2010

BEYOND MODES: A NEW TYPOLOGY OF ISD CONTROL

W. Alec Cram
Queen's University, wcram@bentley.edu

M. Kathryn Brohman
Queen's University, kbrohman@business.queensu.ca

BEYOND MODES: A NEW TYPOLOGY OF ISD CONTROL

Completed Research Paper

W. Alec Cram
Queen's University
Kingston, Ontario, Canada
wcram@business.queensu.ca

M. Kathryn Brohman
Queen's University
Kingston, Ontario, Canada
kbrohman@business.queensu.ca

Abstract

This study focuses on the role of information systems development (ISD) control as a means to better understand the underlying elements of ISD methodologies. Based on Ackoff's general systems theory, we propose a new typology of ISD control that moves beyond the de facto standard in ISD research: control modes. Through twenty-six interviews at four organizations, we find that our typology aids in distinguishing between control dimensions within structured and flexible development to a greater extent than control modes would allow. Our paper also discusses possible reasons why the benefits of agile development may become diluted in organizations where both flexible and structured methods are employed. This work contributes to the advancement of ISD control theory and provides useful insights for practitioners responsible for the governance of ISD projects.

Keywords: IS control, Agile software development, Systems analysis and design

Introduction

Information systems researchers and practitioners claim that IT departments are diluting the benefits of agile software development methods by infusing them with traditional, structured techniques in order to overcome organizational constraints and executive resistance to change (Karlstroem & Runeson 2005; Manhart & Schneider 2004). Such approaches are becoming more widespread in practice as development teams "are puzzling out the mix of methodologies and combining them to fit within their organizational realities, blending agile and non-agile techniques and practices to create a hybrid methodology that fits larger organizations" (West et al. 2010). For example, Daimler-Chrysler did not adopt a full agile approach but added specific agile practices in an attempt to 'break the ice' and foster organizational acceptance of agile elements (Manhart & Schneider 2004). Despite management resistance to agile methods in traditional, stage-gate project management organizations, customized agile methods are increasingly being employed in such environments (Karlstroem & Runeson 2005). As developers and IT professionals rapidly move to adopt agile information systems development (ISD) approaches - 45% of 13,000 organizations recently surveyed said they use agile methods (Taft 2010) – we expect that more IT organizations will face constraints, resistance, and pressure to follow the trend of hybrid systems development methodologies. To date, little research has been done to determine the causes or the effects of hybrid development, particularly in relation to the dilution of agile practices.

One plausible explanation for why agile practices are being diluted is a misalignment between ISD project and organizational controls (Taft 2010). A common mechanism used for ISD project control is through the use of ISD approaches, which refer to a set of related goals, guiding principles, and fundamental concepts that drive interpretations and actions in systems development processes (Iivari et al. 2000/2001). Organizational control consists of the set of structures, processes, policies, and relationships that affect the way people direct and administer a corporation; these mechanisms include organizational structure, standardized processes, and performance management tools (e.g. balanced scorecard). Information systems literature has studied the relationship between performance and alignment of business to IT from a governance and control perspective (Broadbent & Weill 1993; Weill & Ross 2004). Consistent with this research, we have designed a research program to study how organizations effectively control systems development projects when project and organizational controls are misaligned, as well as how contradictory control elements in hybrid development projects lead to performance impacts.

As a preliminary step in our research program, the purpose of this paper is to develop a theoretically grounded typology of ISD project control by clearly differentiating the mechanisms used to control information systems development. Effective control over ISD is widely viewed as an important contributor to achieving project and organizational objectives (Dibbern et al. 2008; Henderson & Lee 1992; Sia & Neo 1997). Past ISD research on control predominantly focuses on how types of control are determined by organizational and behavioral circumstances (Choudhury & Sabherwal 2003; Kirsch 1996). Standardization and use of control mechanisms have been examined in prior research (Kirsch 1997; Nidumolu & Subramani 2003); however, there is a general lack of theory pertaining to the role of control in end-to-end systems development methodologies (Harris et al. 2009). Though a need exists for advancement in control conceptualizations in order to extend knowledge in the field (Kirsch 2004), the role of control in ISD remains dominated by Kirsch's (1996, 1997) portfolio of control modes (see Table 1).

A key mechanism of control in ISD projects is the development methodology, which represents "a cornerstone of disciplinary practice, a means of controlling processes and results" (Orlikowski 1991, p. 16). Traditional software development methodologies (e.g. waterfall) are very structured and sequential in nature; the early dominance of these methods in practice is attributed to advocates that argued for development to be completed via discrete phases, whereby requirements analysis, design, testing, and implementation are individually completed in a pre-defined sequence (Boehm 1976; Royce 1970). Using these plan-driven, structured methods, customers are primarily involved in the requirements gathering, but have minimal participation in subsequent phases. A more recent movement to encourage more flexibility to respond to changing customer needs in software projects has resulted in a number of more agile development methodologies including XP (eXtreme Programming), Scrum, DSDM (Dynamic Systems Development Method), and FDD (Feature-Driven Development). Agile development is based around the concept of collaborative, self-organizing teams that iteratively deliver working software (Beck et al. 2001; Martin 1991). Agile projects take a minimalist approach to design and documentation, relying instead on adaptability, flexibility, and responsiveness (Fitzgerald et al. 2006; Lee & Xia 2010; Nerur & Balijepally 2007). Project team

members tend to have only loosely defined responsibilities and customers are encouraged to be actively involved in the development process (Nerur et al. 2005). Agile development methods allow software teams to increase flexibility by allowing them to embrace and respond to changing environments (Beck & Andres 2004; Coad et al. 1999).

The trade-off between control and flexibility as a central dynamic in today's software development approaches is particularly pertinent to the definition of ISD control. To date, there is no accepted view about the relationship between control and flexibility in agile literature. Some argue that the fundamental agile design views contradict control (Augustine et al. 2005) and others find that less formal control approaches are not a significant predictor of agile development success (Misra et al. 2009). Two articles recently published in the Information Systems Research (ISR) special issue on agile methods apply more theoretical and rigorous methods to examine control of flexible software development practices. Both papers define control according to Kirsch's (1996) portfolio of control modes; however, their results are contradictory, as each defines a different control mode to be appropriate for agile environments. Maruping et al. (2009) conclude that the relationship between agile methodology use and software project quality is moderated by outcome control (see Table 1), while Harris et al. (2009) argue both formal control modes (output and behavioral) are insufficient and introduce a new form of control (emergent outcome control) to explain agile environments. One plausible explanation for such conflicting results in the literature is that control modes do not sufficiently differentiate structured and flexible methods of systems development. As such, this paper uses Ackoff's general systems theory (1974, 1981) to develop a new typology of ISD control and validates the typology using 26 interviews with experienced systems developers in four organizations.

The remainder of this paper is structured as follows. The next section will outline the management control literature and its relationship to ISD. This will be followed by a proposed new typology of ISD control, based on Ackoff's systems theory. Next, the research design, data collection, and analysis methods will be outlined, followed by a presentation of the results and a discussion of the findings. Finally, the paper will conclude with details of the study's limitations and implications for research and practice.

Controlling Systems Development

Control broadly refers to a process in which a person or group intentionally affects the behavior of another person or group, often with the objective of achieving goals (Davis 1940; Flamholtz et al. 1985; Tannenbaum 1962). Researchers from computer science, software engineering, and management information systems (MIS) have examined the relationship between control in ISD at varying degrees; in fact, the entire ISD methodology movement in the 1970s and 1980s was predicated on the importance of structure to encourage the use of different methodologies to guide development (Boehm 1976). Markus and Bjorn-Anderson (1987) examine the role of IS professionals exercising technical, structural, conceptual, and symbolic power. They argue that development methods may aid in the exercise of power by guiding the questions that developers ask and thus influencing the design features.

In MIS, ISD methodologies represent an organized grouping of a development approach, model, method(s), and techniques that are supported by material resources (Hirschheim, Klein & Lyytinen 1995; Huisman & Iivari 2006). The use of ISD methodologies and these underlying components are believed to comprise inherent control mechanisms that are used to enable consistency and quality (e.g. minimized software defects) in systems development.

For example, Westrup (1993) argues that management control can be exerted via development methodology use by demanding specified outcomes at particular project stages through development tools such as CASE and extensive staff training. Orlikowski (1991) argues that methodologies are a form of cultural control over developers as they force an "internalization of...assumptions, values, and interests" (p. 17). Later, Kirsch (1997) indicates that organizations operating in dynamic, changing environments may actually change control approaches throughout the lifecycle of an IS project as priorities change; the result is a portfolio of control modes for information systems projects. Developed from managerial control literature (Ouchi 1978, 1979), she defined four control models in ISD (Kirsch 1996, 1997; Kirsch et al. 2002). This portfolio of modes, defined in Table 1, is the common lens used by IS researchers to study ISD control to date. Although Kirsch references procedures such as development standards and performance goals that would commonly be part of a development methodology, ISD methodologies are not specifically addressed.

Several other researchers have extended the use of control theory into software development environments (Choudhury & Sabherwal 2003; Henderson & Lee 1992; Nidumolu & Subramani 2003) and more recently researchers have examined the control of flexible software development (Harris et al. 2009; Maruping et al. 2009; McHugh et al. 2008). The importance of flexibility in software development processes is supported by a substantial body of research (e.g. Byrd & Turner 2000; Duncan 1995; Gefen & Keil 1998) and reports a significant relationship between flexibility and software development performance (Lee & Xia 2005; MacCormack et al. 2001); however, the relationship between control and flexibility remains uncertain.

Table 1. Modes of Control

Formal	Behavioral control	Exhibited when a supervisor 'checks' on a subordinate. It is subtle, flexible, captures a range of activities, though assessments are subjective and difficult to communicate through organizational levels (Ouchi 1978)
	Outcome control	Exhibited by monitoring an employee's level of output. It is quantifiable, comparable across levels and functions, and captures performance levels specific activities (Ouchi 1978)
Informal	Clan control	Uses selection processes and social mechanisms to control the behavior of individuals (Ouchi 1979). For example, if employee mistakes on a project are publicly criticized by other team members, individuals may become more vigilant in their activities so as not to attract negative attention from their peers.
	Self control	Individuals control their own actions, outside the direction of management (Kirsch 1996). For example, if an employee takes initiative to resolve an issue without being asked to do so by their supervisor.

The fundamental challenge in using modes to examine control in flexible environments is that the scope is too general; in fact, both Maruping et al. (2009) and Harris et al. (2009) incorporate additional theory to examine control in agile environments. Maruping et al. (2009) base their theoretical argument on the importance of team member autonomy, specifically that decentralized decision-making authority is important when requirements are volatile (Nidumolu & Subramani 2003) and outcome control creates an environment for autonomy (Henderson & Lee 1992). They find that outcome control is more appropriate in agile environments as autonomous team members will have a heightened need to respond in changing environments. Harris et al. (2009) conceptualize the role of control in agile development from a product development lens and base their theoretical argument on the theory of dynamic capabilities; high change environments require the ability to reconfigure on-the-fly because change cannot be predicted (Eisenhardt & Martin 2000; Teece et al. 1997). They argue output control is not appropriate in agile environments as outcomes in agile methods are not explicit; rather, they are developed from an iterative development process. Based on these results, we conclude that a typology of ISD control needs to go beyond control modes to effectively define control mechanisms that govern both traditional and flexible development environments.

ISD Control: A New Typology

Doty and Glick (1994) define typologies as a conceptually derived, interrelated set of ideal types, which represent configurations of multiple constructs. Our ISD control typology differentiates control on two theoretical dimensions: 1) the inherent problem solving method of the ISD approach, and 2) the dimensions of control in a problem solving environment. On this basis, we develop and evaluate three research propositions, which are described below.

Inherent Problem Solving Method

Consistent with Ackoff's general systems theory (1974, 1981), our typology posits that variation in ISD control can be explained by the inherent problem solving method of the development approach. The motivation to adopt this theoretical dimension was inspired by a similar movement from structured to flexible methods in project management; flexible methods are defined as integrative project management practices. Shenhar and Dvir (2007) argue that the discipline of project management needs to undergo a fundamental shift in practice and philosophy in order to move away from reductionist project management approaches. Information systems researchers also claim that fundamental design views of integrative project management are contradictory to traditional methods, which are defined by controlling and directive practices appropriate for stable environments, whereas agile methods are

grounded in collaborative and communicative practices appropriate for turbulent environments (Nerur & Balijepally 2007). In software engineering, Adaptive Software Development, Crystal, Dynamic Systems Development Method, Feature-Driven Development, Internet-Speed Development, and Scrum support integrated project management principles (Augustine et al. 2005).

We deem this approach appropriate as ISD methodologies are essentially a set of guidelines, activities, and tools that vary according to some "systematic way of conducting at least one phase of the systems development lifecycle" (Huisman & Iivari 2006, p. 32). To clarify, consider the systematic ways that a project team may choose to define the requirements of a new system. One approach would be to define the high level scope and objectives of the system and break down the work to create a requirements document used to guide development. Ackoff (1974) defines this structured problem solving approach as a reductionistic method of analysis. In ISD, structured approaches see the organization (or the customer) as a 'tool' and attempt to understand their problems (or needs) by breaking them into parts and following a step-by-step process to create a detailed analysis and design of the software to be developed (e.g. waterfall). Early research in computer science and software engineering literature argues that managers break down programming initiatives into tasks as a means to more easily control developers. By utilizing the organizational hierarchy, Kraft (1977) suggests that the fragmentation of programming tasks may lead to increased efficiency and productivity, but also enables managers to remain in charge of development initiatives and staff advancement.

An alternative, systematic approach for a project team to establish systems requirements is by defining a general direction for the project and continuously communicating with users to capture requirements and priorities on an ongoing basis, while concurrently developing the software in short, iterative cycles. This agile approach follows expansionistic methods of analysis where the organization (and other stakeholders) work to develop a holistic view of the problem and use continuous planning approaches to encourage learning in rapidly changing environments (Ackoff 1974). In ISD, these methods have been defined as agile as well as interactionist, speech act-based, and soft systems (Iivari et al. 2000/2001).

Categories and Dimensions of Control

In contrast with ISD control research, the concept of control within the management discipline has a long and varied history (Giglioni & Bedeian 1974). Beginning in the early 1900s and continuing through the present day, publications have varied in their interpretations of the concept of control. After a thorough review of managerial control literature, including both conceptual and empirical work, we identified fourteen distinct dimensions of control (see Table 2).

Table 2. Emergent Categories and Dimensions of Control

Control Categories	Control Objectives						Control Practices					Control Effects		
	Speed	**Budget**	**Functionality**	**Regulatory Compliance**	**Product Quality**	**Risk Reduction**	**Active Monitoring**	**Enforcement**	**Standardization & Measurement**	**Information & Interaction**	**Remediation & Correction**	**Authority**	**Dependence**	**Satisfaction**
Objectives	8	8	5	8	9	8	0	0	0	0	0	0	0	0
Practices	0	1	0	0	0	1	9	9	9	9	9	0	1	0
Effects	1	0	4	1	0	0	0	0	0	0	0	9	8	9

In an effort to categorize these dimensions into higher level categories, we used Gregory's (2007) work on control systems and performance management. Although this research was completed from an operations management perspective, it defines control categories from a general problem solving context using Ackoff's general systems theory (1974, 1981), so the categories of control were deemed appropriate in a systems development context. We asked 9 IS professionals (5 academics and 4 practitioners) with extensive systems development experience to sort the dimensions into three control categories: objectives, practices, and effects; Moore and Benbasat's (1991)

guidelines for confirmatory card sorting were used to structure this exercise. Table 2 shows the results of the confirmatory card sort. All items but one clearly converged on a single category of control (the product functionality control item was removed from the study). The following sections outline the three emergent categories and their underlying dimensions.

Control Objectives

Control objectives are described by five items, which each indicate a specific end result defined by ISD project stakeholders. This type of control is deemed appropriate when desired outcomes can be established a priori and actual accomplishments can be measured against the targets. By applying general systems theory, control objectives are separated into two ideal types: targets and tolerances. Targets represent a reductionistic approach, whereby control is used as a means of achieving targets related to time deadlines, budget limitations, and compliance regulations (Davis 1951; Koontz 1958; Koontz 1959). Tolerances represent an expansionistic approach, whereby control is used to reach an objective that exists within a desired range of acceptability.

The disadvantage of achieving control objectives through target setting is that managers tend to neglect important emergent aspects of the systems behavior (Caulkin 2002). For example, if an ISD project is defined by a specific budget objective, efforts to control behavior toward this objective may inhibit team member ability to see emergent aspects related to the required functionality of the system. General systems theory supports the claim that target-based control objectives are less problematic in simple projects, but claims that more expansionistic approaches (defined as tolerances in our typology) are required to effectively control complex systems (Ackoff 1974). As flexible development methods claim to be more appropriate for managing changing requirements, budgets, and schedules (Harris et al. 2009), we propose that tolerance-based control objectives will be more prominent in flexible, or agile, development environments. We define tolerance-based objectives as quality control and risk reduction. These concepts have a long tradition in the management literature as an objective of control. For example, Cross (1928) cites the avoidance of fraudulent practices as an objective of control and Fayol (1949) suggests that "a good system of control provides against undesirable surprises, capable of degenerating into catastrophes" (p. 109).

> ***Proposition 1****: ISD control objectives will vary between structured and flexible ISD approaches: target-based control objectives will be more prominent in structured ISD approaches and tolerance-based control objectives will be more prominent in flexible ISD approaches.*

Control Practices

Control practices represent specific techniques used to enable the ISD process to achieve a control objective. The distinction between structured and facilitative control practices in our typology is consistent with Miller and Skidmore's (2004) distinction between hyper-organization and disorganization. They define hyper-organization as 'the restless pursuit of greater efficiency and the most rational way to pursue objectives', whereas disorganization 'reflects workers' desire to associate themselves with organizations which give them more autonomy and the facility to develop their own skills and ideas' (p. 16). In hyper-organizations, reductionistic practices are defined by activities focusing on measurement and evaluation (Eisenhardt 1985). Structured practices of control have a long history in management control literature (Cross 1928; Diemer 1915; Emerson 1919; Weber 1947) and remain common today as Project Management Offices (PMOs) attempt to define standard project management tools and techniques to guide project teams toward these set guidelines and regulations. In structured ISD approaches, project team members are assigned defined job roles and responsibilities (e.g. programmer, tester) and an importance is placed on the creation and maintenance of documents such as project plans, technical documents, and requirements specifications (Nerur et al. 2005). Fixed deadlines and budgets are established early in waterfall development and are closely scrutinized through detailed project budgeting, tracking, and reporting. We identified two structured control practices in the managerial literature: standardizing expectations and measuring actual results (Koontz 1959) and enforcing newly implemented rules or procedures (Davis 1940; Urwick 1944; Weber 1947).

In contrast, expansionistic practices of control enable the ISD process to work toward control objectives by effectively facilitating the dissemination of information and interaction between individuals and groups in an optimal manner (Eisenhardt 1985; Ouchi 1978; Urwick 1944), regularly remediating and correcting deficiencies (Brech 1965; Diemer 1915; Reeves & Woodward 1970), and actively monitoring of information, rules, activities, and people through ongoing, iterative processes (Child 1973; Ouchi 1978; Weber 1947). We define these as facilitative practices of control and propose that they are more common when attempting to achieve control in complex, changing environments:

> ***Proposition 2:*** *ISD control practices will vary between structured and flexible ISD approaches: structured practices will be more prominent for structured ISD approaches and facilitative control practices will be more prominent in flexible ISD approaches.*

Control Effects

Effects of control are the consequences that occur as a result of control objectives and practices. We uncovered three control effect dimensions in the managerial control literature and categorized each as either reductionistic or expansionistic. First, Weber (1947) and Tannenbaum (1962) perceive control as a mechanism that can limit discretion, restrict rights, and establish dominance over others. The effect of control is also defined by increased bureaucracy and dependence on static processes (Child 1973) and the enforcement of newly implemented rules or procedures (Davis, 1940; Urwick, 1944; Weber 1947). These effects all relate to the minimization of controlee dissent and the creation of dependence on strict orders, processes, and authority.

Expansionistic control objectives and practices encourage timely, accurate, and relevant information flows and remediation to encourage performance toward quality expectations and risk reduction. We argue that these practices give software development teams the autonomy to determine the best way to deploy resources in an effort to meet project goals. According to the managerial control literature, enabling individuals to gain insight into past, current, and future activities can result in a greater feeling of fulfillment and achievement (Ouchi 1978; Reeves & Woodward 1970). Control literature also claims that empowering employees to accept greater responsibility and exercise self-control and self-direction presents an opportunity for greater productivity and satisfaction (Kirsch 1996; McGregor 1960. Grounded in previous research that autonomy becomes even more consequential when software development teams need to respond to requirements changes (Gerwin & Moffat 1997; Henderson & Lee 1992; Maruping et al. 2009), we propose the following:

> ***Proposition 3:*** *ISD control effects will vary between structured and flexible ISD approaches: dependency control effects will be more prominent for structured ISD approaches and autonomy control effects will be more prominent in flexible ISD approaches.*

Based on the problem solving methods, control categories, and control dimensions discussed above, the proposed typology of ISD control is outlined in Figure 1.

Figure 1. Typology of ISD Control

Research Methodology

In order to provide useful insights into control phenomena, researchers have called for an increased focus on fieldwork and observation of organizational practices (Merchant 1988). In response, a positivist case study approach is employed in this study, with the purpose of building theory pertaining to ISD control.

Two development methodologies, agile and waterfall, are examined at four organizations. These methodologies were chosen as proxies for the plan-driven, structured methodologies and flexible, adaptive methodologies outlined in the propositions. Both waterfall and agile are widely used in practice (Conboy 2009; Nerur et al. 2005) and are perceived to be conflicting in their core strategy and philosophy, particularly in their approach to control. Whereas waterfall development is seen to integrate a high level of control through the use of structured processes and procedures, agile development is commonly perceived to be less controlled through the use of more flexible, adaptive activities (Dyba & Dingsoyr 2008).

Study Design

As organizations advocate a range of development processes and developers may have experience with different methodologies, a multiple case study design was considered the most effective research design in order to compare control phenomena across different systems development methodologies. A multiple-case study approach allows for cross-case analysis and the extension of theory (Benbasat et al. 1987), as well as providing a foundation for generalization (Yin 2009).

The unit of analysis for this study is at the systems development methodology level. Because control phenomena exist at the organizational, project, and individual level (Davis 1951; Flamholtz et al. 1985; Ouchi 1978), focusing exclusively on one level may exclude areas of potentially rich insights at other levels. Due to the exploratory nature of this research, a more encompassing view is desired; therefore, analysis is focused on development methodologies and their relationship with control across multiple levels.

Four organizations were selected to participate in this research. The first organization, referred to as UniCan, was selected as a pilot study, primarily on the basis of access and geographical proximity. Pilot studies are only occasionally used in case study research (Dube & Pare 2003); however, their use can contribute to a refined data collection process, contextual clarification, and field inquiry logistics streamlining (Yin 2009). The data collected from UniCan was used for these purposes, but the data itself was excluded from the data analysis process. UniCan is a mid-sized Canadian University with approximately 18,000 enrolled undergraduate and graduate students. Interviewees were involved in the systems development activities at the University, working on a variety of web and application development activities related to areas such as student registration and awards. UniCan utilizes a hybrid systems development methodology, including techniques characterizing both agile and waterfall methodologies.

The three remaining organizations each utilize different systems development methodologies. LargeMan is a multinational manufacturing conglomerate based in the United States. The company has offices in 32 countries around the world and over 75,000 employees. The location used for this study is in Ontario, where the local development team is responsible for the development of an inventory tracking application for the organization's products. LargeMan uses a waterfall systems development methodology, characterized by structured, sequential development cycles, detailed project budgeting and tracking, and formal approval of phase deliverables.

InfoOrg is a large information and technology company with approximately 50,000 employees. The location used for this study is in British Columbia, where a customer-facing finance application is developed and supported by a local project team. Although the core application utilizes a primarily waterfall methodology, smaller supplementary development projects have adopted an agile methodology, integrating daily 'stand-up' meetings, frequent delivery of working code, and short, iterative development cycles.

The final case study organization is a software development firm referred to as OntCode that develops software for financial services, retail, and manufacturing customers. OntCode uses an agile development methodology, characterized by iterative development cycles, collective code ownership, and pair programming.

Data Collection

The data were collected during June and July 2009 via 26 semi-structured interviews. Four interviews were conducted during the pilot study at UniCan, followed by eight interviews at InfoOrg, seven at LargeMan, and seven at OntCode. The interviews were conducted with participants in a range of systems development roles, including developers, systems analysts, quality assurance analysts, and development managers. By selecting participants from a range of roles within the development process, perspectives from both controllers and controlees were obtained. The nature of the interview data collection was both a retrospective account of participant experiences, as well as perspectives related to on-going projects. A formal interview protocol was utilized, but the interviewer remained open to probing into participant responses when they initiated a new area of inquiry. Subsequent to the pilot study, all interviews were digitally recorded and transcribed. In total, 189 pages of transcribed notes were produced from the interviews.

Prior to each interview, a questionnaire was completed by participants who indicated their level of experience using ten systems development techniques associated with waterfall development methodologies (e.g. specialized staff roles, working software at the end of a project, importance of technical documentation) and ten techniques associated with agile development methodologies (e.g. pair programming, story cards, collective code ownership). The techniques were not labeled by methodology and were randomly ordered in the questionnaire. A six point scale

was used, from 0 (No Experience) to 6 (Extensive Experience). A similar number of interviewees were selected from organizations representing a range of development methodologies: seven from a waterfall organization (LargeMan), seven from an agile organization (OntCode), and eight from a hybrid organization (InfoOrg). An additional four participants were interviewed from a hybrid organization (UniCan) for the pilot study. Experience scores ranged from 20 to 58 in agile experience, with an average of 37.5. For waterfall experience, scores ranged from 10 to 60, with an average of 38.7. Based on these results, the consolidated interviewee sample is concluded to have an approximately equal level of experience with both agile and waterfall methodologies.

In order to increase the reliability of data collection, the use of a case study protocol and a case study database were utilized during the study. Yin (2009) defines a case study protocol as the instrument, procedures and general rules used to guide the data collection process. The protocol was created prior to data collection and in line with Yin's (2009) suggestions, included sections on data collection procedures, an outline of the case study report, the case study questions, and evaluation guidelines. A case study database facilitates the separation of the collected data and the investigator's report, potentially allowing other investigators to independently review the collected evidence (Yin 2009). The use of QSR NVivo version 8.0 was used as a case study database in order to store investigator case study field notes, observations, interview transcripts, case study documents, and narratives.

Data Analysis

Defining an analysis strategy is an important step in identifying what data to analyze and why it should be analyzed in order to develop theory (Dube & Pare 2003; Yin 2009). First, the interviews were reviewed and coded by the first author; the 14 items of control (listed in Table 2) were used as coding categories (Miles & Huberman 1994). Seven interviews were reviewed in detail by the second author in order to validate the reliability of coding. All issues were discussed and coding revisions were made, where necessary. Similar techniques have been used in other studies such as Romm and Pilskin (1999) and Silva and Hirschheim (2007) as a means of facilitating researcher collaboration to increase data analysis reliability. Because the data collection and analysis stages overlapped, it allowed the investigators increased flexibility in data collection activities (Eisenhardt 1989). As a result, two new dimensions were added to the coding scheme: prioritization and eustress. Miles and Huberman (1994) suggest that this process of revision acts as a refinement of the original conceptual structure of the study and integrates empirically grounded insights generated from the field site. Participants identified prioritization as a control practice that enabled the ISD process to work toward control objectives. The use of prioritization was most common amongst agile developers as it applied to the activities of each iteration (called a sprint). The prioritization of tasks and decisions toward product objectives is evident in early control literature (Koontz 1958), but the role of prioritization as an ISD control practice has not been studied to date. The second item introduced by participants is eustress: positive stress resulting from an individual faced with a challenge (Selye 1964). Again, common amongst agile developers, eustress was mentioned as a consequence of flexible control objectives and practices by effectively motivating participants to achieve a high level of performance.

The final data analysis step was the in-depth examination of the coded data. Cross-case patterns are be used to identify within-group similarities and intergroup differences (Dube & Pare 2003). To achieve this objective, NVivo matrix queries were used to identify instances within the interview transcripts where both a systems development technique and a control category coding overlapped within a particular quote. By examining the matrix reports specific to individuals with experience using a particular methodology and development technique, insights were developed into the nature of control variance.

Results

Using the three categories of control in the proposed typology, we highlight the number of interviewee quotes referring to each of the sixteen control dimensions. Figure 2 illustrates the extent to which the data supports the research propositions. For comparative purposes, quotes referring to aspects of control modes are also presented. The frequency of individual quotes attributed to coding categories ranged from 3 (achievement of budget objective) to 30 (information and interaction). A total of 338 quotes were coded in the transcripts, averaging 17 quotes per control dimension and 13 quotes per interviewee. In the remainder of this section, a selection of interview quotes is used to illustrate the extent of the support for the three research propositions.

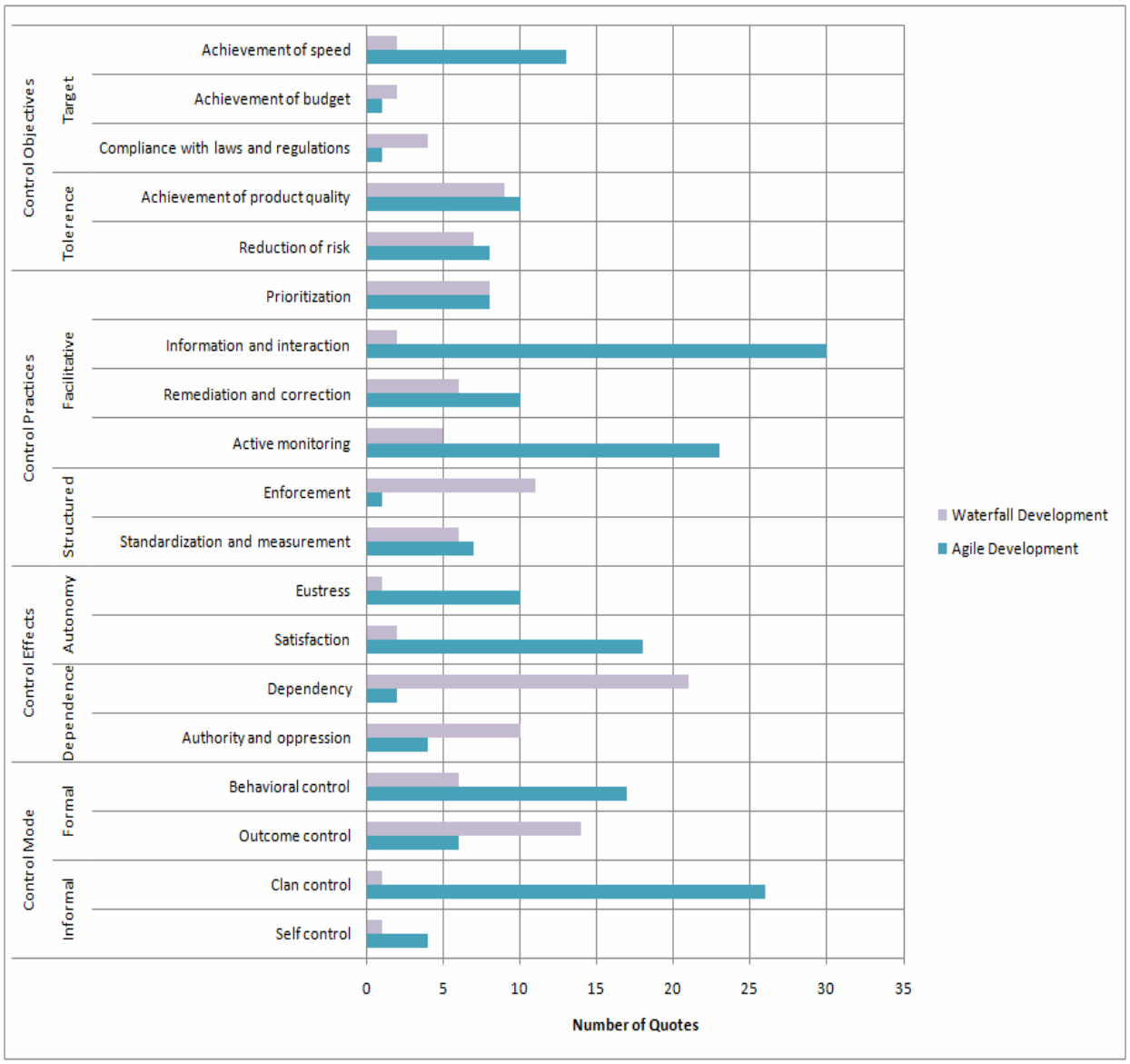

Figure 2. Control References

Control Objectives

Little evidence was found to support Proposition 1. Both budget and compliance were more prominent in waterfall but there was insufficient data to make strong conclusions for either control objective. Analysis of the content and frequency of quotations provided evidence that structured approaches established desired outcomes a priori, "*Generally most of [my experience is] a very strict waterfall type development cycle, in particular [at a former company] because I had to meet ISO standards and certifications so they had a very set process"* (Developer, OntCode) and actual accomplishments are measured against the a priori targets, "*So, if [we] are now accountable for…the release, [we] have got to figure out how to do it. Which can be, depending on your personality, really good, because then you can show off and say, 'I did this great work, and I hit the deadlines and the budget and all that stuff'"* (Developer, LargeMan).

The target-based objective with the largest variance was speed; however, results were opposite to the relationship hypothesized. After careful examination of quotations, we concluded that speed references by agile developers related more to speed generating the need for autonomy as opposed to speed as a target. *"With agile, it's quick, you are always on top of everything that's happening, you can reprioritize quickly; you can change the scope because you gained more understanding and you share it and then your tester is right with you..they get bits and pieces at a*

time" (Project Manager, InfoOrg). This interpretation suggests speed may also be a consequence of control – a control effect. One IS professional identified speed as a control effect in the card sort exercise. A follow-up conversation with this professional confirmed our interpretation that the speed of agile practices encourages autonomy that can present opportunities for greater productivity and satisfaction for developers.

Both quality and risk reduction were conceptualized as tolerances that correspond to an expansionist approach to control. Results show quality and risk were noted more frequently by agile participants than waterfall participants; however, the variance between the two groups was insufficient to conclude that tolerances are more common in flexible development environments.

Control Practices

Results provided stronger support for Proposition 2; four of the six control dimensions align with the proposed ISD typology categories (structured or facilitative), as predicted. The remaining two categories, prioritization and standardization/measurement were not found to vary significantly.

Enforcement of policies and procedures was clearly more prominent in the waterfall approach. Policies and procedures tended to the focus on the early definition of scope, requirements and deadlines, as well as the structured, sequential development phases and milestones. "*We know when we have to do the installation and then from there we will build a time line around that; around requirements gathering, around scoping, requirement gathering, development and a testing phase"* (Business Analyst, LargeMan).

Active monitoring, remediation and correction, and information and interaction were all more prominent in the agile approach. These practices were used in a holistic, expansionistic sense as the means used to ensure that project objectives were being met. Information and interaction was the most talked about control mechanism in our study,

> "*Things get done faster because we have a lot of face- to-face communication. We're in the same room; we don't send emails back and forth where they end up in some folder. I'm going to get up and I'm going to walk over to you...and ask: so what's going on with this, I'm running into this problem, can you come over here and give me a hand?"* (Developer, OntCode).
>
> "*You have to be able to work in the project room and because things change so quickly, if we can't communicate, even under stress, then it becomes an instant problem for the team"* (Developer, OntCode).

There were 30 information and interaction quotations made by agile developers compared to two by waterfall developers. Support for active monitoring in agile practices was also very strong, "*You can't necessarily corral human nature. But at least, in an agile project management world, you actually get, at least daily, an update from them...'what have you done, what are you doing today, what issues are you facing?' And hopefully you learn sooner when they're on these - I call them voyages of discovery - so that you can help them and get them to move forward"* (Developer, InfoOrg). Variance in remediation and correction was less evident, but comments by agile developers focused more on the real-time nature of this control practice, "*So, right away if you start something and the client says, 'oh wait a minute, I want to do that instead' or 'I don't really like the way it looks', right away you can fix it"* (Developer, OntCode). Remediation and correction in waterfall was more scheduled and structured in waterfall approaches: "*[Management] can look at a project and make the decision if they're going throw good money after bad. If there's something that is floundering, they've built these structures in that can look at it and say 'you need more work on this' or 'no, you can't even go any further'"* (Developer, LargeMan).

Control Effects

Control effects clearly varied between structured and flexible ISD approaches; results show strong evidence in support of Proposition 3. Waterfall developers talked about the effects of detailed project budgeting and tracking, intense documentation, and the structured, sequential nature of the development process. Authority was evident as developers described the consequence of structured practices, "*So these guys were handed documents with everything laid out in them. So there, it is, really they were treated almost like code monkeys. You could have handed this to… a group here or a group in India or a group in China; it didn't matter"* (Business Analyst, LargeMan). Structured approaches also prevented autonomy as developers are very dependent on static processes and are often forced to document and share information. Two developers at LargeMan described this best:

> *"I think a lot of my problems with the [development process are that] the analysts work as secretaries rather than actually using analytical thought. And it becomes a sort of customer-driven design where a customer comes up with a screen and then hands it to me through the analyst, which is why we end up re-architecting a lot of the designs because they are just really amateurish, they're not done by people who actually understand technology in any way"* (Developer, LargeMan).

Both satisfaction and eustress align with the autonomous control type and were talked about as consequences of key principles behind the agile manifesto (Beck et al. 2001). Satisfaction was most commonly talked about in relation to frequently delivering working code and the use of short, iterative development cycles (i.e. sprints).

> *"You actually feel like you're delivering something, you're actually incrementally improving your software, you can actually see that. And that's good because I think that gives people gives satisfaction in what they do…there's actually, there's progress, there's a feeling of achievement and I think people value that" (*Developer, InfoOrg).

> "*Clients like seeing new things like, 'wow, I'm getting a lot of stuff, you know weren't they just here last week showing me something new, and they're back again? Wow, these guys are really doing something for me"* (Developer, OntCode).

Eustress was also an effect of frequently delivery working code, "*I think it's good to have that tension that exists sometimes, you don't want it to be too large but primary focus for software development is getting something to your clients, right?... So, having that little bit of tension there I think is actually good because it gives people some focus"* (Developer, OntCode), as well as the need to maintain a constant pace to promote sustainable development:

> "*I would say it probably creates a higher level of stress but I don't think it necessarily means its stress in terms of physiological discomfort but I think it sharpens the mind. Because, you know you can't procrastinate as it will be really obvious when you don't complete something. I think it's a good thing"* (Developer, InfoOrg).

> "*The (agile methodology) originators came up with this idea of work is taking responsibility for the thing that they create. So, in some sense, feeling ashamed that others might look at my code and it is atrocious because I was in a rush and had to leave early"* (Developer, OntCode).

Discussion

Based on the findings noted above, we argue that the proposed typology of ISD control provides valuable new insights into how systems development projects are controlled. The following discussion outlines why the typology is important for supplementing past work using control modes, as well as how our findings provide insights on the causes and effects of development methodology dilution.

Moving Beyond Modes

First, to evaluate whether our proposed ISD control typology supplements previous work on control modes, we coded our interviews by control modes (results are summarized in Table 2). Our results suggest that agile approaches align with behavioral and clan control. Behavioral control was evident in the description of the subtle, flexible approach to assessment in daily stand-up meetings, "*I run the daily stand-up meetings. They are very short; they always focus on what was done yesterday, what's going to be done today"* (Project Manager, InfoOrg). Clan control was also evident in agile practices, "*It's self organizing teams, so the teams control themselves, they steer their destiny. They work, so arguably I guess there is control; it's just different. They work collaboratively with the team or with the client; the team works collaboratively with the client to deliver basically what the client, at any given time, thinks is highest priority"* (Developer, OntCode).

If only modes were considered in our research, our results would seem to support the work by Harris et al. (2009) and refute the work by Maruping et al. (2009), as they deemed outcome control inappropriate in agile environments. However, we argue that our typology can help to clarify conflicting results in previous research. First, if we map Maruping et al.'s (2009) theoretical argument that decentralized decision-making (i.e. autonomy) is important when requirements are volatile to our typology, we would categorize their theoretical argument as facilitative practices with autonomy outcomes. More specifically, their argument that autonomous team members will have a heightened

need to respond in a changing environment is consistent with our position that facilitative practices reflect the workers' desire to associate themselves with organizations that give them more autonomy. Hence, if Maruping et al. (2009) developed their conceptual argument on our proposed typology, their results would likely support facilitative control practices and autonomous effects. Similarly, Harris et al. (2009) argue high change environments require the ability to reconfigure on-the-fly and outcomes in agile methods are not explicit; rather, they are developed from an iterative development process. Using our typology as a theoretical lens, they may have argued high change environments require tolerance-based control objectives and facilitative control practices. Tolerance-based control objectives incorporate changing requirements, budgets, and budgets which is consistent with Harris et al's (2009) concept of emergent outcomes. The iterative development process they describe is consistent with facilitative control practices. Hence, we conclude that had previous research adopted our proposed typology of ISD control, results related to control in flexible environments would be consistent and lend support to our contention that control modes alone are inadequate to examine control in information systems development.

Causes and Effects of Diluted Development Practices

We recognize that there are multiple explanations for why Proposition 3 was strongly supported by our research results, while Propositions 1 and 2 were not strongly supported; some of these alternative explanations are noted in the limitations section of this paper. However, one plausible explanation is that hybrid development methodologies dilute the unique benefits of waterfall and agile practices by confounding control dimensions that draw on both reductionistic and expansionistic perspectives. To investigate this concept further, we re-examined the interview data for each case study to determine if individual organizations are pursuing particular control dimensions using *both* agile and waterfall techniques. We find that in each case where a non-variant or unsupported result was noted in our data analysis, at least one of the three case study organizations employed diluted development methods. That is, the development team utilized both agile and waterfall techniques in relation to a single control objective or practice. Though this is unsurprising for InfoOrg, where a hybrid development approach is used, it is surprising at LargeMan and OntCode where increasingly extreme or 'pure' instantiations of their respective development methodologies are purported to be in place. For the control objectives category, both agile and waterfall techniques were used by: InfoOrg (speed objective); LargeMan and InfoOrg (quality objective); InfoOrg and OntCode (risk objective). In employing control practices, both agile and waterfall techniques were used by: LargeMan and InfoOrg (standardization & measurement practice); and InfoOrg (prioritization practice). Illustrative examples are outlined below.

First, we argue that the focus on quality by waterfall developers presents a threat of dilution of structured approaches in systems development. In practice, integrated performance management initiatives such as earned value analysis attempt to monitor system quality. Earned value analysis (EVA) combines measurements of scope, schedule, and cost in a single integrated system (Fleming & Koppelman 2006). Although the integration of targets may capture a more complete view of the system, our typology supports the argument that integrated performance initiatives are still target-based control objectives. As practical ISD literature presents these more advanced target-based initiatives as quality management practices (Fleming & Koppelman, 2006), it not surprising that the developers using waterfall methodologies indicate quality as a control objective. Using our typology as a theoretical framework, we argue that the adoption of facilitative quality control practices (e.g. continuous improvement, quality circles) will present a threat to structured development teams as they will encourage tolerance-based objectives that do not align to the target-based needs of structured software development approaches. Similarly, we argue that adoption of target-based performance management initiatives (such as EVA) will dilute agile practices; researchers have recognized the restrictions of these techniques in agile environments and are attempting to modify them to support more flexible approaches (Sulaiman et al. 2007).

We adopt a similar logic to explain the reference to standardization and measurement practices in agile environments. We expect that the popularity of structured performance management techniques (e.g. balanced scorecard) have influenced the adoption of standardization and measurement practices in agile environments. Consistent with our typology, we argue that these structured methods conflict with the dominant expansionistic problem solving approach in agile environments. In fact, it is possible that organizations that have made a commitment to balanced scorecard-type performance management practices may be attempting to infuse them into their systems development environments and these practices are one of those being adopted by agile environments in an attempt to overcome organizational constraints and executive resistance to change (Karlstroem & Runeson 2005; Manhart & Schneider 2004). Results from our research only show that standardization and measurement

practices have been infused into agile environments. Future research is needed to determine whether or not these practices are diluting the benefits of flexible systems development approaches.

Finally, our results show that risk management and prioritization are common in both structured and flexible environments. We propose the key to differentiation is how uncertainty is perceived. Structured risk management methods perceive uncertainty as foreseeable and prioritization practices take a preemptive planning approach where risks are identified and prioritized to avoid or mitigate possible negative effects (Kliem & Ludin 1995). In structured risk management environments, efforts are made to prevent change. In rapidly changing environments, risk objectives need to welcome change and recognize that uncertainty is unforeseen. Risk management experts suggest risk management approaches need to focus less on planning and more on flexibility and learning (De Meyer et al. 2002). Based on this reflection, we suggest that risk objectives can be target-based or tolerance-based. We also argue that prioritization practices can be structured using risk planning and identification practices and facilitative using more frequent iterative prioritization methods that are enabled by constant interaction with the customer. Re-evaluation of interview data showed some support for the different risk objectives and practices; however, as the interview protocol did not specifically address different approaches to risk, results are not conclusive. For the risk reduction objective, developers in structured environments sought to limit and clearly define change as a means to limit risk:

> *"It reduces risk, and it sets expectations so that the businesses and the users have an expectation of what they are going to get, and when they are going to be able to get it. So, you are really setting expectations there and I think that it is reducing risk...We know we've got that goal; we have an idea of what has to be done in there. And we just have to figure out how to do it… And so scheduling and managing all of that with a team of developers is what we have to do within this window of time. And having that laid out, does tend to reduce the risk because it gives you a structure to work within"* (Developer, LargeMan).

As well, prioritization practices were more structured in waterfall environments:

> *"We have a release cycle, about three releases a year, and all the change requests will be prioritized…the ones with the most priority will probably get done... The prioritization is done by the business analyst and the country representative. So, each country has a [head office] representative and then they would have a meeting, I think once a month or sometimes more frequently, depending on when the release cycle is. And they would decide on what they want on the next release and how much benefit it's going to give to their business and from that they can decide which one is the highest priority for them.* (Developer, LargeMan)

> *We laid out for a release, if we have a release coming up we may have thirty five or forty tasks. Whether they are technical or business requirements they will be laid out and need to be done. All we do in that is we go, 'okay here is when our coding needs to be done, so we have to be ready for testing by that time, here is our release date'. So everybody knows those two dates. How you get to that date, we don't care"* (Developer, LargeMan).

In agile environments, developers embraced change and prioritization methods are more iterative and customer-focused:

> *"The change in requirements are good because the customer often changes their mind or one or more people came up with the idea that by the time you needed more detail, we realized that the benefits here were different from what was originally thought. So, to be honest if I deal with the customer ongoing and if you have to change requirements I think is critical to end up with something that is useful"* (Developer, OntCode).

> *"It's more of an observation and iterative modification so, you do a bit, you observe, you change your plan, you do a bit more, you observe, you change your plan and that's basically the ideal control process in an agile project"* (Developer, OntCode).

> *"And I keep the backlog of the issues and tasks up to date, so I daily report on how much time we spent, what's left - remaining hours or remaining time as well as how it compares to the estimate from the initial of the project"* (Project Manager, InfoOrg).

In general, this line of reasoning suggests that when organizations adopt hybrid methodologies, they may dilute the core benefits of each individual methodology (e.g. agile and waterfall) by introducing contradictory forms of systems development control. Past research suggests that when inconsistent organizational control and values are present within a development team, it can result in project challenges that are difficult to overcome (Ngwenyama &

Nielsen 2003). As well, due to the varied methods, philosophies, and principles on how systems are developed, "…it could prove very challenging and confusing for ISD teams who wish to be agile when they are given completely conflicting, polar opposite advice" (Conboy 2009, p. 330). This confusion and uncertainty relates closely to the concept of control effects. Our results suggest that when the control values and perspectives of a development team are markedly different from those of management or the broader organization, negative implications of oppression and dependency may result. This observation is consistent with prior research. Hirschheim & Klein (1994) find that when methodologies fail to adequately consider the developer's inherent needs for communication, cooperation, and self-control, the process constrains and oppresses the developer. Other research suggests that the level of dependence between activities is significantly related to the extent of the organizational hierarchy (Ouchi & Maguire 1975). In contrast, where governance and control values are aligned across an organization, effects relating to satisfaction and positive stress will be more prevalent. This implies that control objectives and practices that relate to positive control effects may lead to "meaningful business outcomes at a magnitude that is important to many organizations and that these correlations generalize across companies" (Harter et al. 2002, p. 276). This finding provides preliminary evidence that the dilution of agile techniques may negatively influence the performance of systems development projects.

Implications, Limitations, and Future Research

The objective of this paper was to conduct an in-depth exploration of the dimensions of systems development control as a means to derive theory related to how structured and flexible development methodologies apply control differently. Specifically, we were interested in understanding how hybrid methodologies that integrate aspects of both structured and flexible methods may dilute the performance benefits perceived to exist in a 'pure' form of either method.

Two important contributions stem from this research. First, our findings suggest that when control is viewed using the categories and dimensions proposed in this study, unique insights can be generated regarding the systems development process that go beyond the standard control modes approach. By drawing on Ackoff's general systems theory (1974, 1981), we find support in our assertions that a new typology consisting of control objectives, practices, and effects can help to reconcile past contradictory findings related to systems development control. Previous research examining the relationship between development methodology elements such as Iivari et al. (2000/2001), has been characterized as taxonomic theory (Gregor 2006); however, this study builds on these classifications by incorporating the construct of control. By empirically establishing a typology of ISD control, the findings from this study represent a novel contribution to control theory in systems development.

Second, we provide evidence that ISD methods can be differentiated by reductionist and expansionistic elements and organizations are blending these elements in practice. Future research could examine whether or not these hybrid approaches dilute the benefits inherent in a more pure methodological form because of the contradictory control perspectives. This practical contribution can provide important insights to boards, managers, and development teams by articulating a new perspective on how development projects are controlled. By focusing specifically on the control aspects inherent in agile and waterfall methodologies, practitioners can more clearly determine the development approach that aligns with their intended objectives. The findings highlight areas where caution may be required from practitioners regarding the implications of mixing agile and waterfall development practices. We suggest that such approaches may confuse developers and erode the strength of management control, potentially leading to suboptimal project performance.

There are several limitations of this study to consider. First, because the agile and waterfall development methodologies are used as a proxy for flexible and plan-driven approaches, respectively, interpreting the findings in relation to other methodologies or approaches should be approached with caution. Although the typology components were supported across a range of the control dimensions for the methodologies studied, it should not be inferred that similar findings will necessarily be replicated at other organizations or with other methodologies. Second, all of the organizations and case study sites visited for the study were based in North America, which could limit the applicability of the findings to non-North American cultures. Prior research has noted that culture may affect the nature of control systems (Birnberg & Snodgrass 1988), as well as the manner in which systems are developed (Iivari & Huisman 2007; Leidner & Kayworth 2006); therefore, the findings of this study may be influenced by the cultural norms, values, and procedures that apply to a North American environment.

Future research can build on the findings noted in this study in three areas. First, additional research is required to continue to develop an understanding of the organizational circumstances that lead to diluted agile practices, as well

as to determine the specific project performance implications of such practices. Our findings suggest a relationship between control and diluted practices, but subsequent studies could consider other alternative causes such as the extent of cultural values alignment between organizations and project teams. Second, future research could further validate and expand on the typology of ISD control proposed in this study. Our model contributes to building a better understanding the interrelationship between control and the systems development process; however, other control categories and dimensions may exist that can provide further insights. Finally, we address the governance of ISD projects from the perspective of management control, but do not directly examine the complex role of power, politics, and authority that can influence how and why systems are developed within organizations. Future studies could consider the role of ISD governance and control as it applies from the board room to the server room as a means to better understand the complexities that can influence the process of systems development.

Acknowledgements

The authors would like to acknowledge The Monieson Centre at Queen's University for their financial support of this research, as well as the four organizations who graciously participated in the study.

References

Ackoff, R.L. 1974. *Redesigning the Future: A Systems Approach to Societal Problems,* New York, NY: Wiley-Interscience.

Ackoff, R.L. 1981. *Creating the Corporate Future*. New York, NY: Wiley.

Augustine, S., Payne, B., Sencindiver, F., and S. Woodcock. 2005. "Agile Project Management: Steering From the Edges," *Communications of the ACM* (48:12), pp. 85-89.

Beck, K. 2004. *Extreme Programming Explained: Embrace Change*, Boston: Addison-Wesley.

Beck, K., Beedle, M., Bennekum, A., Cockburn, A., Cunningham, W., Fowler, M., Grenning, J., Highsmith, J., Hunt, A., Jeffries, R., Kern, J., and Marick, B. 2009. *Manifesto for Agile Software Development.* Retrieved June 2009, http://agilemanifesto.org .

Benbasat, I., Goldstein, D. K., and Mead, M. 1987. "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly* (11:3), pp. 369-386.

Birnberg, J. G., and Snodgrass, C. 1998. "Culture and Control: A Field Study," *Accounting, Organizations and Society* (13:5), pp. 447-464.

Boehm, B. W. 1976. "Software Engineering," *IEEE Transactions on Computers* (25:12), pp. 1226-1241.

Brech, E. F. L. 1965. *Organisation,* London, England: Longmans, Green and Co.

Broadbent, M. and Weill, P. 1993. "Improving Business and Information Strategy Alignment: Learning from the Banking Industry," *IBM Systems Journal* (32:1), pp. 162-179.

Byrd, T. A. and Turner, D. E. 2000. "Measuring the Flexibility of Information Technology Infrastructure: Exploratory Analysis of a Construct," *Journal of Management Information Systems* (17:1), pp.167-208.

Caulkin, S. 2002. "Thinking outside the box", *The Observer*, (May 20).

Child, J. 1973. "Strategies of Control and Organizational Behavior," *Administrative Science Quarterly* (18:1), pp. 1-17.

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

Coad, P., De Luca, J. and Lefebre, E. 1999. *Java Modeling in Color*. Englewood Cliffs, NJ: Prentice Hall.

Conboy, K. 2009. "Agility from First Principles: Reconstructing the Concept of Agility in Information Systems Development," *Information Systems Research* (20:3), pp. 329-354.

Cross, M. C. 1928. *Types of Business Enterprise: Structure and Control,* New York, NY: Prentice-Hall.

Davis, R. C. 1940. *Industrial Organization and Management,* New York, NY: Harper.

Davis, R. C. 1951. *The Fundamentals of Top Management,* New York, NY: Harper.

De Meyer, A., Loch, C. H., and Pich, M. T. 2002. "Managing Project Uncertainty: From Variation to Chaos," *MIT Sloan Management Review* (43:2), pp. 60-67.

Dibbern, J., Winkler, J., and Heinzl, A. 2008. "Explaining Variations in Client Extra Costs Between Software Projects Offshored to India," *MIS Quarterly* (32:2), pp. 333-366.

Diemer, H. 1915. *Industrial Organization and Management,* Chicago, IL: La Salle.

Doty, D. H. and Glick, W. H. 1994. "Typologies as a Unique Form of Theory Building: Toward Improved Understanding and Modeling," *Academy of Management Review* (19:2), pp. 230-251.

Dube, L. and Pare, G. 2003. "Rigor in Information Systems Positivist Case Research: Current Practices, Trends, and Recommendations," *MIS Quarterly* (27:4), pp. 597-635.

Duncan, N. B. 1995. "Capturing Flexibility of Information Technology Infrastructure: A Study of Resource Characteristics and their Measure," *Journal of Management Information Systems* (12:2), pp.37-57.

Dyba, T. and Dingsoyr, T. 2008. "Empirical studies of agile software development: A systematic review," *Information and Software Technology* (50), pp. 833-859.

Eisenhardt, K. M. 1985. "Control: Organizational and Economic Approaches," *Management Science* (31:2), pp. 134-149.

Eisenhardt, K. M. 1989. "Building Theories from Case Study Research," *The Academy of Management Review* (14:4), pp. 532-550.

Eisenhardt, K. M. and Martin, J. A. 2000. "Dynamic Capabilities: What are they?," *Strategic Management Journal* (21:10/11), pp. 1105-1121.

Emerson, H. 1919. *The Twelve Principles of Efficiency,* 5th Edition, New York, NY: The Engineering Magazine Co.

Fayol, H. 1949. *General and Industrial Management,* London, England: Sir Isaac Pitman & Sons.

Flamholtz, E. G., Das, T. K., and Tsui, A. S. 1985. "Toward an Integrative Framework of Organizational Control," *Accounting, Organizations and Society* (10:1), pp. 35-50.

Fleming, Q. W. and Koppelman, J. M. 2006. *Earned Value Project Management*, 3rd Edition, Newton Square, PA: Project Management Institute.

Fitzgerald, B., Hartnett, G., and Conboy, K. 2006. "Customising agile methods to software practices at Intel Shannon," *European Journal of Information Systems* (15:2), pp. 200-213.

Gefen, D. and Keil, M. 1998. "The Impact of Developer Responsiveness on Perceptions of Usefulness and Ease of Use: An Extension of the Technology Acceptance Model," *Database for Advances in Information Systems* (29:2), pp. 35-50.

Gerwin, D. and Moffat, L. 1997. "Withdrawl of team autonomy during concurrent engineering," *Management Science* (43:9), pp. 1275-1288.

Giglioni, G. B., and Bedeian, A. G. 1974. "A Conspectus of Management Control Theory: 1900-1972," *Academy of Management Journal* (17:2), pp. 292-305.

Gregor, S. 2006. "The Nature of Theory in Information Systems," *MIS Quarterly* (30:3), pp. 611-642.

Gregory, A. J. 2007. "Target setting, lean systems and viable systems," *Journal of Operational Research Society* (58), pp. 1503-1517.

Harris, M. L., Collins, R. W., and Hevner, A. R. 2009. "Control of Flexible Software Development Under Uncertainty," *Information Systems Research* (20:3), pp. 400-419.

Harter, J. K., Schmidt, F. L., and Hayes, T. L. 2002. "Business-Unit Level Relationship Between Employee Satisfaction, Employee Engagement, and Business Outcomes: A Meta-Analysis," *Journal of Applied Psychology* (87:2), pp. 268-279.

Henderson, J. C. and Lee, S. 1992. "Managing I/S Design Teams: A Control Theories Perspective," *Management Science* (38:6), pp. 757-777.

Hirschheim, R. and Klein, H. K. 1994. "Realizing emancipatory principles in information systems development: The case for ETHICS," *MIS Quarterly* (18:1), pp. 83-109.

Hirschheim, R., Klein, H. K., and Lyytinen, K. 1995. *Information Systems Development and Data Modeling: Conceptual and Philosophical Foundations.* Cambridge: Cambridge University Press.

Huisman, M. and Iivari, J. 2006. "Deployment of Systems Development Methodologies: Perceptual Congruence Between IS Managers and Systems Developers," *Information & Management* (43), pp. 29-49.

Iivari, J. and Huisman, M. 2007. "The Relationship Between Organizational Culture and the Deployment of Systems Development Methodologies," *MIS Quarterly* (31:1), pp. 35-58.

Iivari, J., Hirschheim, R., and Klein, H. K. 2000/2001. "A Dynamic Framework for Classifying Information Systems Development Methodologies and Approaches," *Journal of Management Information Systems* (17:3), pp. 179-218.

Karlstroem, D. and Runeson, P. 2005. "Combining agile methods with stage-gate project management," *IEEE Software* (22:3), pp. 43-49.

Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp. 374-395.

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

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

Kirsch, L. J., Sambamurthy, V., Ko, D.-G., and Purvis, R. L. 2002. "Controlling Information Systems Development Projects: The View from the Client," *Management Science* (48:4), pp. 484-498.

Kliem, R. L. and Ludin, I. 1995. "And just-in-time…," *Journal of Systems Management* (46:6), pp. 24-26.

Koontz, H. 1958. "A Preliminary Statement of Principles of Planning and Control," *Journal of the Academy of Management* (1:1), pp. 45-61.

Koontz, H. 1959. "Management Control: A Suggested Formulation of Principles," *California Management Review*, pp. 47-55.

Kraft, P. 1977. *Programmers and Managers: The Routinization of Computer Programming in the United States,* New York, NY: Springer-Verlag.

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

Leidner, D. E. and Kayworth, T. 2006. "A Review of Culture in Information Systems Research: Toward a Theory of Information Technology Culture Conflict," *MIS Quarterly* (30:2), pp.357-399.

MacCormack, A., Verganti, R., and Iansiti, M. 2001. "Developing Products on 'Internet Time': The Anatomy of a Flexible Development Process," *Management Science* (47:1), pp. 133-150.

Manhart, P. and Schneider, K. 2004. "Breaking the Ice for Agile Development of Embedded Software: An Industry Experience Report," *Proceedings of the 26th International Conference on Software Engineering*.

Markus, M. L. and Bjorn-Andersen, N. 1987. "Power Over Users: ITs Exercise by System Professionals," *Communications of the ACM* (30:6), pp. 498-504.

Maruping, L. M., Venkatesh, V., and Agarwal, R. 2009. "A Control Theory Perspective on Agile MethodologyUse and Changing Requirements," *Information Systems Research* (20:3), pp. 377-399.

Martin, J. 1991. *Rapid Application Development*, New York, NY: Macmillan.

McGregor, D. 1960. *The Human Side of the Enterprise,* New York, NY: McGraw-Hill.

McHugh, O., Conboy, K., and Lang, M. 2008. "A Study of the Use and Effectiveness of Controls in Agile Information Systems Development Projects," *eProceedings of the 3rd International Research Workshop on Information Technology Project Management*, Paris, France.

Merchant, K. A. 1988. "Progressing Toward a Theory of Marketing Control: A Comment," *The Journal of Marketing* (52:3), pp. 40-44.

Miles, M. B. and Huberman, A. M. 1994. *Qualitative Data Analysis,* Thousand Oaks, CA: Sage Publications.

Miller, P. and Skidmore, P. 2004. *Disorganization: Why Future Organizations must 'Loosen Up'*, London: DEMOS.

Misra, S.C., Kumar, V., and Kumar, U. 2009. "Identifying some important success factors in adopting agile software development practices," *Journal of Systems and Software* (82:11), pp. 1869-1890.

Moore, G.C. and Benbasat, I. 1991. "Development of an instrument to measure the perceptions of adopting an information technology innovation," *Information Systems Research* (2:3), pp. 192-222.

Nerur, S. and Balijepally, V. 2007. "Theoretical Reflections on Agile Development Methodologies," *Communications of the ACM* (50:3), pp. 79-83.

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

Ngwenyama, O. and Nielsen, P. A. 2003. "Competing Values in Software Process Improvement: An Assumption Analysis of CMM from an Organizational Culture Perspective," *IEEE Transactions on Engineering Management* (50:1), pp. 100-112.

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

Orlikowski, W. J. 1991. "Integrated Information Environment or Matrix of Control? The Contradictory Implications of Information Technology," *Accounting, Management & Information Technology* (1:1), pp. 9-42.

Ouchi, W. G. 1979. "A Conceptual Framework for the Design of Organizational Control Mechanisms," *Management Science* (25:9), pp. 833-848.

Ouchi, W. G. 1978. "The Transmission of Control Through Organizational Hierarchy," *Academy of Management Journal* (21:2), pp. 173-192.

Ouchi, W. G. and Maguire, M. A. 1975. "Organizational control: Two functions," *Administrative Science Quarterly* (20:4), pp. 559-569.

Reeves, T. K.. and Woodward, J. 1970. "The Study of Managerial Control," in *Industrial Organizations: Behaviour and Control,* Woodward, J. (Ed.), London, England: Oxford University Press.

Romm, C. T. and Pilskin, N. 1999. "The office tyrant - social control through e-mail," *Information Technology & People* (12:1), pp. 27-43.

Royce, W. W. 1970. "Managing the Development of Large Software Systems," *Proceedings, IEEE WESCON,* pp. 1-9.

Selye, H. 1964. *From Dream to Discovery,* New York, NY: McGraw-Hill.

Shenhar, A. J. and Dvir, D. 2007. *Reinventing Project Management: The Diamond Approach to Successful Growth and Innovation*, Boston, MA: Harvard Business School Publishing.

Sia, S. K. and Neo, B. S. 1997. "Reengineering Effectiveness and the Redesign of Organizational Control: A Case Study of the Inland Revenue Authority of Singapore," *Journal of Management Information Systems* (14:1), pp. 69-92.

Silva, L. and Hirschheim, R. 2007. "Fighting Against Windmills: Strategic Information Systems and Organizational Deep Structures," *MIS Quarterly* (31:2), pp. 327-354.

Sulaiman, T., Barton, B., and Blackburn, T. 2006. "Agile EVM – Earned Value Management in Scrum Projects," *Proceedings, Agile Conference*.

Taft, D. K. 2010. "Agile development is hitting the mainstream, report says," www.eweek.com, (January 22).

Teece, D., Pasano, G., and Shuen, A. 1997. "Dynamic Capabilities and Strategic Management," *Strategic Management Journal* (18:7), pp. 509-533.

Tannenbaum, A. S. 1962. "Control in Organizations: Individual Adjustment and Organizational Performance," *Administrative Science Quarterly* (7:2), pp. 236-257.

Urwick, L. 1944. *The Elements of Administration,* New York, NY: Harper.

Weber, M. 1947. *The Theory of Social and Economic Organization,* New York, NY: The Free Press.

Weill, P. and Ross, J. W. 2004. *IT Governance: How Top Performers Manage IT Decision Rights for Superior Results*, Boston, MA: Harvard Business School Press.

West, D., Grant, T., Gerush, M., and D'Silva, D. 2010. "Agile Deelopment: Mainsteam Adoption has Changed Agility: Trends in Real-World Adoption of Agile Methods," *Forrester Research*, (January 20).

Westrup, C. 1993. "Information Systems Methodologies In Use," *Journal of Information Technology* (8), pp. 267-275.

Yin, R. K. 2009. *Case Study Research: Design and Methods,* Thousand Oaks, CA: SAGE.