach decisions, with ideas and direction "*transferred during discussions.*{D}" Additional strategies were identified to mitigate the effect of users not co-located with the project teams. The team included project staff familiar with the legacy system. "*We had people on the team who were involved in the construction of the system we were replacing. We were able to do the design internally. …We knew the requirements of the existing system that needed to be taken care of or reproduced, and there were some additional ones we added. And that was worked out with the steering committee.* {ITM}" A related strategy was establishing user proxies. Because "*users were not present… our approach was to have IT colleagues, who are subject matter experts, perform the role of user proxies, and then on occasion to validate our design choices in the occasional user session.* {PM}"

A final strategy was to conduct planned acceptance testing with extensive user involvement. This activity was intended to catch both quality and performance gaps.

A.2.1.1.2 Unexpected Events/Challenges. The most dramatic unexpected challenge related to project management competence. As the team progressed from requirements gathering to design and implementation the project faced a crisis, with progress significantly slower than expected. IT Management noted *"there was a series of grotesquely missed deadlines… at that point we were three months behind schedule on something that was only supposed to take a year. {ITM}"* The delay was obvious to the whole team: *"at some point you realize that you are not going to be able to meet deadlines. {D}"* Management determined that the underlying cause was related to the organization and execution skills of the project manager. This issue ultimately led to the replacement of the project manager16. This and subsequent staffing changes introduced a new challenge of knowledge transfer and the need for an entirely rebuilt coordination and collaboration scheme.

At a more mundane level, technical challenges emerged in the form of minor system and programming "bugs" identified and managed over the course of the project. The team was also faced with managing infrastructure changes: "*as we were nearing rollout, our colleagues in global IT made some unannounced changes that almost stopped us in our tracks… this broke a number of applications, and we were able to have them roll that back.* {ITM}" A more serious technical problem took the form of a web-services library provided by an external project. The IT developer observed: "*we had technology issues driven by the fact that we were relying on web services, and those*

[16] The second project manager was interviewed as part of data collection.

weren't always available. We had issues with data quality and accessibility in some cases because they were being developed by another team at the same time. {D}" This was reiterated by the IT Project Manager: *"these web services themselves were changing frequently, because they were not mature. This was an unplanned flaw, the changing environment, which caused some difficulty for the development team.* {PM}"

Another surprise came in the form of late stage requirements changes. Despite the up-front efforts to identify all user requirements, and the use of incremental development with periodic demonstrations to collect feedback, the project experienced significant late stage requirements changes. In the words of the team: "*we were dealing with a global group and one site, and it turned out that practices were different on our different sites, but the representatives that we were dealing with didn't bother to tell us that. So we ended up with a number of fairly late stage changes that arose… we accidentally ended up talking to people on the other sites and realized we were working on a partial set of requirements. …what they told us is that an application we thought was more or less complete was missing a number of pieces and would cause them not to be able to do their jobs. It would jeopardize the success of the labs, which for us is a complete no-go, we cannot go forward with the application.* {ITM}" "*A form of scope creep, well-known scope creep, happened in this particular case, which we should have anticipated, but didn't really*. {D}" For example *"a number of additional systems that have direct impact on the users would also need changing, and this additional scope wasn't being analyzed properly.* {D}*"*

A.2.1.1.3 Effectiveness of a-priori risk mitigation. Early emphasis on requirements was not fully effective in transferring knowledge and understanding to the implementation team. *"We had an initial set [of requirements] which came from a business analyst working with the users directly. {PM}"* The primary mechanism to transfer requirements knowledge to the implementation team was through *"wireframes or mockups, and a sort of fairly brief set of requirements. {PM}" "Because the user requirements were done by a different team, the transfer of that knowledge, although the people were still in contact, it wasn't necessarily complete. {ITM}" "These are unregulated systems, so there's never any formal sign-off, internally or externally with the users. It's more informal. {PM}"* Limited user engagement during the project also attenuated the role and effect of prototypes, which were ad hoc and infrequent. The developer reported *"we put out a few prototypes, and I think those were used in relatively small areas by a few people…we could have done more to promote the use of these prototypes and been more active in collecting feedback that is the result of these prototypes. {D}"* In the end, these strategies were limited by the manner in which ongoing interaction with the user community was entrained. Latent and critical user requirements remained hidden until revealed by the user acceptance team. This surprised the core team and jeopardized project success.

The team employed a variety of tools to facilitate the Scrum process. This included the issue tracking software JIRA, and the add-on product Greenhopper, to manage user stories and tasks associated with each sprint. While effective in facilitating coordination at the tactical level, these tools were less effective in managing the disrupting impact of staff changes and externally provided library problems. These tools "*played no role whatsoever… they were unable to help entirely. The tools were not aware of the organizational environment or could not react to changing membership, they couldn't help on the organizational front and the tools were absolutely not linked to the technical environment in which the project's artifacts were being run. They could not help at all in managing the changing technical environment*. {PM}"

A risk mitigation strategy related to the tools was the Scrum development process itself. The iterative nature of Scrum sprints provided a built in means to expose challenges that needed attention. "*Based on performance of individual sprints and the planning process, which roughly gave an idea of what capabilities each sprint should be adding to the product, it was possible to judge whether the entire project was going to be jeopardized or not.*{PM}" Surprisingly, the technical challenges associated with the externally provided web services library "*wasn't really recognized as being a serious disrupter until quite some time later*. {PM}"

A.2.1.1.4 Emergent challenge management tactics. By far the most dramatic change management tactic was the management decision to impose change by replacing the project manager. The steering committee and IT Management intervened and *"replaced the project manager, we replaced the technical lead and we almost started from scratch. {ITM}"* The new project manager imposed strict Scrum operating behavior, which in turn led to friction with a technical individual contributor who was subsequently replaced on the team. *"The biggest piece of that was the project manager needed to be changed. That was the start of a series of dominoes falling… Then we* *brought in a different technical lead, and as soon as that happened, the entire character of the project changed…the rest of it kind of naturally flowed from that point. {ITM}"*

Other effective tactics were: escalation, collaboration and negotiation. "*There was quite a lot of disruption to progress, and the strategy was to communicate that disruption by escalation to management, and also through collaboration with those teams and projects who were initiating the change.* {PM}" The aforementioned infrastructure change that broke many applications was escalated and negotiated, causing the change to be reversed. Knowledge transfer through informal communication and coordination were key tactics to manage the staffing changes, supplemented by the project documentation and notes stored on the JIRA team site. The new project manager reconstituted the team with "*a fairly clear division of responsibility* {ITM}" where legacy team members received modified roles which "*gave them more responsibility.* {PM}" The new team members examined requirements documentation created earlier in the project, participated in code reviews as well as discussions with established team members regarding task assignments, design and architectural intent. A related tactic used to deal with technical challenges involved networking with other teams. The developer noted "*we actually had lots of close interactions with the other teams and exchanged quite a bit of information.* {D}"

A final tactic was to change functionality being delivered (reducing scope) and delay project delivery, which brought additional risk. When initially launched, the project had a certain scope of systems capability it was intending to replace. During the protracted delay of building this replacement system, the business continued to evolve and mature their practices. As global roll-out was delayed, new requirements emerged, creating a viscous cycle with the project team continuing to chase a moving target. As noted by the IT developer: "*we have a project that lasts a substantial amount of time…what we think we need to do changes along with the project quite a bit. And it was relatively extreme in this particular case.* {D}"

A.2.1.1.5 Documentation and direct communication. From a total time perspective *"about half of the project time was spent making sure the users understood what the project was going to deliver. {D}"* Certain strategies used in this project (such as iterative prototypes and demos) seek ongoing feedback. However, some compromises involving communications were made: meetings with users were not face-to-face because the project teams were distributed across two continents; interaction with users was indirect through proxies; key design and architectural decisions were not broadly collaborative but closely held by two technical leads, then transferred out to developers during discussions.

The conventional agile approach of documentation "*in the code* {D}" was followed here, along with code reviews that facilitated disseminating this knowledge across the development team. The project team supplemented communication by using a varied collection of external documentation artifacts. Supplemental artifacts were organized on a wiki-style web page "*which collects various aspects of the project – how the project works and also about the components that are implemented as part of the project. How they are installed, what the design decisions are that led us to a particular design, deployment instructions for the components. This included electronic documents* *with diagrams.*{D}" The issue tracking tool JIRA and the add-on product Greenhopper allowed the team to organize use cases, wire-frame diagrams, meeting minutes and all manner of attached documents. They "*allowed us to plan sprints and manage capacity and gather documentation, and also track changes to code.* {PM}" These documentation efforts were informal, without the content expectations and traceability associated with rigorous templates, review and approval that often accompany plan-driven processes. Furthermore, these artifacts were not shared outside the core technical team: "*The documentation* [users] *had access to was mainly the minutes of our meetings.* {U}"

The team employed multiple strategies to identify and communicate requirements. Early efforts to canvas a broad spectrum of the global user community were not effective in identifying certain key requirements. Similarly iterative demonstrations were restricted to a focused subset of user representatives, unintentionally excluding diagnostic and corrective feedback from other users. The core team's view of requirements was isolated with key requirements hidden until acceptance testing forced engagement of a broader community of users. This project experienced insufficient information and knowledge transfer across boundaries to the core team.

A.2.1.2 Project B

Project B developed a custom built system to replace an existing tool for scheduling resources and managing workload. It was part of a larger overall program to replace a suite of tools. The larger program involved multiple teams working on different components facilitated through bi-weekly global meetings where each team joined a conference call and demonstrated their component. Table A-2 summarizes key findings exposed during Project B interviews. All IT participants described the project philosophy to be agile, and in particular identified the Scrum methodology. Some local practices worth noting include: an up-front information gathering activity that generated a set of documented requirements in the form of wireframes and user stories for the core team; indirect user communication that was channeled through focal point representatives; and an incremental development process with periodic demonstrations for user feedback, with a single production deployment event at the end of the project.

Table A-2 Project B: key findings

<table>
<tr><th colspan="2">Success Criteria</th><th colspan="2">Performance</th></tr>
<tr><td colspan="2">Meets functional requirements</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">User Adoption</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">User Friendly</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">Replace Legacy Tool</td><td colspan="2">Not Met</td></tr>
<tr><td colspan="2">Schedule/time</td><td colspan="2">Not Met</td></tr>
<tr><th>Risk Management Strategies</th><th colspan="2">Unexpected Challenges</th><th>Emergent Challenge Management tactics</th></tr>
<tr><td>Agile project management process</td><td colspan="2">Management expanded scope from one site to multi-site global system</td><td>Defer functionality & write more custom code</td></tr>
<tr><td>Requirements gathering workshops with the extended team</td><td colspan="2">Staffing (assigned to multiple projects, attrition & replacement)</td><td>Networking to leverage organizational knowledge</td></tr>
<tr><td>Methodical technology search for code libraries (seeking reuse)</td><td colspan="2">Technology surprises (performance, complexity, moving target)</td><td>New/additional team members</td></tr>
<tr><td>Acceptance testing (Quality)</td><td colspan="2">Hidden requirements not revealed by iterative prototypes</td><td>Team workspace</td></tr>
</table>

A.2.1.2.1 Risk mitigation strategies. Several key risk mitigation strategies stand out. The first involved the use of the Scrum development process, supplemented by a distinct requirements gathering phase. Next was the technology screening process that preceded selecting key software libraries. Additionally, a user acceptance testing exercise was planned as a quality assurance measure.

The most extensive of these strategies involved the Scrum development process. "*It was also part of what we were trying to do and why we adopted an agile approach in the first place, because we are not terribly strong in business analysis…The approach itself, to put something in the users hands as quickly as possible and then pushing it and really kicking the tires in terms of uncovering missed features, you could say that was mitigating*. {PM}" The team employed a modified form of user interaction where communications were funneled through a focal point representative. The strategy was explained by the IT Manager: "*the way to mitigate risk was by focusing on a small, defined user community*. {ITM}" Other Scrum communication practices were more traditional, including frequent team meetings, time boxed development sprints and use of prototypes in the form of wire-frames and mock-ups.

The Scrum development process was preceded by a multi-month requirements gathering phase led by a business analyst and user experience team co-located with the lead user group. This was done to facilitate the communication and delivery of key user requirements to the globally dispersed development and execution teams.

When making technology choices the team pursued a methodical exploratory search to identify the most promising third party toolsets, then assess through prototyping. "*We would start with a broad sweep to see what's out there, find five or six possibilities by talking to other people with similar experience and seeing what they* *recommend. Going to Google and seeing what can do what we need. Then we would pick a handful take a more detailed look and then prototype with one or two. So that's how we tried to mitigate the risks up front in terms of choices we made*. {PM}"

A final risk mitigation strategy involved extensive user acceptance testing to assure system quality. According to the project manager "*It wasn't until we got to the end where it was more formal with the users saying 'yep, it's good to go, we're satisfied'*. {PM}"

A.2.1.2.2 Unexpected events/challenges. Four types of unexpected events and challenges emerged during this project. These were rooted in misalignment on goals and priorities, staffing and resource problems, technology surprises related to the software libraries developed outside the team, and hidden requirements that became late project scope creep.

The alignment problem involved a goal conflict between the two most influential members on the core team. "*The project manager, wanted to deliver this project in a very timely manner …Whereas our technical lead had other areas he wanted to achieve…there was an initiative across the organization to make a library of widgets that we can reuse. So there was this clash, because the technical lead wanted to use our project to contribute to the wider IT organization, particularly the reusable widgets initiative. …Because of those two conflicting visions, we had a lot of clashes during our planning meetings*. {D}"

Staffing challenges in the form of multiple concurrent assignments, attrition and replacement had unexpected consequences for this project. The situation contributed to the team's frustration interacting with their infrastructure service provider department. "W*hen it comes to integrating your system with the wider organization network, getting those people's attention and those resources when they're also very busy, it's very difficult.* {D}" In a different situation, conflicting assignments eventually led to the replacement of a core team member. The subsequent disruption associated with integrating a new member was substantial. "*It really became clear to the rest of the team that this new person wasn't a big subscriber to Scrum*. {D}" "H*e simply couldn't cope with the Scrum process and he wasn't particularly used to working as part of a team…* [after] *about a month he was then moved off the team. That was unexpected, quite a bump for us.* {PM}" "*That was a huge disruption, because of the time it took to bring someone new on board…It actually took us several weeks to get back to our original speed just because of the time it takes to onboard someone and after the fall out.* {D}"

Multiple unexpected technology challenges emerged. The first involved immature libraries being developed elsewhere in the organization. This library "*was being actively promoted and pushed, but it wasn't quite ready for prime time. We tried very hard to learn and implement that platform, and ultimately we backed out of it.* {PM}" A similar challenge involved third party web services frameworks that turned out to be more difficult to implement than expected. The team had "*big success early, but then struggled later to implement what sounded like simple features*. {PM}" Another such challenge involves performance problems with a "control" library. "*We had massive issues with performance. We spent a lot of time debating whether to switch* *to a different third party control, and we spent a lot of time fine-tuning the performance…. So that was probably the biggest impact to our timelines and the biggest unexpected event*. {PM}"

Another unexpected change came in the form of a strategic scope change. The project team was originally assembled to replace a local tool in one site. Approximately five months into the project, with the core team actively engaged in implementing a solution, there was a management decision to expand the scope and position the product for global deployment across multiple sites. The system needed to accommodate an expanded set of requirements. This resulted in a significant delay in the overall program. Meanwhile, the local user organization continues to evolve and mature their existing tool, providing the project team with a moving target. "*Because of the delay we are still continuing improving our tools…we still have some additional features from time to time over the last two years…The big danger I see now: the* [project team] *is chasing a moving target*. {U}" The team progressed through a series of incremental development sprints and prepared the sub-system for final acceptance testing and deployment to production. At this late stage additional requirements were exposed that surprised the team. The team assumed the iterative prototype process and periodic feedback from user representatives would reveal all requirements. "*When we got into user acceptance testing we found additional things we needed to build, and then it took two months extra… we found during user acceptance that we needed synchronization between our tool and the existing tool... So we ended up building a whole synchronization process that wasn't in the original set of requirements.* {PM}"

A.2.1.2.3 Effectiveness of a-priori risk mitigation. Neither the up-front requirements gathering nor the iterative prototype process accomplished the goal of capturing all user requirements. Despite the extended effort to canvas users, key requirements remained hidden until late in the project. This was noted by the project manager: *"We are not terribly strong in business analysis …In terms of anticipating the additional features what we found when we really got in with the users, we didn't really anticipate that too well. {PM}*" The other up-front exercise to methodically search for component and framework software libraries was also a disappointment. As noted above, two technology surprises involved third party libraries that failed to perform as needed. *"We invested so much time in [the control library] and we weren't sure if we were being objective, whether to stick with it and solve it, or to back out and choose another control. {PM}"* After changing frameworks the team realized that it wasn't good enough *"so went back to the drawing board and chose a different framework. {ITM}"*

The agile approach had a variety of effects, starting with the cadence of iterative sprints that helped signal technical issues. "*Through Scrum itself you see these things very quickly. Maybe you'll hear about it at the daily stand up, this thing was supposed to be done yesterday, but we found a bunch of new problems… The other place is after the sprint itself you find out it took much longer than expected, or if there's still problems with it. So that's really where I was hearing about it and discussing it with the team.* {PM}" Unlike technical problems, the iterative process was ineffective in signaling staffing problems. "*The personnel issue was different. We know how many* *hours people had every week, every two weeks for the sprint. It took us a while to consciously realize somebody has been working 18 weeks on this, and they've only spent 27 hours on ticket, so this was more of a slow boil problem that we didn't realize until later. So that was something we didn't have a mitigation strategy for….It had reached a critical stage… once* [the project manager] *asked for a new resource, that's when we went back and did the math and realized people who had been assigned hadn't really been working on the project*. {ITM}" Later the on-boarding of new team members was facilitated by the communication mechanisms of Scrum. "S*crum kind of helps us with these retrospectives every two weeks where the whole team gets to speak up about what could be better, what they need…where people with knowledge would share it, and …several sessions to understand the requirements… It takes time, but eventually the team really gels and has common understanding about the process and the products and the technology.* {PM}"

The agile emphasis on local decision making by empowered teams in combination with globally distributed teams at the program level added technical complexity. "*We ended up with completely different web technology stacks. Our site ended up using WCF AJAX, and another site used more ASP.NET and VC, while a third site ended up using totally different Java, J2EE… we didn't have an up-front technical design, and it was because we wanted to go with the scrum methodology of building on the demo.* {D}" A side effect of the agile process is a substantial amount rework. *"For the first few sprints we just built a static page with no backend database. It wasn't until sprint 5 or 6 that we started thinking about the database stuff. And as a result we did a* *lot of refactoring because of no up-front design, but again, I think that was intentional.* {D}"

Organizations are often drawn to the agile approach to proactively engage scope and requirements changes. This project held a similar view. "*We would have user stories which would just have a sentence or two, and then it was up to the developer and* [project manager] *to communicate more verbally... so it was really a back and forth with lots of daily communication and watching it evolve and tweaking as we go*. {PM}" However, as implemented in this project the Scrum process failed to reveal important requirements until late in acceptance testing. "*We started out with prototyping and giving some demos, and the users were always very happy with the demos, and they'd always say 'Oh yeah, it looks great, I like it!' …lots of people look at demos and say 'that's great', but then when they actually start using it they'll give you a list of a hundred things that aren't quite what they want, or aren't what they want at all, and that was a surprise to me, the difference between the early response to the demos and the more detailed sessions. And so a few fairly big things came out, such as they needed workflows… And then we got into more detailed user acceptance, that's when we realized we would need to keep our scheduling tool synchronized with their tracking system tool… This Synchronization process took me by surprise. I didn't expect that. I didn't know about it, hadn't thought about it. It hadn't been mentioned.* {PM}"

Acceptance testing proved to be very effective. In particular, a model test practice was singled out. "*Our user experience team likes to get a user who has not been involved in requirements gathering to sit down in front of the product and try it.*

And they'll give this user no information at all. They'll basically tell the user to pick an activity that you do today and try to do it with this tool, with no guidance, and then we watch what happens. …This usability session with an uninitiated user gave us a ton of information, an awesome learning experience for me. That's something I'll carry forward. {PM}"

A.2.1.2.4 Emergent challenge management tactics. Several tactics were employed to deal with technology surprise. Initially the team resorted to *"a lot of work around, custom code. {PM}"* The developer elaborated: *"we spent a couple of sprints where we just did technical death sprints, integration work, no new features, just to get all of the pieces working because we were changing these things midstream. {D}"* In parallel with actions that abandoned code reuse, the team jettisoned functionality. *"We actually had to back off a few features, they were kind of nice to have, but we had to improve the performance. {PM}"* The project depended upon information from outside the core team finding its way into the team. *"We found someone else on a different project team who had downgraded to a previous version of that control, and it solved that problem… And that was pure luck as it happens. The individual who ended up helping us solve the problem happened to be the developer who sits next to me. {PM}"*

Another tactic used in this project was to add resources beyond simply replacing those that left. The project manager noted: "*I fought for more resources and this project team space that we now have, and I got it.* {PM}" Integrating those resources required a lot of time and effort. "*We spent a lot of time just walking them through, one-on-one, …there was a period of time where I wasn't doing any development work because there* *were literally four people where they all were trying really hard to get up to speed.* {D}" These efforts were supported by the project manager's efforts to secure a "*workspace where the entire team would sit together. …So we now have a team space on a new floor where the entire team sits together, they're all within ear-shot, they can all talk to each other and have stand-ups in our team space*. {PM}"

A.2.1.2.5 Documentation and direct communication. Despite Scrum's practices that encourage regular interaction, some key requirements were not exposed until final acceptance testing. A tactical view of the business processes was internalized by the team, without developing an appreciation for the big-picture. It starts with "*user stories [that] were quite brief, just a couple of sentences of what the feature was.* {PM}" These are interpreted in the context of a short duration Scrum sprint where "*the team, they only see a chunk of it for each sprint, it's not very strong at giving them the big picture*.{PM}" The sprints culminate in a developer orchestrated demonstration focused on a limited piece of the business process. "*We had live demos, and at that point they were not providing the right feedback because it was a guided demo. The guided demo was very different from hands-on* [use]…*It wasn't until it was really in their hands that we discovered a lot of things*. {ITM}" Terse user stories conspired with short development sprints and IT run demonstrations to allow crucial requirements to remain hidden.

Early documentation was conventional lightweight agile artifacts and conversations. Initially "*the documentation process was extremely ad hoc, basically code reviews where the team shared knowledge amongst the developers informally.*

{D}" Beyond the wireframe prototypes used during early demonstrations, there was no emphasis or intentional effort to share documentation artifacts outside the core team. Users noted "*I saw wireframes. Real technical documentation or implementation [plans] —I never saw those*. {U}" The team executed the iterative Scrum process with a faith it would methodically expose all relevant requirements, only to find significant requirements during final acceptance testing. The core team's mental model constructed and sustained by input from a limited set of user representatives omitted relevant information from the extended user base. Informal and unstructured communication methods used by the core team to interact with outsiders allowed a gap in information exchange.

A lesson learned from this project now being applied to follow-on projects is that ideas captured in simple user stories "*were not sufficient. So we started working closer with the users… We started making those user stories really quite detailed with very explicit requirements in them.* {PM}" User stories are being expanded to "business stories" that give an overview of the product and a big picture view of the user process, including a better understanding of the scientific methods the lab is working with. "*The components that we're moving to are much more complex and there is much more* [business process knowledge] *involved. …We'll put together a set of wiki pages that provide links for further reading with a good summary and background, and it has pictures of file formats, and plate layouts, and things like that to help with understanding…we'll use wiki both in high level and big picture stuff, but also have some pages that really drill down to the nitty-gritty, and have lots of sample data files* *available for those developers to look at and to really explain things in more detail than what we would in our user stories*. {PM}" What was once tactical task oriented information has subsequently matured into big-picture business process knowledge-bases, supplementing with precise design and implementation details. In so doing the team is adopting not only a detailed documentation artifact, but the rigor of a methodical interaction process with their user community that extends the team's baseline process.

A.2.1.3 Project C

This project was triggered as the organization sought to automate an existing manual business process for invoice receipt, processing and payment. The new system implemented an off-the-shelf software package from an established vendor. A portion of the operational processing (digitizing paper invoices) was outsourced to the vendor, followed by a handoff of data to the newly implemented system for value added processing and loading to the organization's established ERP system. Table A-3 summarizes key findings exposed during Project C interviews. Participants described the project philosophy to be plan-driven and followed the organizations PMBOK based procedure. Some practices of note for this project include: close interaction among users and developers and near uniform agreement on qualitative success criteria. Iteration used with this system implementation involved program level iterative delivery, starting with a non-production pilot implementation project (phase 1) followed by a sequential series of projects that coupled functional enhancement with new site implementation. The initial production site received base invoice receipt capability during phase 2. Phase

Table A-3 Project C: key findings

Success Criteria	Performance
Customer satisfaction	Mixed for different business functions
Standardization	Mixed
Functional objectives	Met
Process satisfaction	Met
Involvement of all parties	Mixed

Risk management strategies	Unexpected challenges	Emergent challenge management tactics
Active steering committee	Quality failure with service provider	Manual work-arounds
Iterative delivery (pilot then phased rollout)	Technology surprise (regression, interfaces, performance)	Networking across teams, engaging experts
Workshops with extended team (seeking alignment)	Scope creep (new requirements, hidden requirements)	Trial & Error trouble shooting
Global standardization (no site specific customization)	Staffing (assigned to multiple projects, attrition)	Late project design changes
Transference (outsource certain operational processes to a 3rd party)		War-room meetings (brainstorming)
Experienced project manager		Special topic task-force
Testing (Quality) and Training		

3 coupled a significant enhancement to automate the payment process with rollout to an additional site. Implementation of phase 3 was examined in this case study.

A.2.1.3.1 Risk mitigation strategies. Several risk mitigation strategies were employed for Project C: the establishment of a steering committee with regular meetings; assignment of an experienced project manager with business domain knowledge; a kickoff workshop with the extended team including multiple distinct user communities representing different business functions; formal requirements documentation and approval for alignment; a rigorous requirements driven testing exercise for both quality and performance; proactive user training; transferring certain operational tasks to a third-party who specialized in that service; and the aforementioned phased delivery. Perhaps the most important risk mitigation was the strategy to deliver a single standardized global system. This objective guided the team to avoid software customizations and drive a change from locally distinct business processes toward adoption of a unified global process. This strategy had obvious efficiencies for IT, but was also supported by high level business leaders who were relocating local country specific processing centers to a regional service center. At the regional service center there were advantages to executing a single process that reduced complexity with an expectation for higher quality and lower effort. A standardization objective existed independently for the infrastructure department that was focused on executing a virtualization strategy to move systems from dedicated physical computers to an on premises "cloud" computing environment.

The business strategy to consolidate staff into a regional service center involved the turnover of many business professionals. This was particularly painful where high performing employees were not willing to relocate from their native country to the regional service center. "*We were dealing with people who were actually losing their job… some of these individuals weren't going to be moving with the shared service center.* {PM}" Retention bonuses were used to minimize this risk and set a known date through which most of the key resources would be available.

A.2.1.3.2 Unexpected events/challenges. Several challenges emerged in the course of this project. A key service provider was not able to meet quality objectives, which resulted in unexpected data errors being fed into the new computer system. Other technical challenges involved computer and network infrastructure problems in the form of disruptively slow performance: "*the production system ground to a halt…it depended on which server* [users] *had signed onto…and all of a sudden all of them came down and nobody could access the system.* {D}" On another front, interfacing errors revealed a local country specific character set constraint that forced customizations inconsistent with the global standardization strategy: "*we encountered a Polish character set issue that had never been identified.* {PM}" In addition to the character set differences, other business processes were tied to local cultural constraints or country specific laws.

Staffing and resource challenges existed in multiple forms. Some staff turnover was anticipated, such as attrition associated with the business move to a consolidated regional service center. Other resource challenges had unexpected consequences. Both business and IT participants had multiple assignments involving other projects or day-to-day operational responsibilities. "*That was one of our biggest challenges in dealing with the timing of the project – it always seemed to hit where people were gone two or three weeks at a time. … It was very difficult to keep our finance users engaged because they had full time jobs and these projects were on top.* {PM}" The staffing challenges led to delays completing project tasks and necessitated ongoing training of a changing user population.

A.2.1.3.3 Effectiveness of a-priori risk mitigation. The a priori risk mitigation strategies identified in Table A-3 were somewhat helpful. The experienced project manager was able to deflect some of the natural scope creep by negotiating adherence to the global strategies. The project manager was also instrumental in activating new tactics for many of the unexpected challenges. Manual workarounds were coordinated where appropriate to allow the business to function in areas the standard system could not accommodate. In addition, effective cooperation with an experienced developer led to innovative technical solutions for problems such as the character set surprise.

Attempts to transfer risk to a third-party service provider for a portion of the operational process delayed exposing problems. Data received from the third party was incomplete or incorrectly indexed. Because the service provider had a track record that suggested a certain maturity, the team did not apply effort in analyzing or addressing the risks that had been transferred until very late in the project, by which time they jeopardized both project success and business operations.

The steering committee provided oversight and endorsement to allow the project team to pursue non-standard designs to accommodate late project requirements changes. The steering committee was less effective in maintaining participation levels and process alignment across distinct functional user teams. Where the early project workshops achieved strategic alignment on goals and objectives, this alignment was not maintained down through the organizational hierarchy to the working level teams where global standards that complicated local processes were resisted. As noted by the user representative: "*that was a problem from day one, people within the markets didn't have the same priorities as the people that were working at implementing this project…the people in the market were more preoccupied with closing their budgets for the year because it was the middle of budget time… it's the business in the end who are* *ultimately responsible for it to work, and in this case the business wasn't really interested.* {U}"

The standardization strategy was sometimes successful in managing requirements change. In other situations it either triggered or amplified the risk. With each locally unique requirement that emerged, IT's first response was to push back and expect the business project manager to enforce adoption of the standard global process. "*One that specifically comes to mind is workflow levels, we do multiple levels historically… that's not the way the software worked, and so they adjusted.* {ITM}" As some of these requirements were actually constraints (e.g., country specific character sets, or laws requiring in-country storage of certain documents), this approach risked delaying the issue until it had an impact on production operations.

This project attempted to execute a strategy that did not always fit the target environment. Strategic direction was set without full awareness of, or appreciation for, technical, cultural and legal constraints. The implementation teams worked faithfully to execute a global standard as they planned tasks, timelines and staffing: "*we had a stated goal: we're not going to change the software.* {ITM}" However, a gap existed between strategic stakeholders and the users who performed business tasks on a day-to-day basis. "*Looking back at that workshop, the buy-in from the business probably wasn't where it should be… you get from different countries 'that's not going to work for us, if you want us to use this system, then you have to change'.* {D}" Eventually compromises made to accommodate local requirements had a ripple effect as development and regression testing plans had not anticipated the volume of system changes that followed.

The team "*didn't anticipate all the cases that we were going to encounter, so it was a bit of a hassle to get everything under control.* {U}" The plan did not expect this workload, inducing a team response that "*was very reactionary.* {D}" These changes introduced bugs to existing system features that jeopardized project timelines and business operations even as unplanned resources were identified and engaged.

Problems emerged with regard to infrastructure standardization as well. The nuances of this particular software package, and the geographic realities of the usage setting, interacted with the virtual hosting platform, resulting in unacceptable system performance. Risk assessments and planning did not account for the possibility of performance issues as "*there was clearly no server performance testing done at the start anywhere… all the way through we had issues with performance, and that was constantly being addressed after the fact.* {D}" The lack of a plan for tuning the infrastructure is highlighted by the trial and error nature of the response. "*The first thought was there must be something wrong with the virtual server, so they put it back to the physical server, …then they moved one virtual server back in and it wouldn't be a problem, so they'd move another one back…so then we're not getting any problems, we're live again, and then a week later something happened, and they come grinding to a halt again… It just seemed to be from an infrastructure point of view it was not clear how problems should be solved, so we should try everything and hope it works.* {D}"

A.2.1.3.4 Emergent Challenge Management Tactics. Faced with unexpected challenges that jeopardized project objectives, the team employed several reactionary tactics. For example, team members worked longer hours: "*I had to stay every day until* *more or less 10:00 in the evening for the first three or four months*. {U}" Another tactic was to bypass limitations in the new systems and resort to manual procedures until system changes could be made. Yet another approach was to transfer the problems to the software vendor. The team went "*back to the vendor and saying our requirements need to change and we need to work with you to implement something that will work for us and others of your customers*. {ITM}" There were several "*additional* [requirements] *we added. And that was all worked out with the steering committee*. {ITM}"

Some of the responses were more systematic, such as the knowledge acquisition and learning tactics employed to manage the technical challenges. This started with simple information gathering and organizing in the form of "*an issue log to capture all of the issues*. {U}" The team reached beyond its borders to supplement its knowledge by networking with other technical teams and outside experts. "*We discussed it with the Business Intelligence team, because they have the same kind of infrastructure*. {D}" In order to fit a purchase price variance scenario to the standard software, the project pulled together a special task force: "*We had to pull in some people who weren't directly involved in the project… we had to escalate it and ask for some help.* {ITM}" In the end, communication and coordination beyond the team facilitated the creation of solutions.

War-room brainstorming meetings provided key crisis management to resolve infrastructure issues. "*We had to bring people together from multiple sets of expertise and sit in a room together and talk through every point of connection that takes place and understand what's going on in the system… let's look at the full picture and try to* *address it, …put our brainpower together… when you get people together in a room with a common goal of correcting, and not in a blame game situation, usually good things can happen.*{ITM}" Ultimately the team "*had to change original designs for the infrastructure to get the performance that was desired.* {ITM}"

A.2.1.3.5 Documentation and direct communication. Informal coordination and direct communication took many forms. This began during the earliest days of the project with the kick-off workshops in which users from multiple sites gathered in face-to-face meetings with the core team to define processes and requirements. Later when challenges emerged, the core team began networking with other technical teams, organized war-room brainstorming sessions and escalation of revised plans to the steering committee. These behaviors reveal the effort applied to communication and coordination beyond the core team.

As is common for plan-driven projects, formal documentation was also present. During the face-to-face workshops discussing business processes and requirements, there was "*a scribe or note taker to capture the points…The team would transfer handwritten notes back into specific user requirements documents as part of the project deliverables.*{D}" The workshops also generated business process flow charts which were referenced throughout the project and continue to provide guidance after putting the system in production. During project execution the team used a formal statement of work to describe requirements and designs contracted to the software vendor for development. Tasks and schedules were managed in Microsoft Project. All documentation spanning formal project deliverables, informal notes and working records such as the issue log are organized on "*a global website… a central, global location for all communication, all documentation…*{PM}" Finally, at the end of the project there was a formal "*system acceptance document.* {D}"

A.2.1.4 Project D

Table A-4 summarizes key findings exposed during Project D interviews. This project implemented a custom configured third party package to replace a legacy warehouse management and shipping system at end of life. All IT participants described

Table A-4 Project D: key findings

<table>
<tr><th colspan="2">Success Criteria</th><th colspan="2">Performance</th></tr>
<tr><td colspan="2">Milestones</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">Functionality</td><td colspan="2">Met</td></tr>
<tr><td colspan="2">End-user training</td><td colspan="2">Not Met</td></tr>
<tr><td colspan="2">No business disruption</td><td colspan="2">Not Met</td></tr>
<tr><td colspan="2">System reliability</td><td colspan="2">Met</td></tr>
<tr><th>Risk management strategies</th><th colspan="2">Unexpected challenges</th><th>Emergent challenge management tactics</th></tr>
<tr><td>Active steering committee</td><td colspan="2">Intervening priority project</td><td>Delay project delivery</td></tr>
<tr><td>PMBOK program with iterative site-by-site delivery projects</td><td colspan="2">Staffing (attrition & replacement, competing assignments)</td><td>Adding resources, including experts</td></tr>
<tr><td>Formal risk assessment & business continuity plan</td><td colspan="2">Excluded use case (hidden requirement)</td><td>Tiger Team (for inventory audit)</td></tr>
<tr><td>Extended team workshops & frequent user interaction</td><td colspan="2">Data Integrity</td><td>More Training</td></tr>
<tr><td>Rigorous testing & user Training</td><td colspan="2">Ineffective user training</td><td>Unplanned expert consultation</td></tr>
</table>

the project philosophy to be plan-driven, using the organization wide project management process modeled on PMBOK. Some practices followed in this project include daily user interaction and program level iterative delivery strategy for system rollout. This project implemented the system to the fourth in a series of distribution centers. The scope of Project D included the addition of significant system capability as the previous sites were much smaller and required only a subset of functionality. Concurrent with the new computer system an important business process change called "cartonization" was implemented.

A.2.1.4.1 Risk mitigation strategies. Several key risk mitigation strategies stand out. The project followed a program level incremental deployment model where simple editions of the system were deployed to small sites, followed by the addition of complex features to support deployment to larger sites. Each deployment was a stand-alone project within the overall program. Other strategies include organizing an active steering committee, workshops with the extended team to get agreement on requirements followed by sustained user interaction, distributed computer architecture to allow scaling to large workloads, and an extensive testing and training activity.

Several strategies were active before the implementation project for this site was initiated. A form of iterative delivery was underway with a subset of the system already deployed at three smaller sites. "*The majority of the requirements were already built into the system… there were maybe 20-30 new requirements that had to be added.* {ITM}" Consistent with PMBOK recommendation, a formal risk assessment was documented with a business continuity plan that allowed transactions to be routed to an alternate warehouse in the event the new system could not perform adequately. In addition, a steering committee was established to provide organization level alignment, secure resources, arbitrate strategic issues and make decisions on proposed plan changes. While the project was underway the *"steering committee met on a monthly* *basis to keep everyone informed of where we stood on milestones, and if we were going to miss this date, or if we had any concerns on the date.* {U}"

Additional risk mitigation strategies were activated as the project ramped up. "*IT sat down with the users, before we did anything with the product. We defined use cases and designed how the system would flow*. {D}" Through a series of user workshops the core team pursued an understanding of both the as-is and to-be business processes. Following these workshops the implementation team maintained frequent, even daily user interaction. "*One of the things to mitigate our risk was to involve the users from day one. In designing the flows and designing the test scripts and designing how the software will be used.* {D1}" In addition, the business appointed a "*business project manager with responsibility for training of the users, assembling the testing teams, budgetary commitment and workforce commitment*. {ITM}"

Extensive "day-in-the-life" testing was a planned mitigation strategy for both user training and system quality. The team invested about two months working with users to build test cases used for formal validation testing. These tests were executed in a formal "conference room pilot" environment to make sure all system enhancements were functioning properly. In addition "*they went back and tested a lot of the existing functionality, so we performed a lot of regression testing.* {D2}" When development was complete, a full week of transactions were extracted from production with users engaged to simulate those transactions on a test system one day at a time over the course of seven weekends. This testing was "*not only to exercise the system, but to help educate and train and get the users comfortable with the new system.*{D2}" In addition to the day-in-the-life simulations, the project conducted classroom sessions with a professional trainer who "*would talk about the* [task]*, and then walk through it on the screen, showing each segment of the process visually on the screen, and then have them do it themselves.*{U}"

A.2.2.4.2 Unexpected events/challenges. "*There were minor technology issues, but they were fairly minor* {D1}" and "*rectified fairly quickly*. {D2}" Of a more serious nature were staffing challenges including turnover among core team members. A high profile example of this involved the project manager, who was reassigned mid-project to a different activity within the organization[17]. This reassignment "*was completely unexpected. He was expected to play the role of project manager for the entire project*. [He] *was an extremely good project manager. To lose somebody of his caliber, you lose a wealth of experience and knowledge*. {D2}" In addition, many participants had other project and production support responsibilities that coexisted with their project role. Consistent participation beyond the core team was a challenge as "*the users came and went. Their time wasn't allocated 100% to the project, they did day-to-day jobs*. {D1}" "*Users had to be taken off their normal day-to-day positions to wear this project hat, and many of them had not been privy to this experience before, so it was completely new for many of them*. {D2}" This situation applied to IT personnel as "*the deployment team also had support duties for other sites*. {D1}" An extreme example of interruptions occurred after passing the project midpoint. The company decided on short

[17] Because the project manager departed mid-project, and the role was shared between a new technical lead and the IT manager, this interview processed involved two developers {D1} and {D2}, in addition to the IT manager {ITM} and user {U}. The combined input of four interviewees provided good triangulation and diverse perspective.

notice to replace its primary parcel carrier. This project was suspended and staff temporarily assigned to the interrupting project. The schedule delay was compounded by holiday and end-of-year events once the project team was reactivated.

An entirely different type of unexpected challenge emerged in the form of excluded use cases. One common use case from the legacy system involved workarounds that individuals would employ to complete tasks expeditiously. "*They had figured out tricks by which they could cheat the system to pick whatever inventory they wanted. The new system really locked down on first-expire first-out.* {D1}" In addition to locking down these tricks in the new system, the project team excluded this use case from their testing, training, and data migration plans. The result was a significant data integrity problem as the data loaded to the new system was not consistent with locations and lot IDs in the warehouse. Furthermore, the new system prevented the user from working around the dilemma; the business process could only proceed with the specified lot. When users went to use the system, product was not available where the system said it should be. "*The inventory in the warehouse was a mess, and …that was not anticipated at all.* {D2}"

Excluding the work-around use case from testing and training contributed to another unexpected challenge. User Training was a failure. Day-in-the-life testing was done "*without physically moving the product. I think that was one of the biggest problems we had. In the day-in-the-life testing it's very easy to just scroll down the different lots that were available. However, when you go live, you have one lot in front of you and it doesn't accept it because of dating issues.* {U}" The users "*didn't know* *how to do their job. When people were asking how to do that, and it's the same thing that they did in the day of the life test, it's like they almost went brain dead. It was like it was a different person in go-live than it was in a day-in-the-life test... we found out the users were just going through the motions, that they hadn't learned in the training, they didn't really understand why they were doing those things.* {ITM}"

A.2.1.4.3 Effectiveness of a-priori risk mitigation. The combination of data integrity problems and insufficiently trained users had an immediate impact on warehouse performance with shipments backing up quickly. This in turn triggered activation of the business contingency plan to process a significant number of orders from an alternate warehouse while the problem was addressed. However, this did have a real cost from higher shipping fees and staff overtime.

Several other strategies proved effective as well. The team and iterative project rollout approach was effective in managing technology challenges. "*There were minor technology issues, but they were fairly minor, we were able to get around those*. {D1}" The steering committee served an effective role in vetting and approving plan changes when the team faced a competing project to switch parcel carriers.

Some risk mitigation strategies were less effective. Continuous and intensive user interaction, including extensive testing and training, was not successful in exposing the data integrity problems or achieving adequate competence with the new system. "*When we went live in January, all of the training that we had done, and all of the time we put in, seemed to go away, and it was like the first time they'd seen the system. So* *from that perspective it was a failure. When that switch came on they forgot everything they had learned, and that was very very frustrating.* {U}"

A.2.1.4.4 Emergent challenge management tactics. Addressing the challenges often involved adding unplanned resources, including outside consultants and experts. "*We brought people in from* [corporate headquarters] *for several weeks to start working through the problems with the people. …IT and the project brought in consultants to assist with not only the setup, but also the testing and solving problems onsite, and the training*. {U}"

When the IT project manager was reassigned IT management responded by replacing the project manager with an analyst possessing extensive experience in the underlying technology. "*He did a good job from an administrative standpoint, but when we got into the latter sections of the project where it was more political, when there were more negotiations that had to be made, then* [the IT manager] *stepped back in and filled that role. It was the relationships, someone had to go to the steering committee and present the pros and cons related to some options as far as negotiating or picking a new go-live date.* {ITM}"

Faced with the interrupting carrier change project, IT management agreed to suspend the system replacement project. "*We had to pull the entire IT project team off for a month and have them work on that change.* {ITM}" In addition, the steering committee was engaged to "*negotiate a slide of the date for the go-live.* {D2}" During the project suspension "*there were still some things that the business was able to do during that month, so it didn't come to a complete stand still. There was still some* *reviewing of test scripts and stuff that they were able to accomplish while IT was off making this other change.* {D2}"

Repairing data integrity required significant effort, including a special sub-team to perform a full inventory audit: "*We had four people taken off of regular duty to count inventory and deposit it in the correct locations. So that now, when a pick happens, they know where to go get it*. {D1}"

The team addressed the apparent failure of training during the testing phase with "*a lot of on-the-job training*. {ITM}" The professional trainer "*went out on the floor, hands-on while they were working through it, trying to help them through the process. …To get them comfortable with the* [new system] *as well as the process that had been tested and trained and worked on prior to go-live*. {U}"

A.2.1.4.5 Documentation and direct communication. Core team members treated active and frequent communication with users as crucial for managing deviations during the project. "*You have to be in constant communication with the business and your management team to ensure that when those events do arise, that you can communicate the impact and adjust accordingly… Communication, I believe, is the key*. {D2}" On this project, communication between the core team and the extended user community was an everyday occurrence. "*We had regular meetings everyday with the warehouse, where they tell us what issues they're facing*. {D1}" "*They were involved every step of the way within the project itself*. {D2}" An activity singled out as particularly effective was the "*status meetings at which you're talking with the business* [and] *pulling the core members together on a weekly basis*. {D2}"

The team captured information and knowledge flowing and moving between participants into a healthy array of documentation artifacts including risk assessments, mitigation plans, requirements lists, functional specifications, technical specifications, and validation testing plans. Starting with the earliest project workshops, flow diagrams were used to capture processes and further discussion. The working documents were posted to serve as a gauge measuring convergence on a shared understanding. "*These flow charts came in and we put them on the board and said this is how we see it moving, and we wouldn't get off of that particular function until everyone was in agreement.* {U}" In many cases these ideas were "*well documented and signed and approved* {ITM}", serving as evidence of alignment between the core team and stakeholders. "*Designs were talked over with the business and then following the procedure we* *typically go through, the requirements for those were given to the analyst who would write up a Functional Requirements specification, and that would be given to the developer.* {D2}" Key information, including "*use cases, were documented as flow charts and stored in the eRoom.* {D1}" The formal documentation was more than a trophy, it served as reference material, transporting ideas across time and space. For example, as the team reassembled following the parcel carrier switch, an effort was made to resynchronize everyone on the revised targets. They "*re-circulated a project plan that had the negotiated dates on it so that everybody was in agreement* {D2}".

A.2.2 Secondary Data Collection

The primary data collection activity involved a deep dive investigation of four specific projects. While this approach provides a detailed view of the tactics project teams employ when unexpected changes emerge, it offers only a limited sample of the type of challenges which may occur. In order to provide a better view of the landscape of IT Project challenges, data mining of status reports across all divisions and businesses of the parent company was conducted. The parent organization involves eight significant multi-national businesses with independent and non-overlapping products, IT departments with their own CIO, and a distinct portfolio of IT projects. The company at large represents an employee population of over 100,000, and the IT portfolio contains over 500 active projects at any given time.

Projects across the entire corporation are registered into a global project portfolio tracking system, with bi-weekly status updates. All projects identify certain descriptive information such as status (active, closed, etc.), project manager, key milestone dates and current challenges. Only challenges which jeopardize the success of the project are reported, and many challenges remain on the report for multiple reporting cycles until resolved. In this study, all challenges were extracted for a single reporting period, generating a point in time cross-sectional view.

In the current examination of a single month, over 460 unique project challenges were reported. The reports do not indicate if the challenges were anticipated or if a mitigation strategy had been established a priori. Organizing and categorizing is needed to extract some meaning and insight from such a lengthy list. As noted in section 2.3.1, issues and challenges are a subset of risks. The IS literature provides several schemes to classify types of risk.

Several scholars have advanced qualitative view of project risk that focuses on availability and quality of information (Daft and Lengel 1986; DeMeyer et al. 2002; Kydd 1989; Pender 2001; Pich et al. 2002; Regev et al. 2006; Yeo 1995). The Uncertainty / Equivocality Framework (Daft and Lengel, 1986) provides a two dimensional classification. In this conceptualization, uncertainty is the absence or incompleteness of information available to decision makers at a point in time. Equivocality, on the other hand, relates to the ambiguity of the information that does exist. This ambiguity is attributable to the multiple and conflicting interpretations that are possible. Decision makers often do not pursue the additional information necessary to reduce uncertainty and equivocality due to perceived limitations of time or means. A similar conceptualization by Pich et al. (2002) prescribed strategies for dealing with unknown-unknowns based on the events emerging from ambiguity (where events or causality is unknown) or complexity (where there are too many variables interacting to evaluate the situation). An alternate classification proposed by De Meyer et al. (2002) identifies a four stage continuum starting with *variation*, progressing to *foreseen uncertainty*, *unforeseen uncertainty*, and *chaos*. These schemes identify the state of a risk at the time of project initiation and prior to a risk being actualized in the form of an issue or event. Once visible the risk is no longer unforeseen or uncertain, making these classifications transient. They furthermore require understanding a project team's knowledge prior to the issue becoming real, a detail that is not available from the records examined in this situation. As a result, these qualitative classifications have limited value for this study. A key insight from this form of classification is that risk fundamentally involves the availability and transfer of knowledge and information (Regev et al. 2006). According to these studies insufficient knowledge leads to uncertainty, and poor quality information leads to ambiguity, both of which are dimensions of risk.

A second approach to classifying risk draws upon the "risk factor" literature. These studies have identified factors commonly associated with project failure (Kappelman et al. 2007; Tesch et al. 2007). Some of the factors identified in the classifications are not really a source of deviations, but correlated with increased occurrence of deviations. Two examples that are very popular in this literature are *project size* and *management support*. Neither of these common risk factors is a source of project deviations. Project size is correlated with deviations because as the size of the project grows more of the underlying causes of deviations are present and therefore an increased number of such risks are realized. Similarly, management support is not a cause of deviations, but may be associated with a project having insufficient access to resources necessary to resolve emergent issues. While both may be correlated with increased project risk, neither is a source of deviations or issues.

An alternate approach involves identifying the source or cause of risk either systemically or organizationally. Spencer and Hine (2005) identified systemic categories associated with project activity clusters: project inception, execution, technical issues, political issues, time and space, and finally event sequence. These categories are somewhat unsatisfying as they appear to overlap. For example, political issues may also be inception or execution issues. Nidumolu (1996) proposed a two dimensional classification of risk involving an organizational layer (individual, group or management) and the method of communication (oral, written, planned meetings, unplanned meetings). Lyytinen et al. (1996 and 1998) defined a socio-technical framework for software risk management across three layers. The first layer is actors vs. task, the second layer is structure vs. technology and the third is a trinary dimension of management environment, project environment or system environment. Additional frameworks by Barki et al. (2001), Boehm (1989), Benaroch (2002) and many others have cataloged over 228 risk factors (Sherer and Alter 2004).

In a review of IT project risk literature Alter and Sherer (2004) proposed the Work Systems Framework as a broad model that subsumes the *source categorization* and *factor* frameworks yet avoids limitations of the *type* frameworks. This framework provides an outline based on nine elements of work system. Four elements are the domain of the core project team: work practices, participants, information and technologies. Others identify elements external to the core-team: customers, products and services, environment, strategies, and infrastructure.

As depicted in Figure A-1, half of the challenges in this sample originated from within the core work-system (Work Practices, Participants, Information and Technologies). Teams treat these "intra-project" deviations as "operational" and expect

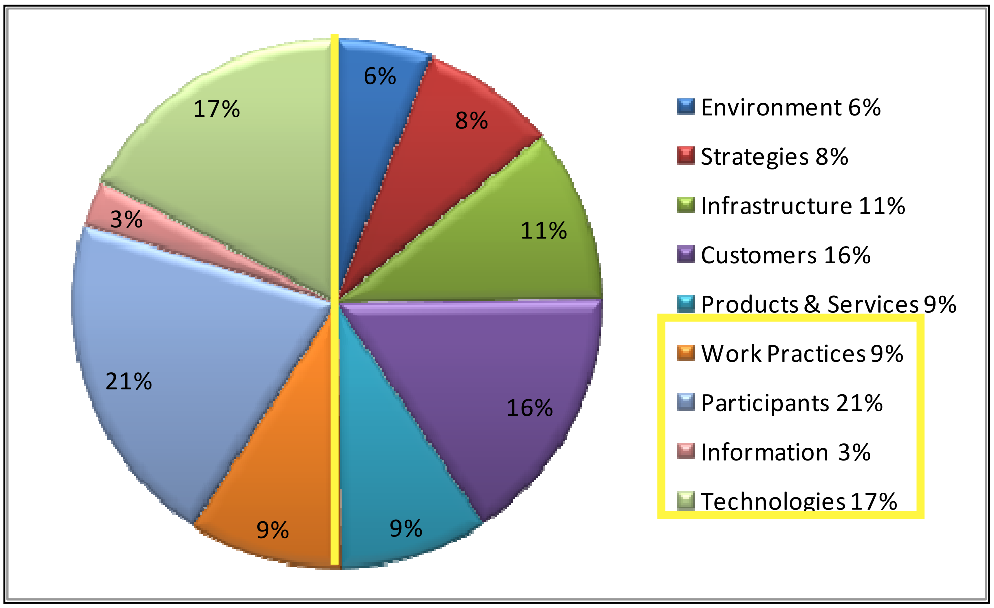


Figure A-1 Sampling of IT project challenges

to address them (Hallgren and Wilson 2007). The other half of challenges originate outside the core-team's domain of operations. These challenges cross the boundary that separates the core-team from the extended team, parallel project teams, and the larger organization. These disruptions represent external interruptions imposed on the decision making and task performing processes of the core team. These challenges often require information and assistance from others outside the team. The idea that challenges, uncertainty and change are different when they are internal versus external has been reported by scholars in the related areas of IT investment (Wu and Ong 2008) and construction projects (Sun and Meng 2009).

A.3 Positivist and Interpretivist Assessment

A.3.1 Assessment of Propositions

Assessing propositions follows a positivist perspective (Pare 2004) to determine if the proposed theory of project challenges is supported by project teams in practice.

P1: Requirements fluctuation decreases IT project performance (supported: Projects A,B,C and D)

Each project faced requirements changes that put project success at risk. Both agile projects (A & B) experienced late stage requirements changes exposed during acceptance testing. The teams reached the point where they thought they were done, only to learn during final acceptance testing that important capabilities were missing. Incremental development and periodic prototype driven feedback sessions provided insufficient mitigation. The plan-driven projects similarly faced requirements changes that jeopardized project success. Project C's standardization strategy intended to mitigate system changes was not able to deflect locally unique constraint driven requirements in the form of local character sets and statutory guidelines. Ultimately the project was required to make system changes that accommodated these requirements. Delays in addressing these requirements had a direct impact on system quality, triggering transaction backlogs and work-around actions that bypassed the system once it was activated in production. Project D experienced what may be the most significant performance impact in that training was a failure and data integrity was a mess. Both these issues had a significant impact on business operations and were at least partially due to the project not including the shortcut use case in testing and training. Experiences reported in these four cases support the proposition that requirement changes decrease IT project performance.

P2: Environmental fluctuation decreases IT project performance (supported: Projects A,B,C and D)

Each project faced unexpected disruptions from outside the core team that put project success at risk. Project A and Project D experienced staffing changes imposed by management; Project A and Project C experienced infrastructure instability; Project B experienced technical problems with software libraries provided by external teams; Project C experienced third party data quality problems; Project D experienced an intervening project of high priority. In each situation the disruption was unexpected and jeopardized project objectives. In each situation the team responded in a manner that protected primary objectives, at the expense of secondary objectives such as cost, time, and functionality. This study does not attempt to second guess the possibility that more optimal courses of action may have been possible. Rather, the simple observation is that unexpected challenges compromised certain aspects of each project's outcome.

P3: External Documentation increases IT project performance (supported: Projects A,B,C, and D)

Assessment of this proposition requires confirmation of the assumption that IT projects do generate external documentation artifacts. An interpretivist assessment of the value of these artifacts is provided in the words of project participants.

The plan-driven philosophy emphasizes formal documentation starting at the earliest stages of the project, and Projects C & D did not disappoint. Throughout the interviews, participants described meeting notes, requirements lists and specifications, business flow diagrams, functional requirements documents, system description documents, conceptual and detail design documents, technical specifications and system interface documents, contractual statements of work, test plans, training documents, quick reference guides, issue logs, risk assessments and business continuity plans stored on a global website. The formality of these documents included review and signoff by management and business/user representatives to assure all participants engaged with the artifacts during the project. Project members believe these artifacts to be more than a trophy, but valuable contributors for interactions both outside the team and problem solving within. Project C's project manager reported that documentation "*continues to help me, it really does because we've got the flow chart, and with our* [business evolving] *we've come into a whole different process/mindset/levels of controls, and actually having those to say this is the way our global model is set up*, *and we can point to it and say where does it need to differ, where our financial controls need to be, and it gives us the opportunity to look at our security and see how our roles and responsibilities within the application are set up. So all of that documentation that we have, that's standard, we can make general discussions and changes that impact the* *whole global model so that everyone's aware that when we make one change it's not just for one country, it's for every country.* {PM}"

While the agile philosophy created an expectation that documentation would be sparse, both Project A & B generated an impressive quantity and variety of documentation. Both projects chose to generate requirements specifications, supplemented by user stories, business flow diagrams, wireframes (a visual user-interface prototype), architecture documents, field specifications, URM diagrams, meeting minutes, issue logs, component technical documents, design decisions, deployment instructions, test plans, and training material centralized on wiki pages. A characteristic behavior of these agile projects is the manner in which the artifacts are used. Two purposes were identified by participants. First, documentation artifacts were used within the core project team to facilitate tasks and decisions: "*The role of documentation was to provide a means of communication between requirements gatherers, specifiers and implementers.* {PM}" Second, the documentation was intended to be used after the project was complete: "*But really, a lot of the documentation around the process and workflow, etc. it was done primarily to be used after as training material.* {ITM}"

The agile project management bias reported by Project B describes a tactical view of interaction generating a limited view of user processes that did not fully appreciation the big-picture. It starts with "*user stories* [that] *were quite brief, just a couple of sentences of what the feature was*. {PM}" These are interpreted in the context of a short duration Scrum sprint where "*the team, they only see a chunk of it for each* *sprint, it's not very strong at giving them the big picture.* {PM}" The sprints culminate in a developer orchestrated demonstration that often reinforces a myopic view of the business process. "*We had live demos, and at that point they were not providing the right feedback because it was a guided demo. The guided demo was very different from hands-on* [use]*…It wasn't until it was really in their hands that we discovered a lot of things.* {ITM}" Terse user stories conspired with short development sprints and IT run demonstrations to sustain a gap in information flow.

Many agile practitioners take pride in continuous improvement. Following the delivery of Project B, the retrospective improvement efforts of this team provide interpretivist support for Proposition 3. The team started with conventional lightweight agile artifacts and conversations. Initially "*the documentation process was extremely ad hoc, basically code reviews where the team shared knowledge amongst the developers informally* {D}" The post-project retrospective led this team to conclude ideas captured in simple user stories "*were not sufficient. So we'll start working closer with the users… We'll start making those user stories really quite detailed with very explicit requirements in them.* {PM}" In the follow-on enhancement projects, user stories are being expanded to business stories that give an overview of the product and a big picture view of the user process, including a better understanding of the scientific methods the lab is working with. "*The components that we're moving to are much more complex and there is much more* [business process knowledge] *involved. …We'll put together a set of wiki pages that provide links for further reading with a good summary and background, and it has pictures of file formats, and plate layouts, and things like* *that to help with understanding…we'll use wiki both in high level and big picture stuff, but also have some pages that really drill down to the nitty-gritty, and have lots of sample data files available for those developers to look at and to really explain things in more detail than what we would in our user stories.* {PM}" Interpretation of participant feedback supports documentation as a value-adding aid to project outcomes.

P4: External documentation attenuates negative impact of environmental fluctuation (Equivocal)

P5: External documentation amplifies negative impact of requirements fluctuation (Equivocal)

An assumption implicit for both proposition 4 and 5 motivated by existing literature involves an expectation of high documentation output by plan-driven teams and low documentation output for agile teams. While the labels given to the artifacts differ, the interviews do not support significant distinction for the quantity or variety of documentation. Therefore a deductive conclusion for propositions 4 and 5 is in doubt. However, opinions expressed by project members do allow some interpretivist progress.

Plan-Driven teams in this investigation employed documentation broadly as an active component of most tasks. Formal requirements documentation was revised and re-circulated for review and signature when scope and requirements changes were encountered. Formal statements of work were created to communicate technical design changes or bug fixes to external technology teams. These projects leveraged the early process flow charts to aid on-boarding of new members. Interface documents were similarly reused as team works with multiple external technology teams.

A very different opinion of the value and purpose of external documentation was expressed by the agile projects. When asked what role the documentation played when dealing with technical and staff challenges the response from Project A was: "*No role whatsoever*. {PM}" A follow-up question specifically asking about certain tools and artifacts previously identified garnered a similar response: "*No, they were unable to help entirely. The tools were not aware of the organizational environment or could not react to changing membership, they couldn't help on the organizational front and the tools were absolutely not linked to the technological environment in which the projects artifacts were being run. They could not help at all in managing the changing technical environment.* {PM}" Interestingly, these artifacts were integral participants when dealing with anticipated changes: "[the tool, notes and logs] *were more instrumental in that anticipated change. In fact, it was part of the mitigation strategy for that, the way documentation was used.* {PM}"

The narratives support the idea that while practitioners in both camps generate and collect documentation that may arguably be similar in variety and quantity, there is a difference in how and when those artifacts are employed. Agile teams appear to view the documents to be important for tasks and challenges that exist within the core project team. When challenges originate outside the agile project team documentation played "*no role whatsoever*."

The bias appears to extend to other forms of communication as well. Agileists encourage minimal documentation while emphasizing informal face-to-face communication. However, there does not appear to be any rigor in the manner in which interactions occur outside the core team. When informal networking was effective, it came as a surprise to members: "*That was pure luck. The individual who ended up helping us solve the problem happened to be a developer who sits next to me.* {PM}"; "*We accidentally ended up talking to people at the other sites and realized we were working on a partial set of requirements.* {D}" The cases in this study suggest that agile teams use both documentation and social interaction proactively to the focal task of the sprint underway, but these communication methods serve only a minor role when the team reacts to unexpected challenges. Agile teams emphasize progress on the task they own in a way that encourages experimentation and innovative solutions generated within the team, while at the same time not proactively employing the same assets to tasks that are shared with other teams or the organization as a whole.

A.3.2 Next Steps

It is apparent that both plan-driven and agile projects generate written documentation, and that these artifacts convey team knowledge across time and space. The formal requirements specifications and designs generated by plan-driven projects were updated during the course of their projects and serve as an engaged cognitive participant in work tasks and also as a means of communicating with users, management and other stakeholders during workshops and later in review and approval exercises. These documents were actively involved in re-planning exercises and communication with stakeholders when schedules and plans were changed. Agile teams also used their written documentation in the form of user stories, wireframe mockups and test scripts to facilitate cognitive processes within the core team, but not as a communication aid with users and stakeholders, where such artifacts were viewed as having little or no value.

Explaining these differences in behavior requires a more nuanced theoretical perspective than that offered in the original five propositions. The assessment of propositions that guided this investigation suggest that project management biases engage unexpected challenges not with the presence or absence of plans and documentation, but the manner in which the team is influenced by plans and uses documentation as a tool. Additional mechanisms beyond the heuristic of Anchoring-and-Adjustment at the group cognitive level must exist.

Section A.3.1 describes how this case study investigation has been pursued using a positivist perspective to draw deductive conclusions to theoretical propositions from field experiences. This is supplemented by interpretivist insights from recorded opinions of project participants. Support for certain propositions is clear, while at the same time suggesting the mechanisms at work are more nuanced than represented in the initial propositions. In order to achieve a more complete explanation of the mechanisms at work when IT projects encounter unexpected challenges, a different perspective is needed. Section 2.4 assesses the events, structures, and themes from a Critical Realist perspective in an effort to identify the mechanisms which may explain observed behaviors and suggest theory for unexpected project challenges.

Appendix B

Survey Items, Descriptive Statistics and Summary Statistics

Table B-1 Survey items and descriptive statistics

Performance: (Chandrasekaran & Mishra, 2012) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
PERF1	This project completed in estimated **time** relative to its goals. (C&M 2012)	4.134	1.480	0.059	0.926	<0.001	CR=0.96 ave=0.67
PERF2	This project completed within estimated **cost** relative to its goals. (C&M 2012)	4.186	1.434	0.070	0.881	<0.001	
PERF3	This project completed with **all functionality** relative to its goals. (C&M 2012)	*4.612*	*1.196*	*Removed, weak indicator reliability (p-value = 0.082)*			
PERF4	This project completed with **promised quality.** (C&M 2012)	4.738	1.052	0.095	0.605	<0.001	
User Satisfaction: (Seddon & Yip, 1992) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
USI1	**Users** feel the new system meets their needs. *(S&Y 1992)*	4.738	0.963	0.057	0.919	<0.001	CR=0.98 ave=0.87
USI2	**Users** feel the system is highly effective. *(S&Y 1992)*	4.623	1.025	0.060	0.935	<0.001	
USI3	Overall, **users** are very satisfied with the system. *(S&Y 1992)*	4.612	1.001	0.058	0.944	<0.001	
Sponsor Satisfaction: (Ferreira & Cohen, 2008) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
SS1	**Sponsors** were satisfied with the project development process. *(F&C 2008)*	4.735	0.980	0.067	0.895	<0.001	CR=0.98 ave=0.84
SS2	**Sponsors** feel the project met their requirements. *(F&C 2008)*	4.937	0.936	0.061	0.922	<0.001	
SS3	On the whole, **sponsors** were satisfied with the project outcome. *(F&C 2008)*	4.925	0.961	0.062	0.934	<0.001	
Member Satisfaction: (Aladwandi, 2002) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
MS1	Generally speaking, **members** of the project were very satisfied with their work. *(A 2002)*	4.813	0.832	0.083	0.595	<0.001	CR=0.96 ave=0.55
MS2	**Team members** were generally satisfied with their role on this project. *(A 2002)*	4.701	0.803	0.071	0.639	<0.001	
MS3	**Team members** frequently thought of quitting the project. *(A 2002) (rev. coded)*	4.377	1.302	0.053	0.938	<0.001	

CR = Composite Reliability; ave = Average Variance Extracted.

Staff Fluctuations: (Carbonell & Rodriguez, 2006; Gopal & Gosain, 2010) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
SF1	All project team members worked full time on this project, with no other work assignments. *(C&R 2006) (rev. coded)*	3.858	1.615	*Removed, weak indicator reliability (p-value = 0.344)*			CR=0.94 ave=0.60
SF2	Member participation level continually changed due to non-project activities. *(new)*	3.705	1.340	0.178	0.502	0.002	
SF3	Turnover of key project team members was common. *(G&G 2010)*	2.866	1.378	0.085	0.870	<0.001	
SF4	Overall, member changes were highly significant. *(new)*	2.922	1.368	0.061	0.890	<0.001	
Technology Fluctuations: 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
TF1	Technology problems occurred frequently during this project. *(new)*	3.287	1.350	0.058	0.826	<0.001	CR=0.98 ave=0.71
TF2	Designs changed frequently to accommodate technology problems. *(new)*	3.213	1.331	0.064	0.847	<0.001	
TF3	Functional capabilities were removed or deferred due to technology problems. *(new)*	3.037	1.351	0.071	0.817	<0.001	
TF4	Overall, technology problems were highly significant. *(new)*	3.112	1.446	0.059	0.884	<0.001	
Requirements Fluctuations: (Wallace, Keil & Rai, 2004a) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	
RF1	System requirements were not adequately identified. *(WKR 2004)*	2.970	1.390	0.082	0.745	<0.001	CR=0.97 ave=0.59
RF2	Requirements never changed during the project. *(rev. coded)*	4.328	1.340	0.100	0.602	<0.001	
RF3	System Requirements frequently needed correction. *(WKR 2004)*	3.362	1.286	0.051	0.819	<0.001	
RF4	Overall, requirements changes were highly significant. *(new)*	3.463	1.457	0.054	0.885	<0.001	
Calculated Flux Intensity Factors		Count	%				
HIF – High Intensity Flux (where SF4, TF4 or RF4 >= 5)		112	41.7%				

Locus of Flux: (Imamoglu & Gozlu, 2008) 1-6: Not At All to Very Great Extent		Mean	Std. Dev.	Std. Err.	Load	P-Value	
LOF1	To what extent was the cause of issues and challenges something the project team anticipated? *(new)*	3.190	1.280	0.236	0.637	<0.001	CR=0.89 ave=0.68
LOF2	To what extent was the cause of issues and challenges something controlled by the core project team? (I&G 2008)	3.373	1.420	0.080	0.975	<0.001	
LOF3	To what extent was the cause of issues and challenges something to do with actions or responsibilities within the core project team? *(I&G 2008)*	2.978	1.411	*Removed, weak indicator reliability (p-value = 0.714)*			
LOF4	To what extent was the cause of issues and challenges something to do with the people or circumstances outside the core project team? *(I&G 2008) (rev. coded)*	2.847	1.441	*Removed, weak indicator reliability (p-value = 0.544)*			
Agile Style: (Chandrasekaran & Mishra 2012; Shenhar & Dvir, 1996) 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value	CR=0.96 ave=0.64
AS1	Key design decisions were communicated primarily in face-to-face discussion. *(new)*	4.336	1.321	*Removed, weak indicator reliability (p-value = 0.166)*			
AS2	Interactions with people outside the core team took place mostly during formal workshops & meetings. *(SD 1996) (rev. coded)*	4.108	1.286	0.114	0.575	<0.001	
AS3	Meetings were held daily with the full team. *(new)*	3.175	1.480	0.135	0.692	<0.001	
AS4	Requirements were collected from users primarily during feedback from prototyping and iterative development. *(new)*	3.955	1.343	0.115	0.736	<0.001	
AS5	The project team had control over *what they were supposed to accomplish. (CM 2012)*	4.224	1.237	0.068	0.954	<0.001	
AS6	The project team was free to assign personnel to the project. *(CM 2012)*	3.478	1.436	0.126	0.849	<0.001	
AS7	The project team had full control over planning and re-planning. *(CM 2012)*	3.869	1.255	0.087	0.874	<0.001	
AS8	The project team worked independently and without supervision. *(new)*	3.537	1.380	0.083	0.866	<0.001	

<table>
<tr><td colspan="2">Plan Driven PM Style
(Chesci, Sillitti, Succi & DePanfilis 2005; Dvir, Lipovetsky, Shenhar & Tishler 1998)
1-6: Strongly Disagree to Strongly Agree</td><td>Mean</td><td>Std. Dev.</td><td>Std. Err.</td><td>Load</td><td>P-Value</td><td rowspan="7">CR=0.97
ave=0.55</td></tr>
<tr><td>PDS1</td><td>This project pursued specific and complete documentation at every phase. (CSSD 2005)</td><td>4.198</td><td>1.237</td><td>0.075</td><td>0.767</td><td><0.001</td></tr>
<tr><td>PDS2</td><td>This project created formal specification documents (DLST 1998)</td><td>4.526</td><td>1.172</td><td>0.071</td><td>0.831</td><td><0.001</td></tr>
<tr><td>PDS3</td><td>Careful planning and documentation was performed before development. (CSSD 2005)</td><td>4.265</td><td>1.307</td><td>0.065</td><td>0.865</td><td><0.001</td></tr>
<tr><td>PDS4</td><td>This project identified a schedule with milestones. (DLST 1998)</td><td>5.075</td><td>0.879</td><td>0.091</td><td>0.601</td><td><0.001</td></tr>
<tr><td>PDS5</td><td>This project tracked budget utilization. (DLST 1998)</td><td>4.537</td><td>1.219</td><td colspan="3">Removed, weak indicator reliability
(p-value = 0.131)</td></tr>
<tr><td>PDS6</td><td>This project performed planning of activities with a Work Breakdown Structure. (DLST 1998)</td><td>4.504</td><td>1.279</td><td>0.114</td><td>0.618</td><td><0.001</td></tr>
<tr><td colspan="2">Reactive Decoupling (responsiveness & distinctiveness)
(Akgun & Lynn, 2002b; Cameron 1986; Sawyer, Guinan, Cooprider 2010)
1-6: Strongly Disagree to Strongly Agree</td><td>Mean</td><td>Std. Dev.</td><td>Std. Err.</td><td>Load</td><td>P-Value</td><td rowspan="6">CR=0.85
ave=0.54</td></tr>
<tr><td>RD1 (resp)</td><td>Communication outside the team involved specific and significant challenges was frequent. (new)</td><td>3.903</td><td>1.256</td><td colspan="3">Removed to correct convergent validity
(AVE < 0.5)</td></tr>
<tr><td>RD2 (resp)</td><td>When dealing with significant challenges, general and routine communication outside the team significantly increased compared to before the disruption. (new)</td><td>4.056</td><td>1.203</td><td>0.222</td><td>0.526</td><td>0.045</td></tr>
<tr><td>RD4 (resp)</td><td>When dealing with significant challenges, this project continued to follow its pre-defined plan. (AL 2002)</td><td>4.216</td><td>1.073</td><td>0.149</td><td>0.894</td><td><0.001</td></tr>
<tr><td>RD5 (resp)</td><td>When dealing with significant challenges, the project team was more focused on quick execution (meeting deadlines) than other considerations. (C 1986)</td><td>3.866</td><td>1.192</td><td colspan="3">Removed, weak indicator reliability
(p-value = 0.171)</td></tr>
<tr><td>RD6 (resp)</td><td>When dealing with significant challenges, the project team secured approval for all unplanned activities with management and stakeholders. (new)</td><td>4.269</td><td>1.216</td><td colspan="3">Removed to correct convergent validity
(AVE < 0.5)</td></tr>
</table>

RD3 (dist)	When dealing with significant challenges, efforts were made to buffer the team from distractions and communications outside the team. *(SGC 2010)*	4.056	1.203	0.188	0.550	0.047	
RD7 (dist)	When dealing with significant challenges, this project team decided exclusively on its own what actions to take. *(C 1986)*	3.239	1.328	0.128	0.854	<0.001	CR=0.86 ave=0.52
RD8 (dist)	When dealing with significant challenges, news of the situation was intentionally kept within the team until the appropriate time. *(SGC 2010)*	3.664	1.340	*Removed to correct convergent validity (AVE < 0.5)*			
RD9 (dist)	When dealing with significant challenges, the project applied more effort improvising new solutions than finding an existing solution from outside the team. *(C 1986)*	3.593	1.225	*Removed, weak indicator reliability (p-value = 0.160)*			
Demographic & Control Variables Scale1 [1-6: strongly disagree – strongly agree] Scale2 [1-6: not at all – very great extent]		Mean	Std. Dev.				
C1	How many members on the core team? ______	19.940	56.228				
C2	To what extent was the core team spread across many locations? *(Scale2)*	3.153	1.760				
C3	To what extent are users spread across many locations? *(Scale2)*	3.739	1.761				
C4	This project worked with technologies that were new to the organization. *(Scale1)*	4.011	1.585				
P1	Identify the Project Management methods that most accurately describe what this project used.	Count	%				
	1 = plan-driven	172	64%				
	2 = agile	59	22%				
	3 = No established PM methodology	16	6%				
	4 = Other ______	19	7%				
C5	How long has it been (in months) since the project completed? ______	9.619	15.728				

Demographic & Control Variables Scale1 [1-6: strongly disagree – strongly agree] Scale2 [1-6: not at all – very great extent]		Mean	Std. Dev.	
C6	How long did the project last (in months)? ______	15.153	20.050	
Q1	Select the percent of the project duration you have been heavily involved? [1 = 0-15%, 2 = 16-30%, 3 = 31-45%, 4 = 46-60%, 5 = 61-75%, 6 = 76-100%]	5.201	1.244	
Q2	You have a deep understanding of this project's events and processes. *(Scale 1)*	5.534	0.620	Filter/qualification question. Exclude 1-3, include 4-6.
Role	Identify your role on the project. [1= User/Customer, 2=Project Manager, 3 = IT Management, 4 = IT Contributor (Analyst/Developer/etc.), 5 = Other]	2.00	0.000	Filter/qualification question. Include only Project Managers.
Q4	Years of experience in your role.			
	1 = less than 1 year	3	1%	*3 respondents did not answer this question.*
	2 = 1-2 years	9	3%	
	3 = 3-4 years	28	10%	
	4 = 5-10 years	81	30%	
	5 = 10+ years	144	54%	
C7	What type of system implementation project is this (select only those that apply)			
	1= custom built IT system	96	36%	
	2=custom built enhancement to IT system	61	23%	
	3=custom built integration(s) between IT systems	85	32%	
	4=customized report(s) for the IT system	45	17%	
	5=customized IT system through configuration	66	25%	
	6=commercial off-the-shelf IT system	67	25%	
	7=system subject to regulatory governance	48	18%	
	8=Other	28	10%	

D1	Your Age					
	1 = under 18	0	0%	*41 respondents did not answer this question.*		
	2 = 18-24	1	<1%			
	3 = 25-34	43	18%			
	4 = 35-44	107	44%			
	5 = 45-54	76	31%			
	6 = 55+	0	0%			
D2	Select the categorization that best describes your organization.					
	1=Agriculture, farming or ranching	1	0.4%	*One respondent did not respond to this question.*		
	2=Computer or telecommunication hardware/software products or services.	92	34.3%			
	3=consumer products or services (including retail)	17	6.3%			
	4=Construction	3	1.1%			
	5=Defense	14	5.2%			
	6=Entertainment, sports and recreation	2	0.7%			
	7=Financial products or services	28	10.4%			
	8=Government or public sector	20	7.5%			
	9=Health care products or services	29	10.8%			
	10=Manufacturing or industrial	20	7.5%			
	11=Not-for-profit organization	8	3.0%			
	12=Transportation	8	3.0%			
	13=Other ____	25	9.3%			
CMV Marker Variable – previous system quality: 1-6: Strongly Disagree to Strongly Agree		Mean	Std. Dev.	Std. Err.	Load	P-Value
MK1	The old system was well understood by users. *(rev. coded)*	4.810	1.303	*Removed, weak indicator reliability (p-value = 0.066)*		
MK2	The old system provided poor quality information.	2.530	1.495	0.231	0.751	<0.001
MK3	Users thought the old system was unreliable.	2.948	1.445	0.137	0.979	<0.001

Table B-2 Latent variable correlations

	AStyle	*Duration*	FLUX	*HIF*	*Industry*	PDStyle	PERF	RDdist	RDresp
AStyle	*0.802*								
Duration	*-0.073*	*1.000*							
FLUX	*-0.299*	*0.125*	*0.912*						
HIF	*-0.063*	*0.100*	*0.617*	*1.000*					
Industry	*-0.008*	*-0.118*	*-0.038*	*-0.039*	*1.000*				
PDStyle	*0.349*	*-0.037*	*-0.319*	*-0.160*	*-0.056*	*0.744*			
PERF	*0.428*	*-0.142*	*-0.414*	*-0.336*	*-0.088*	*0.317*	*0.817*		
RDdist	*0.439*	*-0.106*	*0.010*	*0.089*	*-0.037*	*0.049*	*0.206*	*0.718*	
RDresp	*0.252*	*-0.051*	*-0.403*	*-0.244*	*-0.065*	*0.192*	*0.332*	*0.096*	*0.733*
Square root of AVE along Diagonal (single indicator variables are assigned an AVE of 1.000)									

Table B-3 Item cross loading

	AStyle	*Duration*	*Industry*	LOF	HIF	PDStyle	PERF	RF	RLC_Dist	RLC_resp	SF	TF
D2	0.271	-0.004	3.896	-0.481	0.292	0.266	0.523	0.036	0.367	0.478	0.146	0.100
HIF	-0.020	0.049	0.037	-0.027	0.493	-0.078	-0.168	0.298	0.035	-0.118	0.209	0.283
ab2r	-0.613	-0.009	-0.045	-0.168	-0.014	-0.319	-0.217	0.128	-0.220	-0.075	-0.049	0.099
ab3	0.855	-0.129	-0.110	0.263	0.139	0.187	0.354	0.086	0.392	-0.033	0.136	0.004
ab4	0.791	0.010	0.020	0.327	0.091	0.262	0.228	-0.105	0.336	-0.024	0.243	0.138
ab5	0.942	-0.071	0.066	0.350	-0.311	0.468	0.529	-0.583	0.297	0.390	-0.208	-0.358
ab6	0.858	-0.030	0.075	0.326	0.132	0.107	0.284	-0.024	0.394	0.112	-0.044	0.094
ab7	0.813	0.002	0.169	0.184	-0.079	0.316	0.361	-0.275	0.347	0.268	-0.206	-0.129
ab8	0.767	-0.162	0.127	0.127	0.021	0.217	0.303	-0.207	0.511	0.272	-0.108	-0.063
c6	-1.425	19.732	-0.018	-1.023	1.964	-0.680	-2.815	2.905	-2.547	-0.955	-0.163	3.689
loc1	0.252	0.038	-0.059	0.813	0.119	0.234	0.079	0.076	0.140	0.034	0.100	0.030
loc2	0.431	-0.097	-0.184	1.382	-0.127	0.155	0.248	-0.127	0.242	0.029	0.040	-0.090
pdb1	0.229	-0.014	0.008	0.119	-0.107	0.908	0.234	-0.228	0.033	0.270	-0.115	-0.175
pdb2	0.255	0.000	-0.006	0.111	-0.158	0.952	0.304	-0.309	-0.007	0.220	-0.250	-0.226
pdb3	0.449	-0.144	0.100	0.158	-0.255	1.124	0.402	-0.419	0.046	0.211	-0.185	-0.280
pdb4	0.175	0.016	0.097	0.032	-0.048	0.538	0.138	-0.184	-0.032	0.066	-0.162	-0.122
pdb6	0.362	0.046	0.111	0.166	-0.071	0.848	0.308	-0.192	0.095	0.056	-0.109	-0.199
pp1	0.601	-0.214	0.177	0.280	-0.477	0.433	1.369	-0.527	0.264	0.459	-0.401	-0.463
pp2	0.499	-0.157	0.196	0.095	-0.370	0.346	1.240	-0.478	0.245	0.411	-0.313	-0.295
pp4	0.306	-0.091	0.062	0.173	-0.278	0.329	0.669	-0.305	0.202	0.226	-0.222	-0.315
rf1	-0.301	0.048	0.035	-0.120	0.585	-0.501	-0.389	1.087	0.037	-0.399	0.441	0.829
rf2r	-0.319	0.223	-0.152	-0.019	0.259	-0.256	-0.348	0.708	-0.324	-0.477	0.090	0.255
rf3	-0.183	0.119	0.053	-0.069	0.609	-0.290	-0.387	1.073	0.112	-0.314	0.375	0.709
rf4	-0.349	0.263	0.025	-0.046	0.940	-0.277	-0.540	1.280	0.024	-0.500	0.516	0.883
rd2	-0.020	0.129	0.007	0.015	0.308	0.008	-0.152	0.373	0.140	-0.603	0.273	0.407
rd3	0.262	0.056	-0.031	0.099	0.135	0.108	0.113	0.058	0.425	-0.112	0.008	0.050
rd4	0.264	-0.003	0.155	0.036	-0.161	0.235	0.346	-0.312	0.230	0.970	-0.156	-0.166
rd7	0.500	-0.211	0.149	0.206	0.040	0.006	0.250	-0.025	1.242	0.247	0.095	0.111
sf2	-0.104	-0.120	0.178	-0.009	0.197	-0.039	-0.146	0.140	-0.005	-0.151	0.688	0.174
sf3	-0.064	0.007	-0.007	0.141	0.364	-0.150	-0.336	0.419	0.099	-0.232	1.197	0.542
sf4	-0.055	0.023	0.029	-0.013	0.687	-0.318	-0.368	0.505	0.089	-0.304	1.209	0.575
tf1	-0.098	0.299	0.000	-0.058	0.581	-0.196	-0.332	0.705	0.097	-0.328	0.455	1.126
tf2	-0.075	0.169	0.024	-0.004	0.592	-0.303	-0.308	0.816	0.182	-0.367	0.471	1.147
tf3	-0.218	0.237	-0.015	-0.126	0.589	-0.296	-0.400	0.726	0.035	-0.271	0.547	1.084
tf4	-0.141	0.172	0.099	-0.042	0.866	-0.281	-0.437	0.857	0.116	-0.326	0.561	1.265

Table B-4 Item correlations

	pp1	*pp2*	*pp4*	*usi1*	*usi2*	*usi3*	*mk1r*
pp1	1.000						
pp2	0.680	1.000					
pp4	0.475	0.385	1.000				
usi1	0.340	0.225	0.646	1.000			
usi2	0.339	0.285	0.595	0.783	1.000		
usi3	0.366	0.249	0.636	0.811	0.820	1.000	
mk1r	-0.097	-0.091	-0.078	-0.124	-0.109	-0.075	1.000
mk2	0.018	0.092	0.095	0.078	0.111	0.094	-0.094
mk3	0.172	0.167	0.130	0.134	0.203	0.211	0.013
ss1	0.438	0.443	0.510	0.454	0.471	0.445	-0.098
ss2	0.379	0.335	0.550	0.588	0.545	0.585	-0.153
ss3	0.407	0.385	0.603	0.602	0.576	0.608	-0.123
ms1	0.382	0.306	0.423	0.374	0.431	0.408	-0.040
ms2	0.387	0.358	0.386	0.315	0.372	0.368	-0.039
ms3r	0.279	0.225	0.288	0.264	0.300	0.325	-0.042
pdb1	0.178	0.112	0.212	0.204	0.286	0.280	0.044
pdb2	0.233	0.178	0.273	0.235	0.303	0.302	-0.051
pdb3	0.283	0.249	0.233	0.204	0.276	0.273	-0.047
pdb4	0.110	0.114	0.228	0.134	0.193	0.165	-0.061
pdb6	0.207	0.181	0.243	0.138	0.262	0.194	0.019
ab2r	-0.169	-0.111	-0.143	-0.074	-0.116	-0.111	0.070
ab3	0.270	0.156	0.135	0.085	0.113	0.117	-0.029
ab4	0.139	0.146	0.156	0.098	0.170	0.118	0.022
ab5	0.421	0.346	0.261	0.257	0.350	0.306	-0.017
ab6	0.162	0.178	0.167	0.153	0.163	0.132	0.013
ab7	0.257	0.251	0.204	0.220	0.290	0.198	-0.079
ab8	0.188	0.208	0.143	0.151	0.231	0.230	-0.053
sf2	-0.146	-0.071	-0.004	0.033	0.001	0.001	0.041
sf3	-0.208	-0.201	-0.221	-0.137	-0.153	-0.144	0.085
sf4	-0.265	-0.201	-0.196	-0.084	-0.109	-0.077	0.084
tf1	-0.239	-0.177	-0.200	-0.227	-0.235	-0.294	0.141
tf2	-0.226	-0.164	-0.190	-0.219	-0.246	-0.295	0.091
tf3	-0.298	-0.152	-0.328	-0.375	-0.333	-0.382	0.094
tf4	-0.290	-0.200	-0.286	-0.264	-0.234	-0.272	0.084
rf2r	-0.228	-0.237	-0.175	-0.131	-0.144	-0.159	0.041
rf3	-0.266	-0.266	-0.215	-0.174	-0.265	-0.216	0.102
rf4	-0.336	-0.314	-0.270	-0.258	-0.264	-0.274	0.092
rf1	-0.260	-0.219	-0.223	-0.297	-0.281	-0.307	0.098
rd2	-0.137	-0.078	-0.092	-0.010	0.048	-0.004	0.103
rd3	0.111	0.006	0.182	0.150	0.235	0.219	0.068
rd4	0.290	0.293	0.203	0.185	0.200	0.270	-0.043
rd7	0.151	0.183	0.139	0.087	0.135	0.112	-0.087
loc1	0.091	-0.015	0.087	0.047	0.166	0.087	0.091
loc2	0.193	0.082	0.166	0.107	0.153	0.118	0.067

	mk2	*mk3*	*ss1*	*ss2*	*ss3*	*ms1*	*ms2*
pp1							
pp2							
pp4							
usi1							
usi2							
usi3							
mk1r							
mk2	1.000						
mk3	0.565	1.000					
ss1	0.072	0.139	1.000				
ss2	0.081	0.146	0.713	1.000			
ss3	0.131	0.192	0.743	0.828	1.000		
ms1	0.086	0.117	0.486	0.500	0.526	1.000	
ms2	0.108	0.043	0.480	0.443	0.500	0.702	1.000
ms3r	-0.022	-0.016	0.325	0.293	0.274	0.314	0.369
pdb1	0.074	0.130	0.223	0.189	0.220	0.265	0.305
pdb2	0.081	0.092	0.331	0.310	0.328	0.282	0.327
pdb3	-0.008	0.068	0.362	0.289	0.257	0.294	0.375
pdb4	0.070	0.023	0.306	0.320	0.264	0.275	0.286
pdb6	0.043	-0.008	0.307	0.336	0.244	0.300	0.282
ab2r	-0.252	-0.126	-0.142	-0.164	-0.146	-0.211	-0.216
ab3	0.167	0.159	0.102	0.051	0.120	0.136	0.114
ab4	0.187	0.190	0.190	0.081	0.079	0.096	0.154
ab5	0.164	0.197	0.408	0.307	0.326	0.357	0.403
ab6	0.104	0.040	0.173	0.131	0.159	0.128	0.218
ab7	0.064	0.012	0.261	0.175	0.191	0.271	0.296
ab8	0.000	0.059	0.156	0.142	0.183	0.153	0.193
sf2	-0.037	0.025	-0.063	-0.027	-0.046	-0.107	-0.166
sf3	0.050	0.030	-0.229	-0.236	-0.214	-0.247	-0.250
sf4	0.058	0.053	-0.264	-0.250	-0.181	-0.263	-0.257
tf1	-0.051	-0.108	-0.163	-0.184	-0.229	-0.109	-0.131
tf2	0.015	-0.029	-0.250	-0.275	-0.275	-0.221	-0.238
tf3	-0.016	-0.047	-0.338	-0.386	-0.356	-0.277	-0.235
tf4	-0.033	-0.031	-0.225	-0.282	-0.282	-0.225	-0.236
rf2r	-0.050	-0.185	-0.156	-0.133	-0.141	-0.073	-0.152
rf3	0.041	-0.038	-0.236	-0.267	-0.272	-0.196	-0.196
rf4	0.019	-0.150	-0.318	-0.267	-0.302	-0.213	-0.224
rf1	0.043	-0.022	-0.253	-0.321	-0.282	-0.225	-0.317
rd2	-0.005	0.037	-0.089	-0.023	-0.045	-0.023	-0.099
rd3	0.104	0.086	0.113	0.089	0.095	0.153	0.119
rd4	0.004	0.019	0.201	0.241	0.241	0.117	0.227
rd7	0.048	0.120	0.032	0.024	0.096	0.098	0.151
loc1	-0.039	-0.022	0.040	0.076	0.073	0.114	0.103
loc2	0.020	-0.064	0.082	0.046	0.048	0.100	0.098

	ms3r	*pdb1*	*pdb2*	*pdb3*	*pdb4*	*pdb6*	*ab2r*
pp1							
pp2							
pp4							
usi1							
usi2							
usi3							
mk1r							
mk2							
mk3							
ss1							
ss2							
ss3							
ms1							
ms2							
ms3r	1.000						
pdb1	0.058	1.000					
pdb2	0.154	0.672	1.000				
pdb3	0.172	0.528	0.618	1.000			
pdb4	0.211	0.331	0.401	0.449	1.000		
pdb6	0.169	0.261	0.322	0.406	0.496	1.000	
ab2r	0.004	-0.168	-0.206	-0.159	-0.138	-0.251	1.000
ab3	-0.069	0.065	0.076	0.106	0.024	0.151	-0.250
ab4	-0.033	0.026	0.044	0.231	0.127	0.236	-0.352
ab5	0.296	0.238	0.244	0.380	0.212	0.274	-0.359
ab6	0.084	-0.032	0.033	0.064	0.114	0.119	-0.104
ab7	0.175	0.140	0.169	0.261	0.179	0.169	-0.155
ab8	0.122	0.133	0.123	0.205	0.047	0.020	-0.085
sf2	-0.088	0.017	-0.027	-0.030	0.038	-0.055	0.014
sf3	-0.427	-0.048	-0.188	-0.067	-0.162	-0.012	-0.063
sf4	-0.362	-0.141	-0.222	-0.196	-0.219	-0.129	-0.020
tf1	-0.358	-0.128	-0.141	-0.128	-0.078	-0.060	0.018
tf2	-0.364	-0.167	-0.182	-0.203	-0.132	-0.149	0.013
tf3	-0.342	-0.103	-0.161	-0.222	-0.135	-0.160	0.065
tf4	-0.309	-0.090	-0.167	-0.174	-0.124	-0.150	0.145
rf2r	-0.060	-0.127	-0.144	-0.234	-0.024	-0.084	0.175
rf3	-0.352	-0.125	-0.186	-0.215	-0.216	-0.127	0.033
rf4	-0.378	-0.105	-0.189	-0.206	-0.153	-0.053	0.041
rf1	-0.366	-0.225	-0.287	-0.361	-0.210	-0.213	0.120
rd2	-0.176	-0.010	-0.013	-0.029	0.056	0.062	-0.093
rd3	0.079	0.002	0.056	0.083	0.073	0.133	-0.130
rd4	0.180	0.247	0.210	0.173	0.114	0.081	-0.113
rd7	-0.033	0.028	-0.028	0.007	-0.067	0.030	-0.136
loc1	0.026	0.120	0.083	0.173	0.120	0.168	-0.113
loc2	0.049	0.077	0.086	0.090	0.008	0.102	-0.119

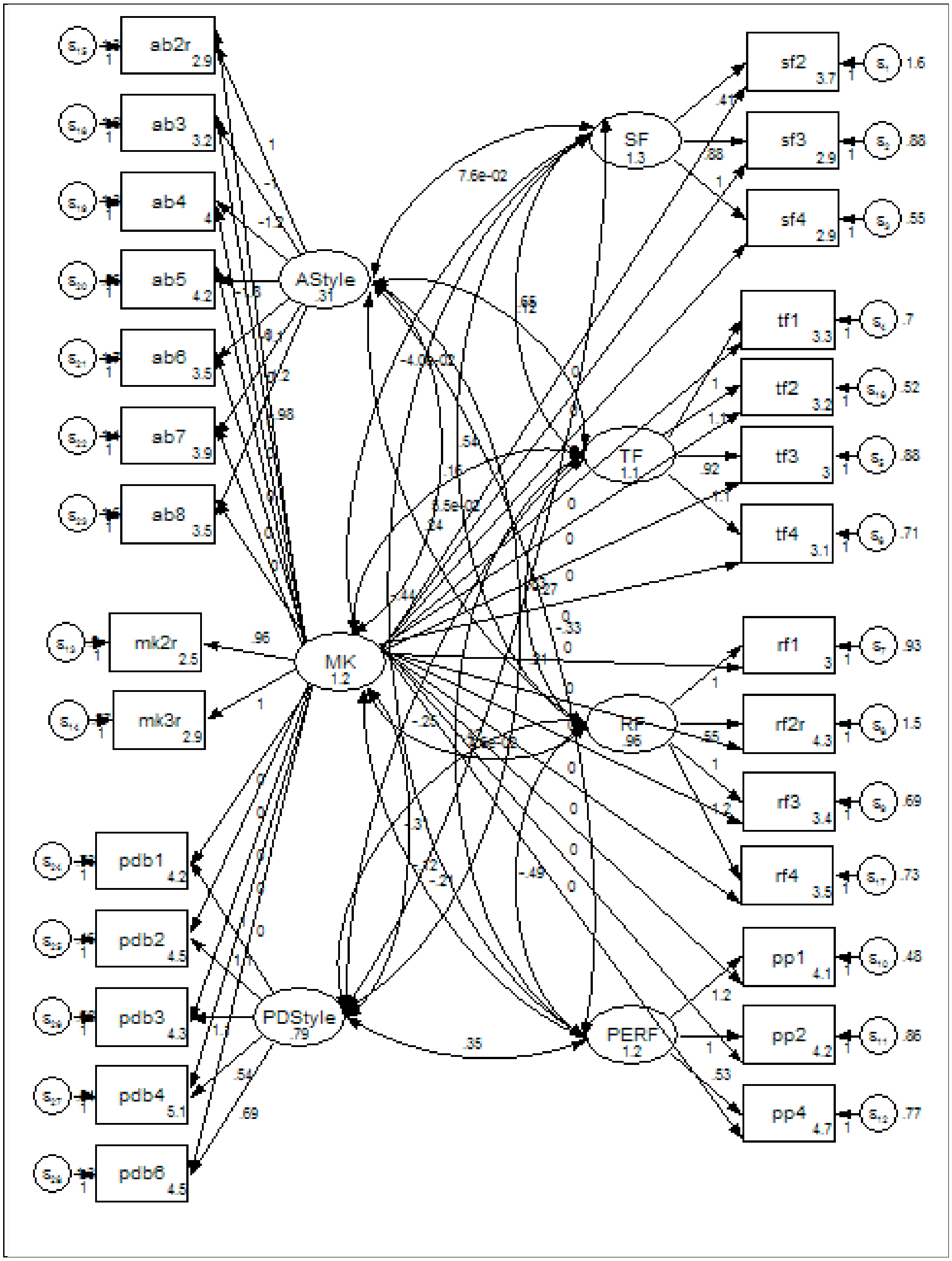
	ab3	*ab4*	*ab5*	*ab6*	*ab7*	*ab8*	*sf2*
pp1							
pp2							
pp4							
usi1							
usi2							
usi3							
mk1r							
mk2							
mk3							
ss1							
ss2							
ss3							
ms1							
ms2							
ms3r							
pdb1							
pdb2							
pdb3							
pdb4							
pdb6							
ab2r							
ab3	1.000						
ab4	0.336	1.000					
ab5	0.283	0.394	1.000				
ab6	0.221	0.256	0.287	1.000			
ab7	0.101	0.201	0.431	0.411	1.000		
ab8	0.176	0.191	0.245	0.295	0.391	1.000	
sf2	-0.093	0.078	-0.105	-0.041	-0.036	-0.040	1.000
sf3	0.056	0.185	-0.105	-0.060	-0.125	-0.100	0.281
sf4	0.164	0.141	-0.180	0.015	-0.183	-0.039	0.302
tf1	-0.001	0.098	-0.200	0.093	-0.077	-0.079	0.070
tf2	0.057	0.133	-0.232	0.096	-0.100	-0.038	0.098
tf3	-0.058	0.083	-0.339	0.018	-0.097	-0.057	0.138
tf4	0.010	0.045	-0.215	0.025	-0.077	0.007	0.127
rf2r	-0.012	-0.071	-0.273	-0.076	-0.166	-0.213	-0.017
rf3	0.061	-0.012	-0.331	0.001	-0.098	-0.066	0.062
rf4	0.060	-0.031	-0.421	-0.043	-0.243	-0.143	0.093
rf1	0.046	-0.139	-0.406	0.043	-0.159	-0.091	0.138
rd2	0.070	0.013	-0.132	0.002	-0.017	-0.011	0.180
rd3	0.191	0.145	0.147	0.083	0.125	0.188	-0.005
rd4	0.009	-0.014	0.299	0.091	0.238	0.222	-0.041
rd7	0.215	0.216	0.204	0.266	0.253	0.330	-0.002
loc1	0.174	0.125	0.172	0.172	0.051	0.012	0.037
loc2	0.156	0.246	0.278	0.213	0.156	0.104	-0.019

	sf3	*sf4*	*tf1*	*tf2*	*tf3*	*tf4*	*rf2r*	*rf3*
pp1								
pp2								
pp4								
usi1								
usi2								
usi3								
mk1r								
mk2								
mk3								
ss1								
ss2								
ss3								
ms1								
ms2								
ms3r								
pdb1								
pdb2								
pdb3								
pdb4								
pdb6								
ab2r								
ab3								
ab4								
ab5								
ab6								
ab7								
ab8								
sf2								
sf3	1.000							
sf4	0.610	1.000						
tf1	0.313	0.325	1.000					
tf2	0.322	0.336	0.706	1.000				
tf3	0.381	0.362	0.519	0.585	1.000			
tf4	0.319	0.393	0.633	0.653	0.619	1.000		
rf2r	0.071	0.069	0.148	0.194	0.148	0.155	1.000	
rf3	0.262	0.289	0.447	0.513	0.430	0.477	0.331	1.000
rf4	0.277	0.381	0.477	0.540	0.484	0.546	0.415	0.644
rf1	0.266	0.302	0.446	0.546	0.493	0.531	0.221	0.563
rd2	0.163	0.208	0.315	0.294	0.181	0.345	0.179	0.219
rd3	-0.018	0.033	0.048	0.087	-0.084	0.092	-0.019	0.105
rd4	-0.114	-0.154	-0.126	-0.174	-0.142	-0.090	-0.323	-0.174
rd7	0.085	0.058	0.060	0.115	0.060	0.052	-0.255	0.054
loc1	0.089	0.051	0.029	0.064	-0.052	0.035	0.073	0.022
loc2	0.093	-0.025	-0.058	-0.022	-0.093	-0.044	-0.037	-0.068

	rf4	*rf1*	*rd2*	*rd3*	*rd4*	*rd7*	*loc1*	*loc2*
pp1								
pp2								
pp4								
usi1								
usi2								
usi3								
mk1r								
mk2								
mk3								
ss1								
ss2								
ss3								
ms1								
ms2								
ms3r								
pdb1								
pdb2								
pdb3								
pdb4								
pdb6								
ab2r								
ab3								
ab4								
ab5								
ab6								
ab7								
ab8								
sf2								
sf3								
sf4								
tf1								
tf2								
tf3								
tf4								
rf2r								
rf3								
rf4	1.000							
rf1	0.517	1.000						
rd2	0.325	0.207	1.000					
rd3	0.107	-0.063	0.398	1.000				
rd4	-0.237	-0.229	-0.088	0.079	1.000			
rd7	-0.022	0.053	-0.025	0.036	0.203	1.000		
loc1	0.119	-0.033	0.071	0.122	0.065	0.072	1.000	
loc2	-0.071	-0.091	-0.006	0.068	0.021	0.159	0.449	1.000

Appendix C

Common Method Variance - CFA Model



ab2r
ab3
ab4
ab5
ab6
ab7
ab8
mk2r
mk3r
pdb1
pdb2
pdb3
pdb4
pdb6
AStyle
MK
PDStyle
SF
TF
RF
PERF
sf2
sf3
sf4
tf1
tf2
tf3
tf4
rf1
rf2r
rf3
rf4
pp1
pp2
pp4
7.6e-02
.35

References

Abran, Alain and Pierre Bourque. 2004. SWEBOK: Guide to the Software Engineering Body of Knowledge. Los Alamitos, California: IEEE Computer Society.

Agarwal, Nitin and Urvashi Rathod. 2006. "Defining 'success' for Software Projects: An Exploratory Revelation." International Journal of Project Management 24 (4): 358-370.

Akgun, Ali E., John C. Byrne, Halit Keskin, and Gary S. Lynn. 2006. "Transactive Memory System in New Product Development Teams." IEEE Transactions on Engineering Management 53 (1): 95-111.

Akgun, Ali E., Halit Keskin, and John Byrne. 2012. "Antecedents and Contingent Effects of Organizational Adaptive Capability on Firm Product Innovativeness." Journal of Product Innovation Management 29 (S1): 171-189.

Akgun, Ali E. and Gary S. Lynn. 2002a. "Antecedents and Consequences of Team Stability on New Product Development Performance." Journal of Engineering and Technology Management 19 (3): 263-286.

———. 2002b. "New Product Development Team Improvisation and Speed-to-Market: An Extended Model." European Journal of Innovation Management 5 (3): 117-129.

Aladwani, Adel M. 2002. "An Integrated Performance Model of Information Systems Projects." Journal of Management Information Systems 19 (1): 185-210.

Allen, Gove and Jeffrey Parsons. 2010. "Is Query Reuse Potentially Harmful? Anchoring and Adjustment in Adapting Existing Database Queries." Information Systems Research 21 (1): 56-77.

Alter, Steven. 2006. The Work System Method: Connecting People, Processes, and IT for Business Results. Larkspur California: The Work Systems Press.

Alter, Steven and Susan Sherer. 2004. "A General, but Readily Adaptable Model of Information System Risk." Communications of the AIS 14 (1): 1-28.

Ambler, Scott. "Best Practices for Agile/Lean Documentation.", accessed 01/04, 2013, http://www.agilemodeling.com/essays/agileDocumentationBestPractices.htm.

Angst, Corey M. and Ritu I. Agarwa. 2009. "Adoption of Electronic Health Records in the Presence of Privacy Concerns: The Elaboration Likelihood Model and Individual Persuasion." MIS Quarterly 33 (2): 339-370.

Appan, Radha and Glenn J. Browne. 2010. "Investigating Retrieval-Induced Forgetting during Information Requirements Determination." Journal of the Association for Information Systems 11 (5): 250-275.

Armitage, John. 2004. "Are Agile Methods Good for Design?" Interactions 11 (1): 14-23.

Arrow, Holly and Joseph E. McGrath. 1993. "Membership Matters how Member Change and Continuity Affect Small Group Structure, Process, and Performance." Small Group Research 24 (3): 334-361.

Ashforth, Blake E. and Fred Mael. 1989. "Social Identity Theory and the Organization." Academy of Management Review 14 (1): 20-39.

Austin, Robert D. and Lee Devin. 2009. "Research Commentary—Weighing the Benefits and Costs of Flexibility in Making Software: Toward a Contingency Theory of the Determinants of Development Process Design." Information Systems Research 20 (3): 462-477.

Avison, David E. and Guy Fitzgerald. 2003. "Where Now for Development Methodologies?" Communications of the ACM 46 (1): 78-82.

Barber, Elizabeth and James Warn. 2005. "Leadership in Project Management: From Firefighter to Firelighter." Management Decision 43 (7/8): 1032-1039.

Barker, James R. 1993. "Tightening the Iron Cage: Concertive Control in Self-Managing Teams." Administrative Science Quarterly 38 (3): 408-437.

Barker, James R., and Phillip K. Tompkins. 1994. "Identification in the Self-Managing Organization Characteristics of Target and Tenure." Human Communication Research 21 (2): 223-240.

Barki, Henri, Suzanne Rivard, and Jean Talbot. 1993. "Toward an Assessment of Software Development Risk." Journal of Management Information Systems 10 (2): 203-225.

Barki, Henri, Suzanne Rivard, and Jean Talbot. 2001. " An Integrative Contingency Model of Software Project Risk Management." Journal of Management Information Systems 17 (4): 37-70.

Bartsch, Vera, Mark Ebers, and Indre Maurer. 2012. "Learning in Project-Based Organizations: The Role of Project Teams' Social Capital for Overcoming

Barriers to Learning." International Journal of Project Management 31: 239-251.

Bates, Scott C. and Jared M. Cox. 2008. "The Impact of Computer Versus paper–pencil Survey, and Individual Versus Group Administration, on Self-Reports of Sensitive Behaviors." Computers in Human Behavior 24 (3): 903-916.

Batra, Dinesh, Weidong Xia, Debra VanderMeer, and Kaushik Dutta. 2010. "Balancing Agile and Structured Development Approaches to Successfully Manage Large Distributed Software Projects: A Case Study from the Cruise Line Industry. " Communications of the Association for Information Systems 27 (1): 379-394.

Beck, Kent. 1999. Extreme Programming Explained: Embrace Change. First ed.

Becker, H. S. 1998. Tricks of the Trade: How to Think about Your Research while You'Re Doing it. Chicago, Illinois: University of Chicago Press.

Benaroch, Michel. 2002. "Managing Information Technology Investment Risk: A Real Options Perspective." Journal of Management Information Systems 19 (2): 43-84.

Besner, Claude and Brian Hobbs. 2012. "An Empirical Identification of Project Management Toolsets and a Comparison among Project Types." Project Management Journal 43 (5): 24-46.

Bhaskar, Roy. 1975. A Realist Theory of Science. First ed. New York, NY: Routledge.

Bierly, Paul E. and J-C Spender. 1995. "Culture and High Reliability Organizations: The Case of the Nuclear Submarine." Journal of Management 21 (4): 639-656.

Block, Richard A. and David R. Harper. 1991. "Overconfidence in Estimation: Testing the Anchoring-and-Adjustment Hypothesis." Organizational Behavior and Human Decision Processes 49 (2): 188-207.

Boehm, Barry. 2002. "Get Ready for Agile Methods, with Care." Computer 35 (1): 64-69.

———. 1986. "A Spiral Model of Software Development and Enhancement." ACM SIGSoft Software Engineering Notes 11 (4): 14-24.

———. 1988. "A Spiral Model of Software Development and Enhancement." Computer 21 (5): 61-72.

———. 2006. "A View of 20th and 21st Century Software Engineering."Proceedings of the 28th International Conference on Software Engineering (ICSE'06).May 20-28, 2006.

Boehm, Barry. 1989. Software Risk Management. Los Alamitos, CA: CS Press.

Boehm, Barry and Richard Turner. 2003. "Using Risk to Balance Agile and Plan-Driven Methods." Computer 36 (6): 57-66.

Boons, Frank and Marjolein Berends. 2001. "Stretching the Boundary: The Possibilities of Flexibility as an Organizational Capability in Industrial Ecology." Business Strategy and the Environment 10 (2): 115-124.

Bresman, Henrik. 2010. "External Learning Activities and Team Performance: A Multimethod Field Study." Organization Science 21 (1): 81-96.

Briand, Lionel C. 2003. "Software Documentation: How Much is enough?"Proceedings. Seventh European Conference on Software Maintenance and Reengineering, 2003.

Brusoni, Stefano and Andrea Prencipe. 2002. "Managing Knowledge in Loosely Coupled Networks: Exploring the Links between Product and Knowledge Dynamics." Journal of Management Studies 38 (7): 1019-1035.

Bunderson, J. Stuart and Peter Boumgarden. 2010. "Structure and Learning in Self-Managed Teams: Why "Bureaucratic" Teams can be Better Learners." Organization Science 21 (3): 609-624.

Burnes, Bernard. 2004. "Kurt Lewin and Complexity Theories: Back to the Future?" Journal of Change Management 4 (4): 309-325.

Busby, J. S. 2001. "Practices in Design Concept Selection as Distributed Cognition." Cognition, Technology & Work 3 (3): 140-149.

Cacciatori, Eugenia. 2008. "Memory Objects in Project Environments: Storing, Retrieving and Adapting Learning in Project-Based Firms." Research Policy 37: 1591-1601.

Cameron, Kim S. 1986. "Effectiveness as Paradox: Consensus and Conflict in Conceptions of Organizational

Effectiveness." Management Science 32 (5): 539-553.

Caniëls, Marjolein C. J. and Ralph J. J. M. Bakens. 2012. "The Effects of Project Management Information Systems on Decision Making in a Multi Project Environment." International Journal of Project Management 30 (2): 162-175.

Carbonell, Pilar and Ana I. Rodriguez. 2006. "Designing Teams for Speedy Product Development: The Moderating Effect of Technological Complexity." Journal of Business Research 59 (2): 225-232.

Card, David N., Frank E. McGarry, and Gerald T. Page. 1987. "Evaluating Software Engineering Technologies." IEEE Transactions on Software Engineering SE13 (7): 845-851.

Carroll, John M., Gregorio Convertino, Mary Beth Rosson, and Craig H. Ganoe. 2008. "Toward a Conceptual Model of Common Ground in Teamwork." In Macrocognition in Teams: Theories and Methodologies, edited by Michael P. Letsky, Norman W. Warner, Stephen M. Fiore and C. A. P. Smith, 87-105. Hampshire, England: Ashgate.

Ceschi, Martina, Alberto Sillitti, Giancarlo Succi, and Stefano De Panfilis. 2005. "Project Management in Plan-Based and Agile Companies." IEEE Software 22 (3): 21-27.

Chandrasekaran, Aravind and Anant Mishra. 2012. "Task Design, Team Context, and Psychological Safety: An Empirical Analysis of R&D Projects in High Technology Organizations." Production and Operations Management 21 (6): 977-996.

Chapman, N., M. Fox, E. Keravnou, M. Lee, M. Levene, D. Long, P. Rounce, R. Offen, P. Samet, and R. Winder. 1993. "'Slick Systems' and 'Happy Hackers': Experience with Group Projects at UCL." Software Engineering Journal 8 (3): 132-136.

Chau, Thomas and Frank Maurer. 2004. "Knowledge Sharing in Agile Software Teams." In Logic Versus Approximation, Lecture Notes in Computer Science, edited by W. Lenski. Vol. 3075, 173-183. Berlin Heidelberg: Springer.

Chen, Yi-Da, Susan A. Brown, Paul Jen-Hwa Hu, Chwan-Chuen King, and Hsinchun Chen. 2011. "Managing Emerging Infectious Diseases with Information Systems: Reconceptualizing Outbreak Management through the Lens of Loose Coupling." Information Systems Research 22 (3): 447-468.

Cheng, Peter C. H. 2002. "Electrifying Diagrams for Learning: Principles for Complex Representational Systems." Cognitive Science 26 (6): 685-735.

Chin, Gary. 2004. Agile Project Management. New York, NY: Amacon.

Chittister, Clyde and Yacov Y. Haimes. 1994. "Assessment and Management of Software Technical Risk." IEEE Transactions on Systems, Man and Cybernetics, 24 (2): 187-202.

Chong, Jan and Rosanne Siino. 2006. "Interruptions on Software Teams: A Comparison of Paired and Solo Programmers." Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work. ACM, 2006." Banff, Alberta, Canada, November 4–8, 2006.

Choppella, Venkatesh, Arijit Sengupta, Edward L. Robertson, and Steven D. Johnson. 2007. "Preliminary Explorations in Specifying and Validating Entity-Relationship Models in PVS."AFM '07 Proceedings of the Second Workshop on Automated Formal Methods.ACM, .

Chou, Shih-Wei and Mong-Young He. 2011. "The Factors that Affect the Performance of Open Source Software development–the Perspective of Social Capital and Expertise Integration." Information Systems Journal 21 (2): 195-219.

Choudhury, Vivek and Rajiv Sabherwal. 2003. "Portfolios of Control in Outsourced Software Development Projects." Information Systems Research 14 (3): 291-314.

Cleland, David I. and Lewis R. Ireland. 2006. "The Evolution of Project Management (Chapter 1)." In Global Project Management Handbook: Planning, Organizing, and Controlling International Projects (Second Edition), edited by David I. Cleland and Lewis R. Ireland, 1-3-1-19. New York, NY: McGraw Hill.

Cockburn, Alistair and Jim Highsmith. 2001. "Agile Software Development: The People Factor." Computer 34 (11): 131-133.

Cooke, Nancy J., Jamie C. Gorman, and Jennifer L. Winner. 2007. "Team Cognition." In Handbook of Applied Cognition, edited by Francis T. Durso, Raymond S. Nickerson, Susan T. Dumais, Stephan Lewandowsky and Timothy J. Perfect. Second ed., 240-268. West Sussex, England: John Wiley & Sons.

Cooke, Nancy J., Eduardo Salas, Janice A. Cannon-Bowers, and Ruby Stout. 2000. "Measuring Team Knowledge." Human Factors 42: 151-173.

Cooper, JackS. 1984. "Software Development Management Planning." IEEE Transactions on Software Engineering SE-10 (1): 22-26.

Cort, G., J. A. Goldstone, R. O. Nelson, R. B. Poore, L. Miller, and D. M. Barrus. 1985. "A Development Methodology for Scientific Software." IEEE Transactions on Nuclear Science 32 (4): 1439-1443.

Crichton, Margaret T., Kristina Lauche, and Rhonda Flin. 2008. "Learning from Experience: Incident Managment Team Leader Training." In Naturalistic Decision Making and Macrocognition, edited by J. M. Schraagen, L. G.

Militello, T. Ormerod and R. Lipshitz, 103-120. Hampshire, England: Ashgate.

Cullom, Charmayne and Richard Cullom. 2006. "Software Development: Cowboy Or Samurai." Communications of the IIMA 6 (2): 1-8.

Curtis, B., H. Krasner, and N. Iscoe. 1988. "A Field Study of the Software Design Process for Large Systems." Communications of the ACM 31 (11): 1268-1287.

Cyert, Richard M. and James G. March. 1992. A Behavioral Theory of the Firm. Second ed. Cambridge, Mass: Blackwell.

Daft, Richard L. and Robert H. Lengel. 1986. "Organizational Information Requirements, Media Richness and Structural Design." Management Science 32 (5): 554-571.

de Bakker, Karel, Albert Boonstra, and Hans Wortmann. 2010. "Does Risk Management Contribute to IT Project Success? A Meta-Analysis of Empirical Evidence." International Journal of Project Management 28 (5): 493-503.

De Meyer, Arnoud, CHristoph H. Loch, and Michael T. Pich. 2002. "Managing Project Uncertainty: From Variation to Chaos." MIT Sloan Management Review 30 (3): 60-67.

De Witt, Anton. 1988. "Measurement of Project Management Success." International Journal of Project Management 6 (3): 164-170.

Deeter-Schmelz, Dawn R. and Rosemary P. Ramsey. 2003. "An Investigation of Team Information Processing in Service Teams: Exploring the Link between Teams and Customers." Journal of the Academy of Marketing Science 31 (4): 409-424.

Dennis, Alan R., R. N. Burns, and R. Brent Gallupe. 1987. "Phased Design: A Mixed Methodology for Application System Development." ACM SIGMIS Database 18 (4): 31-37.

DeRosa, Darleen M., Carter L. Smith, and Donald A. Hantula. 2007. "The Medium Matters: Mining the Long-Promised Merit of Group Interaction in Creative Idea Generation Tasks in a Meta-Analysis of the Electronic Group Brainstorming Literature." Computers in Human Behavior 23 (3): 1549-1581.

DeSanctis, Gerardine and Marshall Scott Poole. 1994. "Capturing the Complexity in Advanced Technology use: Adaptive Structuration Theory." Organization Science 5 (2): 121-147.

Dickson, Mark. 2013. "PMI Fact File." PMI Today, July 2013, 4.

Dieckmann, P., S. M. Friis, A. Lippert, and D. Østergaard. 2012. "Goals, Success Factors, and Barriers for Simulation-Based Learning A Qualitative Interview Study in Health Care." Simulation & Gaming 43 (5): 627-647.

Doherty, Michael J. 2012. "Examining Project Manager Insights of Agile and Traditional Success Factors for Information Technology Projects: A Q-Methodology Study." PhD, Marian University, Doctoral Dissertation.

Dror, I. E. and S. Harnad. 2008. Cognition Distributed: How Cognitive Technology Extends our Minds. Philadelphia, PA: John Benjamins Publishing Co.

Dvir, D. and T. Lechler. 2004. "Plans are Nothing, Changing Plans is Everything: The Impact of Changes on Project Success." Research Policy 33: 1-15.

Dvir, Dov, Stan Lipovetsky, Aaron Shenhar, and Asher Tishler. 1998. "In Search of Project Classification: A Non-Universal Approach to Project Success Factors." Research Policy 27: 915-935.

Eisenhardt, Kathleen M. and Jeffrey A. Martin. 2000. "Dynamic Capabilities: What are they?" Strategic Management Journal 21 (10-11): 1105-1121.

Ellis, Aleksander PJ. 2006. "System Breakdown: The Role of Mental Models and Transactive Memory in the Relationship between Acute Stress and Team Performance." Academy of Management Journal 49 (3): 576-589.

Elshamy, Ahmed and Amr Elssamadisy. 2006. "Divide After You Conquer: An Agile Software Development Practice for Large Projects." In Extreme Programming and Agile Processes in Software Engineering, Lecture Notes in Computer Science, edited by P. Abrahamsson, M. Marchesi and G. Succi. Vol. 4044, 164-168. Berlin Heidelberg: Springer.

Engwall, Mats and Charlotta Svensson. 2004. "Cheetah Teams in Product Development: The most Extreme Form of Temporary Organization?" Scandinavian Journal of Management 20 (3): 297-317.

Epley, Nicholas and Thomas Gilovich. 2006. "The Anchoring-and-Adjustment Heuristic: Why the Adjustments are Insufficient." Psychological Science 17 (4): 311-318.

Espinosa, J. A., S. A. Slaughter, R. E. Kraut, and J. D. Herbsleb. 2007. "Familiarity, Complexity, and Team Performance in Geographically Distributed Software Development." Organization Science 18 (4): 613-630.

Faraj, Samer and Lee Sproull. 2000. "Coordinating Expertise in Software Development Teams." Management Science 46 (12): 1554-1568.

Fernandez, Daniel J. and John D. Fernandez. 2009. "Agile Project management–agilism Versus Traditional Approaches." Journal of Computer Information Systems 49 (2): 10-17.

Ferreira, Carlos and Jason Cohen. 2008. "Agile Systems Development and Stakeholder Satisfaction: A South African Emprical Study." Wilderness, South Africa, ACM, 6-9 October 2008.

Firestone, William A. 1984. "The Study of Loose Coupling: Problems, Progress, and Prospects." NewOrleans, LA, 68th Annual Meeting of the American Educational Research Association.US Department of Education, April 23-27.

Fitzgerald, Brian. 2000. "Systems Development Methodologies: The Problem of Tenses." Information Technology & People 13 (3): 174-185.

Flach, John M. 2008. "Mind the Gap: A Skeptical View of Macrocognition." In Naturalistic Decision Making and Macrocognition, edited by J. M. Schraagen, L. G. Militello, T. Ormerod and R. Lipshitz, 27-40. Hampshire, England: Ashgate.

Flor, Nick and Edwin Hutchins. 1992. "Analyzing Distributed Cognition in Software Teams: A Cast Study of Collaborative Programming during Adaptive Software Maintenance." In Empirical Studies of Programmers: Fourth Workshop, edited by J. Koenemann-Belliveau, T. Moher and S. Robertson, 36-64. Norwood, NJ: Ablex.

Foltz, Peter W., Cheryl A. Bolstad, Haydee M. Cuevas, Marita Franzke, Mark Rosenstein, and Anthony M. Costello. 2008. "Measuring Situational Awareness through Automated Communication Analysis." In Macrocognition in Teams, edited by Michael P. Letsky, Norman W. Warner, Stephen M. Fiore and C. A. P. Smith, 259-275. Hampshire, England: Ashgate.

Forsberg, Kevin, Hal Mooz, and Howard Cotterman. 2000. Visualizing Project Management: A Model for Business and Technical Success. New York, NY: Wiley.

Fowler, K. 1999. "Documentation-do Or Die!" IEEE Instrumentation & Measurement Magazine 2 (2): 53-56.

Fowler, Martin and Kent Beck . 1999. Refactoring: Improving the Design of Existing Code. Addison-Wesley.

Fox, T. L. 1997. "An Examination of the Effect of Decision Style on the use of a Computerized Project Management Tool."University of North Texas - Doctoral Dissertation.

Fried, Louis. 1992. "The Rules of Project Management." Information Systems Management 9 (3).

Frohlich, Markham T. 2002. "Techniques for Improving Response Rates in OM Survey Research." Journal of Operations Management 20 (1): 53-62.

Furnham, Adrian and Hua Chu Boo. 2011. "A Literature Review of the Anchoring Effect." The Journal of Socio-Economics 40 (1): 35-42.

Gareis, Roland. 2006. "Project Management: A Business Process of the Project-Oriented Company (Chapter 2)." In Global Project Management Handbook: Planning, Organizing, and Controlling International Projects (Second Edition), edited by David I. Cleland and Lewis R. Ireland, 2-1-2-26. New York, NY: McGraw Hill.

Gefen, D., D. W. Straub, and M. C. Boudreau. 2000. "Structural Equation Modeling and Regression: Guidelines for Research Practice." Communications of the AIS 4 (7): 1-77.

Gemino, Andrew, Blaize Horner Reich, and Chris Sauer. 2008. "A Temporal Model of Information Technology Project Performance." Journal of Management Information Systems 24 (3): 9-44.

George, Joey F., Kevin Duffy, and Manju Ahuja. 2000. "Countering the Anchoring and Adjustment Bias with Decision Support Systems." Decision Support Systems 29 (2): 195-206.

Germain, Éric and Pierre N. Robillard. 2005. "Engineering-Based Processes and Agile Methodologies for Software Development: A Comparative Case Study." Journal of Systems and Software 75 (1): 17-27.

Gilb, Tom. 1976. Software Metrics Little, Brand & Co.

Glass, Robert L. 2004. "Matching Methodology to Problem Domain." Communications of the ACM 47 (5): 19-21.

Goodhue, Dale L., William Lewis, and Ron Thompson. 2012. "Does PLS have Advantages for Small Sample Size Or Non-Normal Data?" MIS Quarterly 36 (3): 891-1001.

Gopal, Anandasivam and Sanjay Gosain. 2010. "The Role of Organizational Controls and Boundary Spanning in Software Development Outsourcing: Implications for Project Performance." Information Systems Research 21 (4): 960-982.

Graham, Alan K. 2000. "Beyond PM 101: Lessons for Managing Large Development Programs." Project Management Journal 31 (4): 7-19.

Grossman, Rebecca and Eduardo Salas. 2011. "The Transfer of Training: What really Matters." International Journal of Training & Development 15 (2): 103-120.

Grote, Gudela, Rogert L. Helmreich, Oliver Strater, Ruth Hausler, Eniko Zala-Mezo, and J. Bryan Sexton. 2004. "Setting the Stage: Characteristics of Organizations, Teams and Tasks Influencing Team Processes." In Group Interaction in High Risk Environments, edited by Rainer Dietrich and Traci Michelle Childress, 111-140. Hampshire, England: Ashgate.

Hackman, J. Richard. 1987. "The Design of Work Teams." In Handbook of Organizational Behavior, edited by J. W. Lorsch, 315-342. Englewood Cliffs, NJ: Prentice-Hall.

Hällgren, Markus and Joakim Lillieskold. 2006. "Three Modes of Deviation Handling: Coping with Unexpected Events in Project Management."Technology Management for the Global Future, 2006. PICMET 2006. Vol. 5.8-13 July 2006.

Hällgren, Markus and Eva Maaninen-Olsson. 2005. "Deviations, Ambiguity and Uncertainty in a Project-Intensive Organization." Project Management Journal 36 (3): 17-26.

Hallgren, Markus and Anders Soderholm. 2010. "Orchestrating Deviations in Global Projects: Projects-as-Practice Observations." Scandinavian Journal of Management 26 (4): 352-367.

Hansen, Morten T. 1999. "The Search-Transfer Problem: The Role of Weak Ties in Sharing Knowledge Across Organization Subunits." Administrative Science Quarterly 44 (1): 82-111.

Harris, Michael L., Rosann Webb Collins, and Alan R. Hevner. 2009b. "Control of Flexible Software Development Under Uncertainty." Information Systems Research 20 (3): 400-419.

Harris, Michael L., Alan R. Hevner, and Rosann Webb Collins. 2009a. "Controls in Flexible Software Development." Communications of the Association for Information Systems 24 (43): 757-776.

Hayne, Steven and C. A. P. Smith. 2008. "Collaboration, Training and Pattern Recognition." In Macrocognition in Teams: Theories and Methodologies, edited by Michael P. Letsky, Norman W. Warner, Stephen M. Fiore and C. A. P. Smith, 35-50. Hampshire, England: Ashgate.

Hazlehurst, B., P. N. Gorman, and C. K. McMullen. 2008. "Distributed Cognition: An Alternative Model of Cognition for Medical Informatics." International Journal of Medical Informatics 77: 226-234.

He, J., B. S. Butler, and W. R. King. 2007. "Team Cognition: Development and Evolution in Software Project Teams." Journal of Mangement Information Systems 24 (2): 261-292.

Heylighen, A., F. Heylighen, J. Bollen, and M. Casaer. 2007. "Distributed (Design) Knowledge Exchange." Ai & Society 22 (2): 145-154.

Higgins, Monica C., Jennie Weiner, and Lissa Young. 2012. "Implementation Teams: A New Lever for Organizational Change." Journal of Organizational Behavior 33 (3): 366-388.

High, R., G. Krishnan, and M. Sanchez. 2008. "Creating and Maintaining Coherence in Loosely Coupled Systems." IBM Systems Journal 47 (3): 357-376.

Highsmith, Jim. 2000. "Retiring Lifecycle Dinosaurs: Using Adaptive Software Development to Meet the Challenges of a High-Speed High-Change Environment." Software Testing & Quality Engineering: 22-28.

Highsmith, Jim and Alistair Cockburn. 2001. "Agile Software Development: The Business of Innovation." Computer 34 (9): 120-122.

Hinds, Pamela and Cathleen McGrath. 2006. "Structures that Work: Social Structure, Work Structure and Coordination Ease in Geographically Distributed Teams." Banff, Alberta, Canada, Proceedings of the 2006 20th Anniversary Conference on Computer Supported Cooperative Work.November 4-8, 2006.

Hoda, R., J. Noble, and S. Marshall. 2012. "Self-Organizing Roles on Agile Software Development Teams." IEEE Transactions on Software Engineering (pre-press, online publication): 1-27.

Homburg, C., M. Klarmann, M. Reimann, and O. Schilke. 2012. "What Drives Key Informant Accuracy?" Journal of Marketing Research 49 (4): 594-608.

Hsu, J. S., J. Y. Chang, G. Klein, and J. J. Jiang. 2011. "Exploring the Impact of Team Mental Models on Information Utilization and Project Performance in System Development." International Journal of Project Management 29 (1): 1-12.

Hughes, Chris. " User Documentation in Agile and Scrum Projects.", last modified November 19th, 2009, accessed December 14, 2012, http://www.straightforward.se/storyserver/user_documentation_agile_scrum_projects_problems_and_solutions.

Ika, Lavagnon, A. 2009. "Project Success as a Topic in Project Management Journals." Project Management Journal 40 (4): 6-19.

Imamoglu, Oksan and Sitki Gozlu. 2008. "The Sources of Success and Failure of Information Technology Projects: Project Manager's Perspective."Portland International Conference on Management of Engineering & Technology, 2008. PICMET 2008.27-31 July, 2008.

Ives, Blake, Margrethe H. Olson, and Jack J. Baroudi. 1983. "The Measurement of User Information Satisfaction." Communications of the ACM 26 (10): 785-793.

Jiang, Li and Armin Eberlein. 2009. "An Analysis of the History of Classical Software Development and Agile Development

."IEEE International Conference on Systems, Man and Cybernetics, 2009. SMC 2009..

Jones, William and Brian H. Ross. 2010. "Personal Information Management." In Macrocognition Metrics and Scenarios: Design and Evaluation for Real-World Teams, edited by Emily S. Patterson and Janet E. Miller, 162-178. Surrey, England: Ashgate.

Jones, Matthew R. and Helena Karsten. 2008. "Giddens Structuration Theory and Information Systems Research." MIS Quarterly 32 (1): 127-157.

Jugdev, Kam and Ralf Muller. 2005. "A Retrospective Look at our Evolving Understanding of Project Success." Project Management Journal 36 (4): 19-31.

Kaplan, Robert S. and David P. Norton. 1992. "The Balanced Scorecard: Measures that Drive Performance." Harvard Business Review 70 (1): 71-80.

Kappelman, Leon A., Robert McKeeman, and Lixuan Zhang. 2006. "Earling Warning Signs of IT Project Failure: The Dominant Dozen." Information Systems Management 23 (4): 31-36.

Karau, Steven J. and Kipling D. Williams. 1993. "Social Loafing: A Meta-Analytic Review and Theoretical Integration." Journal of Personal and Social Psychology 65 (4): 681-706.

Karlstrom, Daniel and Per Runeson. 2005. "Combining Agile Methods with Stage-Gate Project Management." IEEE Software 22 (3): 43-49.

———. 2006. "Integrating Agile Software Development into Stage-Gate Managed Product Development." Empirical Software Engineering 11 (2): 203-225.

Kärreman, Dan and Mats Alvesson. 2004. "Cages in Tandem: Management Control, Social Identity, and Identification in a Knowledge-Intensive Firm." Organization 11 (1): 149-175.

Kautz, Karlheinz. 2010. "The Impact of Pricing and Opportunistic Behavior on Information Systems Development." Journal of Information Technology Theory and Application (JITTA) 10 (3): 24-41.

Kautz, Karlheinz and Sabine Zumpe. 2008. "Just enough Structure at the Edge of Chaos: Agile Information System Development in Practice." Limerick Ireland, 9th International Conference, XP 2008

.Springer Berlin Heidelberg, June 10-14, 2008.

Keil, Mark, Richard Mixon, Timo Saarinen, and Virpi Tuunainen. 1994. "Understanding Runaway Information Technology Projects: Results from an International Research Program Based on Escalation Theory." Journal of Management Information Systems 11 (3): 65-85.

Keith, Mark, Haluk Demirkan, and Michael Goul. 2009. "Understanding Coordination in IT Project-Based Environments: An Examination of Team Cognition and Virtual Team Efficacy."42nd Hawaii International Conference on system Sciences, 2009 (HICSS'09).5-8 Jan. 2009.

Kerr, Norbert L. and R. Scott Tindale. 2004. "Group Performance and Decision Making." Annual Review of Psychology 55: 623-655.

Kerzner, Harold. 2004. Advanced Project Management: Best Practices on Implementation. Hoboken, NJ: John Wiley & Sons, Inc.

Keskin, Halit. 2009. "Antecedents and Consequences of Team Memory in Software Development Projects." Information & Managment 46: 388-396.

Kirsch, Laurie J. 1997. "Portfolios of Control Modes and IS Project Management." Information Systems Research 8 (3): 215-239.

Kirsch, Laurie J., Dong-Gil Ko, and Mark H. Haney. 2010. "Investigating the Antecedents of Team-Based Clan Control: Adding Social Capital as a Predictor." Organization Science 21 (2): 469-489.

Kirsch, Laurie J. 1996. "The Management of Complex Tasks in Organizations: Controlling the Systems Development Process." Organization Science 7 (1): 1-21.

Klein, Gary and Robert T. Hoffman. 2008. "Macrocognition, Mental Models, and Cognitive Task Analysis Methodology." In Naturalistic Decision Making and Macrocognition, edited by J. M. Schraagen, L. G. Militello, T. Ormerod and R. Lipshitz, 57-80. Hampshire, England: Ashgate.

Kohli, Rajiv and William J. Kettinger. 2004. "Informating the Clan: Controlling Physicians' Costs and Outcomes." MIS Quarterly 28 (3): 363-394.

Koskela, Lauri and Gregory Howell. 2002. "The Theory of Project Management: Explanation to Novel Methods." Gramado, Brazil, Proceedings, Int. Group for Lean Construction 10th Annual Conf.(IGLC-10). International Group for Lean Construction., August 2002.

Kreiner, Kristian. 1992. "The Postmodern Epoch of Organization Theory." International Studies of Management & Organization 22 (2): 37-52.

Kutsch, E. and M. Hall. 2005. "Intervening Conditions on the Management of Project Risk: Dealing with Uncertainty in Information Technology Projects." International Journal of Project Management 23 (8): 591-599.

Kwak, Young Hoon. 2005. "Brief History of Project Management." In The Story of Managing Projects: An Interdisciplinary Approach, edited by Elias G. Carayannis, Young Hoon Kwak and Frank T. Anbari, 1-9. Westport Connecticut: Praeger.

Kydd, Christine T. 1989. "Understanding the Information Content in MIS Management Tools." MIS Quarterly 13 (3): 277-290.

Lamm, Helmut and Gisela Trommsdorff. 1973. "Group Versus Individual Performance on Tasks Requiring Ideational Proficiency (Brainstorming): A Review." European Journal of Social Psychology 3 (4): 361-388.

Lance, C. E., B. Dawson, D. Birkelbach, and B. J. Hoffman. 2010. "Method Effects, Measurement Error, and Substantive Conclusions." Organizational Research Methods 13 (3): 435-455.

Larkin, Jill H. and Herbert A. Simon. 1987. "Why a Diagram is (Sometimes) Worth Ten Thousand Words." Cognitive Science 11 (1): 65-100.

Larman, Craig and Victor R. Basili. 2003. "Iterative and Incremental Development: A Brief History." Computer 36 (6): 47-56.

Leach, Larry P. 1999. "Critical Chain Project Management Improves Project Performance." Project Management Journal 30: 39-51.

Lee, Gwanhoo and Weidong Xia. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility." MIS Quarterly 34 (1): 87-114.

Lee, Gwanhoo and Weidong Xia. 2010. "Toward Agile: An Integrated Analysis of Quantitative and Qualitative Field Data on Software Development Agility." MIS Quarterly 34 (1): 87-114.

Leifer, Richard and Andre Delbecq. 1978. "Organizational/environmental Interchange: A Model of Boundary Spanning Activity." Academy of Management Review 3 (1): 40-50.

Leonardi, Paul M. and Diane E. Bailey. 2008. "Transformational Technologies and the Creation of New Work Practices: Making Implicit Knowledge Explicit in Task-Based Offshoring." MIS Quarterly 32 (2): 411-436.

Levardy, Viktor and Tyson R. Browning. 2009. "An Adaptive Process Model to Support Product Development Project Management." IEEE Transactions on Engineering Management 56 (4): 600-620.

Leybourne, Steve. 2010. "Project Management and High-Value Superyacht Projects:An Improvisational and Temporal Perspective." Project Management Journal 41 (1): 17-27.

Lin, Chieh-Peng, Yehuda Baruch, and Wei-Chi Shih. 2012. "Corporate Social Responsibility and Team Performance: The Mediating Role of Team Efficacy and Team Self-Esteem." Journal of Business Ethics 108 (2): 167-180.

Lundin, Rolf A. and Anders Soderholm. 1995. "A Theory of the Temporary Organization." Scandinavian Journal of Management 11 (4): 437-455.

Lyytinen, Kalle, Lars Mathiassen, and Janne Ropponen. 1996. "A Framework for Software Risk Management." Scandinavian Journal of Information Systems 8: 51-68.

Lyytinen, Kalle, Lars Mathiassen, and Janne Ropponen. 1998. "Attention Shaping and Software risk—a Categorical Analysis of Four Classical Risk Management Approaches." Information Systems Research 9 (3): 233-255.

Maaninen-Olsson, E. and T. Mullern. 2009. "A Contextual Understanding of Projects - the Importance of Space and Time." Scandinavian Journal of Management 25: 327-339.

MacCormack, Alan, Roberto Verganti, and Marco Iansiti. 2001. "Developing Products on “Internet Time”: The Anatomy of a Flexible Development Process." Management Science 47 (1): 133-150.

Mackenzie, Adrian and Simon Monk. 2004. "From Cards to Code: How Extreme Programming Re-Embodies Programming as a Collective Practice." Computer Supported Cooperative Work (CSCW) 13 (1): 91-117.

MacKenzie, Scott B. and Philip M. Podsakoff. 2012. "Common Method Bias in Marketing: Causes, Mechanisms, and Procedural Remedies." Journal of Retailing 88 (4): 542-555.

Macomber, Hal and Gregory Howell. 2003. "Linguistic Action: Contributing to the Theory of Lean Construction." Blacksburg, Virginia, .

Madni, Azad M. and Scott Jackson. 2009. "Towards a Conceptual Framework for Resilience Engineering." IEEE Systems Journal 3 (2): 181-191.

Mahaney, Robert C. and Albert L. Lederer. 2003. "Information Systems Project Management: An Agency Theory Interpretation." Journal of Systems and Software 68 (1): 1-9.

Majchrzak, Ann, Arvind Malhotra, and Richard John. 2005. "Perceived Individual Collaboration Know-how Development through Information technology–enabled Contextualization: Evidence from Distributed Teams." Information Systems Research 16 (1): 9-27.

Malhotra, Naresh K., Sung S. Kim, and Ashutosh Patil. 2006. "Common Method Variance in IS Research: A Comparison of Alternative Approaches and a Reanalysis of Past Research." Management Science 52 (12): 1865-1883.

Maruping, Likoebe M., Viswanath Venkatesh, and Ritu Agarwal. 2009. "A Control Theory Perspective on Agile Methodology use and Changing User Requirements." Information Systems Research 20 (3): 377-399.

Mathieu, John E. and Gilad Chen. 2011. "The Etiology of the Multilevel Paradigm in Management Research." Journal of Management 37 (2): 610-641.

Mathur, Gita, Kam Jugdev, and Tak Shing Fung. 2013. "Project Management Assets and Project Management Performance Outcomes: Exploratory Factor Analysis." Management Research Review 36 (2): 112-135.

Maynard, M. Travis, John E. Mathieu, Tammy L. Rapp, and Lucy L. Gilson. 2012. " Something(s) Old and Something(s) New: Modeling Drivers of Global Virtual Team Effectiveness " Journal of Organizational Behavior 33 (3): 342-365.

McChesney, Ian R. and Seamus Gallagher. 2004. "Communication and Co-Ordination Practices in Software Engineering Projects." Information and Software Technology 46: 473-489.

McComb, Sara A. 2008. "Shared Mental Models and their Convergence." In Macrocognition in Teams: Theories and Methodologies, edited by

Michael P. Letsky, Norman W. Warner, Stephen M. Fiore and C. A. P. Smith, 35-50. Hampshire, England: Ashgate.

McHugh, Orla. 2011. "A Study of Clan Control in Agile Software Development Teams."National University of Ireland - Doctoral Dissertation.

Meso, Peter and Radhika Jain. 2006. "Agile Software Development: Adaptive Systems Principles and Best Practices." Information Systems Management 23 (3): 19-30.

Meyerson, Debra and Joanne Martin. 1987. "Cultural Change: An Integration of Three Different Views." Journal of Management Studies 24 (6): 623-647.

Milis, Koen and Roger Mercken. 2004. "The use of the Balanced Scorecard for the Evaluation of Information and Communication Technology Projects." International Journal of Project Management 22 (2): 87-97.

Mingers, J. 2012. "Abduction: The Missing Link between Deduction and Induction. A Comment on Ormerod's 'Rational Inference: Deductive, Inductive and Probabilistic Thinking'." Journal of Operational Research Society 63: 860-861.

Mintzberg, Henry and Alexandra McHugh. 1985. " Strategy Formation in an Adhocracy." Administrative Science Quarterly 30 (2): 160-197.

Mitchell, Victoria L. and Robert W. Zmud. 1999. "The Effects of Coupling IT and Work Process Strategies in Redesign Projects." Organization Science 10 (4): 424-438.

Moe, Nils Brede, Torgeir Dingsøyr, and Emil A. Røyrvik. 2009. "Putting Agile Teamwork to the test–an Preliminary Instrument for Empirically Assessing and Improving Agile Software Development." In Agile Processes in Software Engineering and Extreme Programming., edited by P. Abrahamsson, M. Marchesi and F. Maurer. Vol. 31, 114-123: Springer Berlin Heidelberg.

Moløkken-Østvold, Kjetil and Magne Jørgensen. 2005. "A Comparison of Software Project Overruns - Flexible Versus Sequential Development." IEEE Transactions on Software Engineering 31 (9): 754-766.

Morcos, Maged Sedky. 2005. "Analyzing Failure in Project Management Systems using an Integral Approach of the Fish Bone and Fault Tree Techniques." In The Story of Managing Projects: An Interdisciplinary Approach, edited by Elias G. Carayannis, Young Hoon Kwak and Frank T. Anbari, 152-175. Westport Connecticut: Praeger.

Morrison, P. D., J. H. Roberts, and E. Von Hippel. 2000. "Determinants of User Innovation and Innovation Sharing in a Local Market." Management Science 46 (12): 1513-1527.

Mosier, Kathleen L. 2008. "Technology and "Naturalistic" Decision Making: Myths and Realities." In Naturalistic Decision Making and Macrocognition, edited by J. M. Schraagen, L. G. Militello, T. Ormerod and R. Lipshitz, 41-54. Hampshire, England: Ashgate.

Muller, Matthias M. and Walter F. Tichy. 2001. "Case Study: Extreme Programming in a University Environment."Proceedings of the 23rd International Conference on Software Engineering, 2001. (ICSE 2001).IEEE, .

Muller, Ralf and J. Rodney Turner. 2007. "Matching the Project Manager's Leadership Style to Project Type." International Journal of Project Management 25: 21-32.

Murphy, Alan and Ann Ledwith. 2007. "Project Management Tools and Techniques in High-Technology SMEs." Management Research News 30 (2): 153-166.

Mutch, Alistair. 2010. "Technology, Organization, and structure—a Morphogenetic Approach." Organization Science 21 (2): 507-520.

Nandhakumar, Joe and David E. Avison. 1999. "The Fiction of Methodological Development: A Field Study of Information Systems Development." Information Technology & People 12 (2): 176-191.

Naumann, Justus D. and A. Milton Jenkins. 1982. "Prototyping: The New Paradigm for Systems Development." MIS Quarterly 6 (3): 29-44.

Nelson, W. M. 2003. "Embraceable Processes: Adapting Methods to Fit the Team." IT Professional 5 (3): 42-47.

Nemeth, Christopher P., Richard I. Cook, Michael O'Connor, and P. Allan Klock. 2004. "Using Cognitive Artifacts to Understand Distributed Cognition." IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans, 34 (6): 726-735.

Nerur, S., R. Mahapatra, and G. Mangalaraj. 2005. "Challenges of Migrating to Agile Methodologies." Communications of the ACM 48 (5): 73-78.

Nidumolu, Sarma 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.

Nidumolu, Sarma R. and Mani R. Subramani. 2003. "The Matrix of Control: Combining Process and Structure Approaches to Managing Software Development." Journal of Management Information Systems 20 (3): 159-196.

Noe, Raymond A. and Steffanie L. Wilk. 1993. "Investigation of the Factors that Influence Employees' Participation in Development Activities." Journal of Applied Psychology 78 (2): 291-302.

Olsson, Nils OE. 2006. "Management of Flexibility in Projects." International Journal of Project Management 24 (1): 66-74.

Orlikowski, Wanda J. 1992. "The Duality of Technology: Rethinking the Concept of Technology in Organizations." Organization Science 3 (3): 398-427.

Orr, Ken. 2002. "CMM Versus Agile Development: Religious Wars and Software Development." Agile Project Management Executive Report 3.7.

Orton, J. Douglas and Karl E. Weick. 1990. "Loosely Coupled Systems: A Reconceptualization." Academy of Management Review 15 (2): 203-223.

Ouchi, William G. 1979. "A Conceptual Framework for the Design of Organizational Control Mechanisms." Management Science 25 (9): 833-847.

Overmyer, Scott P. 1990. "The Impact of DOD-Std-2167A on Iterative Design Methodologies: Help Or Hinder?" ACM SIGSoft Software Engineering Notes 15 (5): 50-59.

Pare, Guy. 2004. "Investigating Information Systems with Positivist Case Study Research." Communications of the Association for Information Systems 13: 233-264.

Parnas, David Lorge and Paul C. Clements. 1986. "A Rational Design Process: How and Why to Fake it." IEEE Transactions on Software Engineering 2: 251-257.

Parsons, Jeffrey and Chad Saunders. 2004. "Cognitive Heuristics in Software Engineering Applying and Extending Anchoring and Adjustment to Artifact Reuse." IEEE Transactions on Software Engineering 30 (12): 873-888.

Pavlak, Alex. 2004. "Project Troubleshooting: Tiger Teams for Reactive Risk Management." Project Management Journal 35 (4): 5-14.

Pavlou, Paul A. and Omar A. El Sawy. 2010. "The "Third Hand": IT-Enabled Competitive Advantage in Turbulence through Improvisational Capabilities." Information Systems Research 21 (3): 443-471.

Pawlowski, Suzanne D. and Daniel Robey. 2004. "Bridging User Organizations: Knowledge Brokering and the Work of Information Technology Professionals." MIS Quarterly 28 (4): 645-672.

Pearsall, Matthew J., Aleksander PJ Ellis, and Jordan H. Stein. 2009. "Coping with Challenge and Hindrance Stressors in Teams: Behavioral, Cognitive, and Affective Outcomes." Organizational Behavior and Human Decision Processes 109 (1): 18-28.

Pender, S. 2001. "Managing Incomplete Knowledge: Why Risk Management is Not Sufficient." International Journal of Project Management 19: 79-87.

Perminova, Olga, Magnus Gustafsson, and Kim Wikstrom. 2008. "Defining Uncertainty in Projects - a New Perspective." International Journal of Project Management 26: 73-79.

Persson, John Stouby, Lars Mathiassen, and Ivan Aaen. 2012. "Agile Distributed Software Development: Enacting Control through Media and Context." Information Systems Journal 22 (6): 1-23.

Petter, Stacie, William DeLone, and Ephraim McLean. 2008. "Measuring Information Systems Success: Models, Dimensions, Measures, and Interrelationships." European Journal of Information Systems 17 (236): 263.

Pich, Michael T., Christoph H. Loch, and Arnoud De Meyer. 2002. "On Uncertainty, Ambiguity, and Complexity in Project Management." Management Science 48 (8): 1008-1023.

Pietri, William1. 2009. 10 Rules for Great Development Spaces. Agile Focus, Http://agilefocus.com/2009/04/20/10-Rules-for-Great-Development-Spaces/.

Pikkarainen, M., O. Salo, R. Kuusela, and P. Abrahamsson. 2012. " Strengths and Barriers Behind the Successful Agile deployment—insights from the Three Software Intensive Companies in Finland " Empirical Software Engineering 17: 1-28.

Pikkarainen, M., J. Haikara, O. Salo, P. Abrahamsson, and J. Still. 2008. "The Impact of Agile Practices on Communication in Software Development." Empirical Software Engineering 13: 303-337.

Pinto, Jeffrey K. and Dennis P. Slevin. 1988. "Project Success: Definitions and Measurement Techniques." Project Management Journal 19 (1): 67-72.

PMBOK. 2008. Project Management Body of Knowledge (PMBOK Guide). Forth ed. Newtown Square, Penn: Project Management Institute, Inc.

Pomeroy-Huff, Marsha, Robert Cannon, Timothy A. Chick, Julia Mullaney, and William Nichols. 2009. the Personal Software ProcessSM (PSPSM) Body of Knowledge, Version 2.0. Vol. CMU/SEI-2009-SR-018. Pittsburg PA: Carnegie-Mellon Univ. Software Engineering Institute.

Proença, Teresa. 2010. "Self-Managed Work Teams: An Enabling Or Coercive Nature." The International Journal of Human Resource Management 21 (3): 337-354.

Rai, Arun, Likoebe M. Maruping, and Viswanath Venkatesh. 2009. "Offshore Information Systems Project Success: The Role of Social Embeddedness and Cultural Characteristics." MIS Quarterly 33 (3): 617-641.

Ramesh, Balasubramaniam, Lan Cao, and Richard Baskerville. 2010. "Agile Requirements Engineering Practices and Challenges: An Empirical Study." Information Systems Journal 20 (5): 449-480.

Ramesh, Balasubramaniam, Lan Cao, Kannan Mohan, and Peng Xu. 2006. "Can Distributed Software Development be Agile?" Communications of the ACM 49 (10): 41-46.

Ratanotayanon, S., J. Kotak, and S. E. Sim. 2006. "After the Scrum: Twenty Years of Working without Documentation." San Francisco, CA, Eighteenth International Conference on Software Engineering and Knowledge Engineering (SEKE).

Regev, Sary, Avraham Shtub, and Yakov Ben-Haim. 2006. "Managing Project Risks as Knowledge Gaps." Project Management Journal 37 (5): 17-25.

Reinartz, Werner, Michael Haenlein, and Jörg Henseler. 2009. "An Empirical Comparison of the Efficacy of Covariance-Based and Variance-Based SEM." International Journal of Research in Marketing 26 (4): 332-344.

Ren, Yuqing, Kathleen M. Carley, and Linda Argote. 2006. "The Contingent Effects of Transactive Memory: When is it More Beneficial to Know what Others Know?" Management Science 52 (5): 671-682.

Richardson, Hettie A., Marcia J. Simmering, and Michael C. Sturman. 2009. "A Tale of Three Perspectives Examining Post Hoc Statistical Techniques for Detection and Correction of Common Method Variance." Organizational Research Methods 12 (4): 762-800.

Rijsdijk, Serge A., and Jan van den Ende. 2011. "Control Combinations in New Product Development Projects." Journal of Product Innovation Management 28 (6): 868-880.

Ringle, Christian M., Wende, Sven and Will, Alexander. "SmartPLS 2.0 (Beta)."2005, http://www.smartpls.de.

Ringle, Christian, Marko Sarstedt, and Detmar Straub. 2012. "A Critical Look at the use of PLS-SEM in MIS Quarterly." MIS Quarterly 36 (1): iii-xiv.

Ritterskamp, Carsten. 2011. "The Collaborative Nature of Interruption Handling."44th Hawaii International Conference on System Sciences (HICSS), 2011.4-7 Jan. 2011.

Robert, Lionel P., Alan R. Dennis, and Manju K. Ahuja. 2008. "Social Capital and Knowledge Integration in Digitally Enabled Teams." Information Systems Research 19 (3): 314-334.

Rogelberg, S. G., D. J. Leach, P. B. Warr, and J. L. Burnfield. 2006. "Not another Meeting!" are Meeting Time Demands Related to Employee Well-being?" Journal of Applied Psychology 91 (1): 86-89.

Roloff, Michael E. and Lyn M. Van Swol. 2007. "Shared Cognition and Communication with Group Decision Making and Negotiation." In Communication and Social Cognition: Theories and Methods, edited by David R. Roskos-Ewoldsen and Jennifer L. Monahan, 171-195. Mahwah, NJ: Lawrence Erlbaum Associates.

Romme, A. Georges L. 1999. "Domination, Self-Determination and Circular Organizing." Organization Studies 20 (5): 801-832.

Rouse, William B., Janice A. Cannon-Bowers, and Eduardo Salas. 1992. "The Role of Mental Models in Team Performance in Complex Systems." IEEE Transactions on Systems, Man and Cybernetics 22 (6): 1296-1308.

Rowen, Robert B. 1990. "Software Project Management Under Incomplete and Ambiguous Specifications." IEEE Transactions on Engineering Management 37 (1): 10-21.

Royce, Winston W. 1970. "Managing the Development of Large Software Systems."Proceedings of IEEE WESCON.August 1970.

Rumpe, Bernhard and Astrid Schroder. 2002. "Quantitative Survey on Extreme Programming Projects." Sardinia, Italy, International Conference on Extreme Programming.

Rungtusanatham, M., C. H. Ng, X. Zhao, and T. S. Lee. 2008. "Pooling Data Across Transparently Different Groups of Key Informants: Measurement Equivalence and Survey Research." Decision Sciences, 39 (1): 115-145.

Ruscher, Janet B. and Elliott D. Hammer. 1994. " "Revising Disrupted Impressions through Conversation." Journal of Personality and Social Psychology 66 (3): 530-541.

Sabherwal, Rajiv and Daniel Robey. 1995. "Reconciling Variance and Process Strategies for Studying Information System Development." Information Systems Research 6 (4): 303-327.

Saeed, Khawaja A., Manoj K. Malhotra, and Varun Grover. 2005. "Examining the Impact of Interorganizational Systems on Process Efficiency and Sourcing Leverage in buyer–supplier Dyads." Decision Sciences 36 (3): 365-396.

Saldana, Johnny. 2011. Fundamentals of Qualitative Research. Series in Understanding Qualitative Research., edited by Patricia Leavy. New York, NY: Oxford University Press.

Salo, Outi and Pekka Abrahamsson. 2008. "Agile Methods in European Embedded Software Development Organisations: A Survey on the Actual use and Usefulness of Extreme Programming and Scrum." IEEE Software 2 (1): 58-64.

Sauer, Chris and Blaize Horner Reich. 2009. "Rethinking IT Project Management: Evidence of a New Mindset and its Implications." International Journal of Project Management 27 (2): 182-193.

Sault, Robert J., Teuben, Peter J. and Wright, Mel C. "A Retrospective View of Miriad.", last modified December 28, 2006, accessed August 11, 2013, arXiv preprint astro-ph/0612759, http://arxiv.org/pdf/astro-ph/0612759.pdf.

Sawyer, Steve, Patricia J. Guinan, and Jay Cooprider. 2010. "Social Interactions of Information Systems Development Teams: A Performance Perspective." Information Systems Journal 20 (1): 81-107.

Scaife, M. and Y. Rogers. 1996. "External Cognition: How do Graphical Representations Work?" International Journal of Human-Computer Studies 45: 185-213.

Schraagen, Jan Maarten, Gary Klein, and Robert R. Hoffman. 2008. "The Macrocognition Framework of Naturalist Decision Making." In Naturalistic Decision Making and Macrocognition, edited by J. M. Schraagen, L. G. Militello, T. Ormerod and R. Lipshitz, 3-25. Hampshire, England: Ashgate.

Seddon, P. and S. K. Yip. 1992. "An Empirical Evaluation of User Information Satisfaction (UIS) Measures for use with General Ledger Accounting Software." Journal of Information Systems 6 (1): 75-92.

Selic, Bran. 2009. "Agile Documentation, Anyone?" IEEE Software 26 (6): 11-12.

Sethi, Rajesh and Zafar Iqbal. 2008. "Stage-Gate Controls, Learning Failure, and Adverse Effect on Novel New Products." Journal of Marketing 72 (1): 118-134.

Shanks, G., E. Tansley, J. Nuredini, D. Tobin, and Weber R. 2008. "Representing Part-Whole Relations in Conceptual Modeling: An Empirical Evaluation." MIS Quarterly 32 (3): 553-573.

Shenhar, Aaron J. and Dov Dvir. 1996. "Toward a Topological Theory of Project Management." Research Policy 25: 607-632.

Shenhar, Aaron J., Dov Dvir, Ofer Levy, and Alan C. Maltz. 2001. "Project Success: A Multidimensional Strategic Concept, Long Range Planning." Long Range Planning 34 (6): 299-725.

Sherer, Susan and Steven Alter. 2004. "Information System Risks and Risk Factors: Are they mostly about Information Systems." Communications of the AIS 14 (2): 29-64.

Siemsen, Enno, Aleda Roth, and Pedro Oliveira. 2010. "Common Method Bias in Regression Models with Linear, Quadratic, and Interaction Effects." Organizational Research Methods 13 (3): 456-476.

Sloman, Martyn and Toni Borsattino. 2007. "Barriers to Progress in learning–a Global Convergence of Concern?" Development and Learning in Organizations 21 (2): 14-16.

Sniezek, Janet A. 1992. "Groups Under Uncertainty: An Examination of Confidence in Group Decision Making." Organizational Behavior and Human Decision Processes 52 (1): 124-155.

Soderholm, Anders. 2008. "Project Management of Unexpected Events." International Journal of Project Management 26 (1): 80-86.

Sole, T. D. and G. Bist. 1991. "The Design Document: Key to Managing Documentation Development." IEEE Transactions on Professional Communication 34 (2): 70-74.

Spencer, Carolyn R. and Trevor Hine. 2005. "Project Manager's Changing Cognitive Schemata and the Achievement of Group Goals." In The Story of Managing Projects: An Interdisciplinary Approach, edited by Elias G. Carayannis, Young Hoon Kwak and Frank T. Anbari, 134-151. Westport Connecticut: Praeger.

Staber, Udo and Jorg Sydow. 2002. "Organizational Adaptive Capacity: A Structurational Perspective." Journal of Mangement Inquiry 11 (4): 408-424.

Stacey, Ralph D. 1993. Strategic Management and Organizational Dynamics. London, UK: Pitman Publishing.

Stephens, Matt. "The Case Against Extreme Programming." Software Reality., last modified January 26, 2003, accessed 8/1, 2013, http://www.softwarereality.com/lifecycle/xp/case_against_xp.jsp.

Stettina, C. J., W. Heijstek, and T. E. Faegri. 2012. "Documentation Work in Agile Teams: The Role of Documentation Formalism in Achieving a Sustainable Practice."Agile Conference (AGILE), 2012.

Stewart, Katherine J. and Sanjay Gosain. 2006. "The Impact of Ideology on Effectiveness in Open Source Software Development Teams." MIS Quarterly 30 (2): 291-314.

Steyn, Herman. 2002. "Project Management Applications of the Theory of Constraints Beyond Critical Chain Scheduling." International Journal of Project Management 20 (1): 75-80.

Strauss, Anselm and Juliet Corbin. 1998. Basics of Qualitative Research Techniques and Procedures for Developing Grounded Theory. Second ed. Thousand Oaks, California: Sage Publications.

Strode, Diane E., Sid L. Huff, Beverley Hope, and Sebastian Link. 2012. "Coordination in Co-Located Agile Software Development Projects." Journal of Systems and Software 85: 1222-1238.

Sun, Ming and Xianhai Meng. 2009. "Taxonomy for Change Causes and Effects in Construction Projects." International Journal of Project Management 27: 560-572.

Susi, T. and T. Ziemke. 2001. "Social Cognition, Artifacts, and Stigmergy: A Comparative Analysis of Theoretical Frameworks for the Understanding of Artifact-Mediated Collaborative Activity." Cognitive Systems Research 2 (4): 273-290.

Takeuchi, Hirotaka and Ikujiro Nonaka. 1986. "The New New Product Development Game." Harvard Business Review 64 (1): 137-146.

Tang, Anthony. 2011. "Software Designers, are You Biased?"Proceedings of the 6th International Workshop on Sharing and Reusing Architectural Knowledge.ACM, .

Tang, Antony, Aldeida Aleti, Janet Burge, and Hans van Vliet. 2010. "What Makes Software Design Effective?" Design Studies 31 (6): 614-640.

Tannenbaum, Scott I., John E. Mathieu, Eduardo Salas, and Debra Cohen. 2012. "Teams are Changing: Are Research and Practice Evolving Fast enough?" Industrial and Organizational Psychology 5 (1): 2-24.

Tatikonda, Mohan V. and Stephen R. Rosenthal. 2000. "Technology Novelty, Project Complexity and Product Development Project Execution Success: A Deeper Look at Task Uncertainty in Product Innovation." IEEE Transactions on Engineering Management 47 (1): 74-87.

Taylor, Alva and Constance E. Helfat. 2009. "Organizational Linkages for Surviving Technological Change: Complementary Assets, Middle Management, and Ambidexterity." Organization Science 20 (4): 718-739.

Tesch, Debbie, Timothy J. Kloppenborg, and Mark N. Frolick. 2007. "IT Project Risk Factors: The Project Management Professionals Perspective." Journal of Computer Information Systems 47 (4): 61-69.

Thamhain, Hans. 2013. "Managing Risks in Complex Projects." Project Management Journal 44 (2): 20-35.

Thomas, Gary. 2010. "Doing Case Study: Abduction no Induction, Pronesis Not Theory." Qualitative Inquiry 16 (7): 575-582.

Tiwana, Amrit and Ephraim R. McLean. 2005. "Expertise Integration and Creativity in Information Systems Development." Journal of Management Information Systems 22 (1): 13-43.

Tompkins, Phillip K. and George Cheney. 1985. "Communication and Unobtrusive Control in Contemporary Organizations." Organizational Communication: Traditional Themes and New Directions 13: 179-210.

Turner, J. Rodney. 2006. "Towards a Theory of Project Management: The Nature of the Project." International Journal of Project Management 24: 1-3.

Tushman, Michael L. and Thomas J. Scanlan. 1981. "Boundary Spanning Individuals: Their Role in Information Transfer and their Antecedents." Academy of Management Journal 24 (2): 289-305.

Tversky, Amos and Daniel Kahneman. 1974. "Judgment Under Uncertainty: Heuristics and Biases." Science 185 (4157): 1124-1131.

Urbach, Nils and Frederik Ahlemann. 2010. "Structural Equation Modeling in Information Systems Research using Partial Least Squares." Journal of Information Technology Theory and Application 11 (2): 5-40.

van der Vegt, Gerben S., Stuart Bunderson, and Ben Kuipers. 2010. "Why Turnover Matters in Self-Managing Work Teams: Learning, Social Integration, and Task Flexibility." Journal of Management Studies 36 (5): 1168-1191.

van Mierlo, Heleen, Jeroen K. Vermunt, and Christel G. Rutte. 2009. "Composing Group-Level Constructs from Individual-Level Survey Data." Organizational Research Methods 12 (2): 368-392.

Vessey, Iris. 1991. "Cognitive Fit: A Theory-Based Analysis of the Graphs Versus Tables Literature." Decision Sciences 22 (2): 219-240.

Vidgen, Richard and Xiaofeng Wang. 2009. "Coevolving Systems and the Organization of Agile Software Development." Information Systems Research 20 (3): 355-376.

Vinekar, Vishnu, Craig W. Slinkman, and Sridhar Nerur. 2006. "Can Agile and Traditional Systems Development Approaches Coexist? an Ambidextrous View." Information Systems Management 23 (3): 31-42.

Visconti, Marcello and Curtis Cook. 1993. "Software System Documentation Process Maturity Model."Proceedings of the 1993 ACM Conference on Computer Science.

Wageman, Ruth, Heidi Gardner, and Mark Mortensen. 2012. "The Changing Ecology of Teams: New Directions for Teams Reasearch." Journal of Organizational Behavior 33 (3): 301-315.

Wagner, Stephan M., Christian Rau, and Eckhard Lindemann. 2010. "Multiple Informant Methodology: A Critical Review and Recommendations." Sociological Methods & Research 38 (4): 582-618.

Wakefield, Robin L., Dorothy E. Leidner, and Gary Garrison. 2008. "Research Note—A Model of Conflict, Leadership, and Performance in Virtual Teams." Information Systems Research 19 (4): 434-455.

Wallace, Linda, Mark Keil, and Arun Rai. 2004a. "Understanding Software Project Risk: A Cluster Analysis." Information & Management 42 (1): 115-125.

Wallace, Linda, Mark Keil, and Arun Rai. 2004b. "How Software Project Risk Affects Project Performance: An Investigation of the Dimensions of Risk and an Exploratory Model." Decision Sciences 35 (2): 289-321.

Wang, Eric TG, Pei-Hung Ju, James J. Jiang, and Gary Klein. 2008. " the Effects of Change Control and Management Review on Software Flexibility and Project Performance " Information & Management 45 (7): 438-443.

Wang, X. and R. Vidgen. 2007. "Order and Chaos in Software Development: A Comparison of Two Software Development Teams in a Major Company." St. Gallen, Switzerland, Proceedings of the 15th ECIS.June 7-9.

Warner, Normal W. and Michael Letsky. 2008. "Empirical Model of Team Collaboration: Focus on Macrocognition (Chapter 2)." In Macrocognition in Teams: Theories and Methodologies, edited by Michael P. Letsky, Norman W. Warner, Stephen M. Fiore and C. A. P. Smith, 15-33. Hampshire, England: Ashgate.

Wateridge, John. 1998. "How can IS/IT Projects be Measured for Success?" International Journal of Project Management 16 (1): 59-63.

Watson-Manheim, Mary Beth, Katherine M. Chudoba, and Kevin Crowston. 2012. "Perceived Discontinuities and Constructed Continuities in Virtual Work." Information Systems Journal 22: 29-52.

Wearne, Stephen. 2006. "Managing Unexpected Urgent Projects." Project Management Quarterly 37 (5): 97-102.

Weick, Karl E. 1976. "Educational Organizations as Loosely Coupled Systems." Administrative Science Quarterly 21: 1-19.

West, David, Mike Gilpin, Tom Grant, and Alissa Anderson. 2011. Water-Scrum-Fall is the Reality of Agile for most Organizations Today. Cambridge, Mass: Forrester Research, Inc.

Whittaker, James A. and Jeffrey M. Voas. 2002. "50 Years of Software: Key Principles for Quality." IT Professional 4 (6): 28-35.

Whitworth, Elizabeth. 2008. "Experience Report: The Social Nature of Agile Teams."Agile Conference 2008.IEEE, .

Williams, Larry J., Nathan Hartman, and Flavia Cavazotte. 2010. "Method Variance and Marker Variables: A Review and Comprehensive CFA Marker Technique." Organizational Research Methods 13 (3): 477-514.

Willmott, Hugh. 1993. "Strength is Ignorance; Slavery is Freedom: Managing Culture in Modern Organizations." Journal of Management Studies 30 (4): 515-552.

Wilson, Bruce L. and H. Dickson Corbett. 1983. "Organization and Change: The Effects of School Linkages on the Quantity of Implementation." Educational Administration Quarterly 19 (4): 85-104.

Wittenbaum, G. M., S. I. Vaughn, and G. Stasser. 1998. "Coordination in Task Performing Groups." In Theory and Research on Small Groups, edited by R. Scott Tindale, Linda Heath, John Edwards, Emil J. Posavac, Fred B.

Bryant, Yolanda Suarez-Balcazar, Eaaron Henderson-King and Judith Myers, 177-204. New York, NY: Plenum Press.

Wixom, Barbara H. and Hugh J. Watson. 2001. "An Empirical Investigation of the Factors Affecting Data Warehousing Success." MIS Quarterly 25 (1): 17-32.

Wu, Liang-Chuan and Chorng-Shyong Ong. 2008. "Management of Information Technology Investment: A Framework Based on a Real Options and Mean-Variance Theory Perspective." Technovation 28: 122-134.

Wynn, Donald and Clay K. Williams. 2012. "Principles for Conducting Critical Realist Case Study Research in Information Systems." MIS Quarterly 36 (3): 787-810.

Yang, Seung-Bum and Mary E. Guy. 2011. "The Effectiveness of Self-Managed Work Teams in Government Organizations." Journal of Business and Psychology 26 (4): 531-541.

Yeo, K. T. 1995. "Strategy for Risk Management through Problem Framing in Technology Acquisition." International Journal of Project Management 13 (4): 219-224.

Yin, Robert K. 2009. Case Study Research: Design and Methods. Applied Social Research Methods Series, Volume 5., edited by Leonard Bickman, Debra J. Rog. Fourth ed. Thousand Oaks, California: SAge Inc.

Yoon, Youngohc, Tor Guimaraes, and Quinton O'Neal. 1995. "Exploring the Factors Associated with Expert Systems Success." MIS Quarterly 19 (1): 83-106.

Zellmer-Bruhn, Mary E. 2003. "Interruptive Events and Team Knowledge Acquisition." Management Science 49 (4): 514-528.

Zhang, Jiajie and Vimla L. Patel. 2006. "Distributed Cognition, Representation, and Affordance." Pragmatics & Cognition 14 (2): 333-341.

Zhu, G., J. F. Bard, and G. Yu. 2005. "Disruption Management for Resource-Constrained Project Scheduling." Journal of Operational Research Society 56: 365-381.

Zmud, Robert W. 1980. "Management of Large Software Development Efforts." MIS Quarterly 4 (2): 45-55.

Zviran, Moshe and Zippy Erlich. 2003. "Measuring IS User Satisfaction: Review and Implications." Communications of the AIS 12 (12): 81-103.

Biographical Information

Kurt Schmitz is a thirty year IT professional, having served in multiple roles including software developer, systems analyst, project manager, and IT manager. In his current capacity as IT Head for R&D Surgical Franchises at Alcon Laboratories he has coordination and application service delivery responsibility across multiple R&D facilities in Europe and North America. Kurt received his bachelor's degree in business administration from Samford University in Birmingham AL in 1984 while working for a small phototypesetting company as a computer operator and junior programmer. He relocated to upstate New York and completed a Master of Science in Computer Science at Rensselaer Polytechnic Institute in 1990 while employed with General Electric's Factory Automation group. In the 1990s Kurt managed networking, application integration and electronic commerce technology groups at Nortel before joining Alcon Laboratories in 2002. Kurt returned to academia in a part-time capacity in 2008 to pursue a PhD in Information Systems at the University of Texas at Arlington.