Communications of the Association for Information Systems

Volume 49 Article 9

9-21-2021

Governing Intra-project Modular Interdependencies in ISD Projects: A Control Theory Perspective

Maduka Subasinghage
Auckland University of Technology, maduka.subasinghage@aut.ac.nz

Darshana Sedera
Southern Cross University, arshana.sedera@gmail.com

Shirish C. Srivastava
HEC School of Management, srivastava@hec.fr

Recommended Citation

Subasinghage, M., Sedera, D., & Srivastava, S. C. (2021). Governing Intra-project Modular Interdependencies in ISD Projects: A Control Theory Perspective. Communications of the Association for Information Systems, 49, pp-pp. https://doi.org/10.17705/1CAIS.04907

Communications of the Association for Information Systems

Research Article DOI: 10.17705/1CAIS.04907 ISSN: 1529-3181

Governing Intra-project Modular Interdependencies in ISD Projects: A Control Theory Perspective

Maduka Subasinghage
Faculty of Business Economics and Law
Auckland University of Technology
Auckland, New Zealand
maduka.subasinghage@aut.ac.nz

Darshana Sedera
Digital Enterprise Lab
Southern Cross University
Gold Coast, Australia
darshana.sedera@gmail.com

Shirish C. Srivastava
Information Systems and Operations Management
HEC Paris
France
srivastava@hec.fr

Abstract:

Though information systems development (ISD) projects use modularization as an approach to better manage complex tasks by decomposing them into simpler intra-project modules, we lack clearly established modalities for managing such modularized ISD projects. Adopting a control theory perspective and leveraging a case study research approach, we unearth the underlying "control mechanisms" that an organization leveraged to manage eight modularized ISD projects. Specifically, we explore the intra-project modular dependencies that the projects' business requirement documents indicated and use results from semi-structured interviews with project members to identify the corresponding control mechanisms. Our results indicate that, in scenarios with a low level of intra-project modular interdependencies, formal outcome and formal behavior constitute the preferred control mechanisms. However, specific situations related to flexible project practices and volatile client requirements may minimize the level of formal outcome and formal behavior control mechanisms in such projects. A low level of interdependencies between intra-project modules minimizes the need for informal clan control; nonetheless, informal clan-control mechanisms may help team members understand project requirements in a shared manner. Projects with a high level of interdependencies between intra-project modules have a high level of informal clan control. However, in some situations, projects with a high level of intra-project modular interdependencies have a low level of informal clan control often due to time pressures. Organizations may govern projects with a high level of intra-project modular interdependencies and poor structures through an enabling control style. Organizations can effectively govern projects with a low level of intra-project modular interdependencies through authoritative control style except in the projects where they assign team members to multiple projects simultaneously. By leveraging control theory to examine the intra-project modular dependencies, we add to the ongoing discourse on control theory and ISD project governance.

Keywords: Information Systems Development, Modularization, Control, Interdependencies, Coupling.

This manuscript underwent peer review. It was received 4/17/2020 and was with the authors for seven months for one revision. Julie Kendall served as Associate Editor.

1 Introduction

Approximately 70 percent of information systems development (ISD) projects fail to deliver their stipulated outcomes (Baghizadeh, Cecez-Kecmanovic, & Schlagwein, 2019) primarily due to poor project governance (Wiener, Mähring, Remus, & Saunders, 2016). ISD projects commonly employ modularization to create smaller, more manageable yet interrelated "modules" that reduce project complexity (Gershenson, Prasad, & Zhang, 2003) and increase project governance (Ravishankar & Pan, 2013). Researchers have defined modularization as a process in which one decomposes "complex tasks into simpler portions (i.e., modules) so they can be managed independently and yet operate together as a whole" (Mikkola & Skjøtt-Larsen, 2004, p. 354). Even though organizations have extensively used intra-project modules to achieve better project governance, we do not sufficiently understand the modalities through which organizations can efficiently govern intra-project modules. We posit that using a control theory perspective to theorize the mechanisms through which organizations facilitate intra-project modular governance will contribute to both research and practice.

In this study, we focus on the requirement analysis stage in ISD projects (Al-Otaiby, AlSherif, & Bond, 2005). In this stage, team members perform requirements modularization; that is, they derive mutually exclusive, independent, and yet integrated requirements modules that allow software to function as one system. In situations with high interdependencies between intra-project requirement modules, ISD projects require a higher control level because a change in one module may affect the other interrelated modules (Sanchez & Mahoney, 1996). Similarly, modules with fewer interdependencies have "embedded coordination" in that an organization can govern particular modules independently because they have fewer interactions with other modules (Sanchez, 1995). However, fewer interactions between modules imply greater communication barriers (Sosa, Eppinger, & Rowles, 2004). As a result, each modularized project team tends to work independently and has little communication with the other teams (Holmqvist & Persson, 2004). Although researchers recommend minimum interdependencies between modules to foster project efficiencies (Goulão, 2001), requirements modules in ISD projects innately depend on one another.

In Figure 1, we graphically depict three interrelated requirements modules in an ISD project: A, B and C. In ISD projects, a document called a business requirements specification (BRS) specifies each requirement. Each BRS for modules A, B and C contains the requirement specifications that technical staff members use to develop software. In Figure 1, the boundaries between modules highlight the interdependencies between them. For example, any updates to module A specifications will result in changes to modules B and C. Moreover, some requirements may result in modules having overlapping functionalities (e.g., as with modules A and C in Figure 1). Accordingly, in this example, ISD team members would need to update module C in parallel to reflect the corresponding changes in module A (and vice versa). Moreover, as module C depicts, team members can extend a module's boundaries due to changes in scope, expected outcomes, procedures to achieve expected outcomes, and the modules' inputs and outputs.

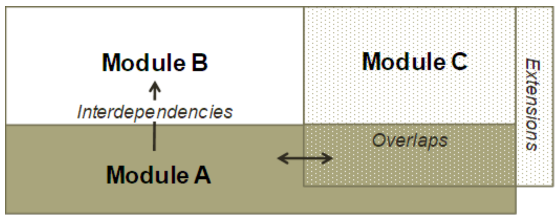

Figure 1. Interdependencies between Modules

Consistent with modularization's overarching objectives, ISD projects focus on governing each requirement module as an autonomous entity. However, in reality, interdependencies between modules in ISD projects that an integrated software solution requires yield a complex governance paradox for ISD project managers. We view this governance paradox through the control theory perspective and address the following research question:

> **RQ**: Which control mechanisms ISD project managers use for governing modularized ISD projects with varying degrees of intra-project modular interdependencies?

With this research, we make four contributions to the literature. First, although modularization interrelates with project controls (Sanchez, 1995; Sosa et al., 2004), the ISD literature has rarely explored the linkages between intra-project modules and control mechanisms. Integrating intra-project modules as contemporary systems require is a unique and painful perspective in ISD project management. Second, our research highlights the need for project managers to consider modular interdependencies in ISD projects, which may be a prerequisite to determine how to govern ISD projects and their expected output quality. Third, our research better explains the specific control mechanisms that project managers can use to govern modularized ISD projects with interdependent modules. Fourth, researchers have mainly used control theory to examine the governance between a client and a vendor in ISD outsourcing projects (e.g., see Tiwana, 2008; Rustagi, King, & Kirsch, 2008). In our study, we extend control theory by explaining how project managers can apply control theory to govern ISD projects' intra-project module interdependencies.

The paper proceeds as follows: in Section 2, we discuss control theory and module interdependencies in ISD projects. In Section 3, we explain the case study approach that we followed in this study. In Section 4, we present and discuss our results. In Section 5, we discuss our study's implications for research and practice. Finally, in Section 6, we conclude the paper.

2 Theoretical Background

Researchers have frequently employed control theory (Eisenhardt, 1985; Kirsch, 1996) to better understand how project managers devise management interventions using control mechanisms to govern projects. (Goldbach, Benlian, & Buxmann, 2018). Control theory provides the scope required to observe how project managers can govern modules to attain the relevant goals. Control theory explains two main parties in control relationships: controller and controllee. According to Kirsch (1996), "when a controller exercises control over a controllee", the controller takes "some action in order to regulate or adjust the behavior of the controllee" (p. 1). In an ISD project context, project managers play the controller role by attempting to regulate team members' (i.e., controllees) behavior.

The control theory literature (Choudhury & Sabherwal, 2003; Kim & Tiwana, 2016; Wiener et al., 2016) outlines the need for organizations to maintain a portfolio of management controls that comprises a mix of formal and informal controls. Formal control involves controlling employees via measuring, evaluating, and rewarding their performance based on their outcomes or behaviors (Kirsch, 1996). Organizations attain formal controls through outcome-based and behavior-based control modes. They implement the outcome-based mode through mechanisms that specify the expected outcomes, whereas they implement the behavior-based mode using the mechanisms that influence appropriate behaviors (Kirsch, Sambamurthy, Dong-Gil, & Purvis, 2002). Organizations use BRSs, timelines, and regular meetings to communicate the expected outcomes and behaviors in a project (Abubakre, Ravishankar, & Coombs, 2015; Choudhury & Sabherwal, 2003).

The informal control comprises social or people strategies (Kirsch et al., 2002) to govern employees. According to Jaworski (1988), informal control mechanisms refer to "unwritten, typically worker-based mechanisms that influence individual or group behavior" (p. 27). The informal control comprises the "clan" and "self-control" modes. Ouchi (1978) discusses clan control as a way to promote common values and beliefs in a clan (i.e., a group of individuals who share common goals). In contrast, self-control occurs when an organization's employees control their own actions (Manz & Angle, 1986). Because ISD projects assign team members to different modules, they may comprise multiple interdependent clans.

The control theory literature highlights the advantages that control mechanisms have in organizations, such as: 1) ensuring organizations achieve their organizational goals such as quality products and client satisfaction (Kirsch, 1996; Maruping, Venkatesh, & Agarwal, 2009), 2) motivating individuals to work in accordance with organizational goals (Cram, Brohman, Chan, & Gallupe, 2016a; Jaworski, 1988; Kirsch et al., 2002; Mao, Lee, & Deng, 2008), 3) increasing the ISD performance (Gopal & Gosain, 2010; Kirsch et al., 2002; Nidumolu & Subramani, 2003), and 4) ensuring cooperation among individuals who have partially congruent objectives (Choudhury & Sabherwal, 2003; Ouchi, 1979). Wiener, Mähring, Remus, Saunders, and Cram (2019) categorized previous the control theory literature into three categories: 1) control configuration—the control modes in a control portfolio (i.e., formal-outcome, formal-behavior, informal-clan and informal-self), 2) control enactment—the interactions between controller and controllee through authoritative style (i.e., unilateral style with limited interactions) or enabling style (i.e., bilateral style with frequent interactions), and 3) control purpose—why a controller would use control configurations and control enactments (i.e., value-appropriation purpose to harness value or value-creation purpose to maximize value).

Researchers (Cram et al., 2016a; Cram, Brohman, & Gallupe, 2016b, 2016c) have argued that, to select control mechanisms, organizations need to specifically understand the context in which they will apply them. We argue that project managers need to better understand interdependencies between intra-project modules when selecting appropriate control mechanisms to manage ISD projects.

Several studies discuss why organizations choose control modes such as outcome control (Kirsch, 1997; Snell, 1992), behavior control (Eisenhardt, 1985; Jaworski & MacInnis, 1989), clan control (Kirsch, 1996; Ouchi, 1979) and self-control (Henderson & Lee, 1992; Kirsch, 1997; Wiener, Remus, Heumann, & Mähring, 2015). Researchers have employed control theory to understand the factors that influence how organizations exercise control such as relationship characteristics (e.g., resource availability, role expectations) (Choudhury & Sabherwal, 2003; Cram & Brohman, 2013; Rao, Brown, & Perkins, 2007) and task characteristics (e.g., behavior observability, outcome measurability) (Kirsch, 1996; Remus & Wiener, 2012). However, previous research has not adequately explained the need for project managers to consider the level of interdependencies between intra-project modules when deciding control mechanisms for projects.

Two main attributes relate to how well ISD projects decompose modules include coupling and cohesion (Hitz & Montazeri, 1995). According to Kwong, Mu, Tang, and Luo (2010), "coupling is about the measure of interactions among software modules while cohesion is about the measure of interactions among the software components which are within a software module" (p. 619). Although several researchers (Allen & Khoshgoftaar, 1999; Goulão, 2001) have discussed the need for ISD projects to have high cohesion and low coupling, such projects cannot easily obtain these attributes (Taube-Schock, Walker, & Witten, 2011). Taube-Schock et al. (2011) state that "high coupling is impracticable to eliminate entirely from software design" (p. 23). As a result, projects may include interactions between different software modules and, thereby, create the need for high control in projects. In this study, we focus on coupling (i.e., interactions between software modules).

In Appendix A, we show existing research that has integrated modularization, interdependencies, project control, and governance concepts in the ISD project context. In summary, we address two concerns. First, modules with fewer interdependencies have "embedded coordination" because ISD projects can govern individual modules separately when they have fewer interdependencies between them. As such, project managers can provide clear instructions for each module so that the team members assigned to that module can complete assigned tasks without needing to interact much with team members assigned to other modules, which minimizes the need for organizational control (Sanchez, 1995). However, projects require controls to ensure that the tasks that team members from multiple intra-project modules perform align with one another. Second, little literature has discussed the linkages between intra-project modularization and control (Tiwana, 2008). Tiwana (2008) states that modularity could substitute for control but does not describe the mechanisms for governing intra-project module interdependencies. Furthermore, Tiwana (2008) focuses solely on the formal controls and does not explicitly discuss the relationship between intra-project modularization and informal-clan control. We address these key concerns in this paper.

2.1 The Nature of Interdependencies and the Use of Control Portfolios

In this section, we combine control theory's theoretical scaffolds with modular interdependencies' fundamentals to assess how one can develop a control portfolio to govern intra-project module interdependencies.

2.1.1 Interdependencies and Formal Controls

ISD projects employ BRSs to provide a formal structure that focuses on a module's outcomes. Since ISD projects comprise many modules, these modules will likely depend on one another. Modularizing architectural components can minimize the technical interdependencies among the teams that develop interrelated components and, thereby, increasing the likelihood of development productivity (Mirani, 2007) and project success (Cataldo & Nambiar, 2012). Modularization can increase the software development process's efficiency by minimizing communication overheads (Cataldo, Bass, Herbsleb, & Bass, 2007). Since unanticipated interdependencies between software development tasks can occur (Kraut & Streeter, 1995), project managers can find it challenging to govern projects through modularization.

According to Sethi, Yuanfang, Wong, Garcia, and Sant'Anna (2009), in situations with fewer interdependencies between modules, organizations can easily replace or update the modules. Cai and

Huynh (2007) highlight that modularization techniques, such as aspect-oriented programming and object-oriented design patterns, allow organizations to make changes to some IS project modules without affecting other modules. Therefore, changes to a particular module may not affect the tasks of team members who work on other modules.

When project managers modularize IS functionalities, they can provide team members with tasks that have fewer interdependencies. In doing so, project managers gain the ability to measure team members' outcomes and behaviors effectively. Therefore, when projects include intra-project modules with fewer interdependencies, project managers tend to use formal controls to govern team members.

2.1.2 Interdependencies and Informal Controls

ISD projects comprise various teams, such as the business analyst team, software engineering team, and software quality assurance engineer team (Nuwangi, Sedera, Srivastava, & Murphy, 2013). Although these teams differ significantly in their formal structure, cognitive orientation, and departmental mission (Andres, 2002), they all share the same project goals and objectives (Ouchi, 1980). For example, while business analysts need to identify business requirements and write the BRS (Capretz & Ahmed, 2010), software engineers need to develop software according to the BRS (Hofmann & Lehner, 2001). Hoegl, Parboteeah, and Gemuenden (2003) state that insufficient collaboration between team leaders can result in task duplication and, thus, reduce team members' ability to complete the project within the stipulated time and budget. Frequent interactions between team members provide the ability to effectively share knowledge (Zimmermann & Ravishankar, 2014) and understand the project's goals in a shared manner (Levesque, Wilson, & Wholey, 2001).

In modularized ISD projects, the team members work on different modules. For example, a financial management module could comprise members from the software engineering team and members from the business analyst team. According to Sosa et al. (2004), modularization creates boundaries between teams and, thereby, increases communication barriers. Due to fewer interdependencies between modules, teams can complete tasks independently. A positive level of interdependencies between modules leads to corporative behaviors between team members (Ghobadi & D'Ambra, 2013). For example, when a requirement interdependency between team members from two different modules exists, the team members from the different modules tend to work cooperatively (Ghobadi, 2015). This cooperation increases the extent to which project managers use informal-clan controls in projects.

When a project has a low level of interdependencies between intra-project modules, project managers can give team members more authority to make their own decisions (i.e., a high level of self-control) (Sanchez & Mahoney, 1996). For example, in situations with fewer interdependencies between modules, software engineers assigned to a module can develop the software module with fewer interactions with software engineers from other modules. In contrast, in situations with more interdependencies between modules, software engineers assigned to a particular module must collaborate frequently with software engineers assigned to other independent modules to manage the interdependencies.

2.1.3 Interdependencies, Control Styles, and Control Purpose

Wiener et al. (2016) discuss two control styles—authoritative control style and enabling control style—that project managers can use to govern team members depending on the level of interdependencies between intra-project modules. When the team members in different intra-project modules conduct highly interdependent tasks, team members in one module may not be able to commence their tasks until team members in the other modules complete their tasks, which can lead to unexpected delays in the software development process. Thus, while following an authoritative control style, project managers can find it challenging to guide team members with strict guidelines. Project managers can effectively govern projects with high level of interdependencies through an enabling control style since it allows them to work collaboratively with team members to manage contingencies. However, in some situations, project managers may find it beneficial to use authoritative control style to guide projects with a high level of interdependencies to ensure that they achieve expected outcomes on time. In contrast, having fewer interdependencies may enhance project managers' ability to guide team members through authoritative control style as they can find it easier to provide clear guidelines and monitor team members' progress. When projects have intra-project modules with fewer interdependencies, an enabling control style can support more collaboration between team members in different modules and, thus, lead to quality products.

The two types of control purpose include: 1) value-appropriation purpose and 2) value-creation purpose (Wiener et al., 2019). When intra-project modules have a high level of interdependencies, team members can find it difficult to achieve value appropriation (i.e., harness value). For example, when the tasks that software engineers perform in different modules have a high level of interdependencies, the software engineers in one a module may not be able to commence their software development tasks until the software engineers in the other module complete their development tasks (Mirani, 2007). Thus, team members may find it difficult to complete software development tasks on time and at the expected quality (i.e., they find value appropriation difficult). The level of interdependencies between intra-project modules can affect value creation negatively or positively. For example, in situations with a high level of interdependencies between intra-project modules, team members tend to collaborate more with one another (Ghobadi, 2015), which can ultimately increase their ability to create value. However, in some situations, a high level of interdependencies can lead to disturbances in the software development process (Kraut & Streeter, 1995) and, thus, to difficulties in the value-creation process.

3 Research Methodology and Context

This study constitutes an exploratory study given that little research has investigated how organizations govern intra-project modular interdependencies. Accordingly, we adopted a multiple-case study approach since researchers have recommended such an approach for exploratory studies (Yin, 2003). Moreover, researchers have recognized the qualitative case study approach as appropriate for research that explores complex environments and contemporary events (Benbasat, Goldstein, & Mead, 1987; Srivastava & Shainesh, 2015).

To select our organizational sample, we followed the purposeful sampling strategy in line with our research objectives (Patton, 2002). We used three conditions as benchmark criteria to select ISD organizations. First, the organization needed to modularize its business requirements. Second, it needed to have multiple ISD projects to provide flexibility in data collection. Third, it needed to be sufficiently large with a standard employment hierarchy so we could collect data from team members at different employment levels.

After applying these criteria, we selected a company called “Soft-dev” (a pseudonym) as the case organization. We selected eight ISD projects (see Appendix B for the project descriptions) in the company as the cases to ensure control and replicability. In this paper, we focus on identifying the types of control mechanisms that organizations use to govern modularized ISD projects that vary in their level of intra-project module interdependencies. Thus, we collected data from the eight projects to identify the level of interdependencies in each project and delineate the corresponding control mechanisms that each project used. To select the projects, we followed the opportunistic/emergent sampling strategy. According to Patton (2002), opportunistic/emergent sampling follows new leads during fieldwork and takes advantage of unexpected flexibility. The projects we selected resembled one another based on their industry sector (ISD projects), the information system they developed (stock exchange systems), and project stage (completed) but varied in their clients (from different countries) and team members (different personnel). The organization had already finished the selected projects, which meant the respondents could discuss the control mechanisms that each project used throughout its life.

We conducted 23 semi-structured interviews that each lasted 25 to 30 minutes with employees from the eight ISD projects (see Appendix C for the details). In order to avoid key informant bias, we conducted interviews with multiple informants from each project (Kumar, Stern, & Anderson, 1993). We used non-probabilistic, purposive, and “snowball” sampling to ensure we interviewed opinion leaders with well-developed views on the research topic (Minichiello, Aroni, Timewell, & Alexander, 1995). The interviews comprised semi-structured, open-ended questions, such as questions about general project information and questions related to project controls and interdependencies. We present the guiding interview questions in Appendix D. We asked general, non-directive, and flexible interviews questions as a guide in our interviews rather than direct and specific questions on project controls and interdependencies so that we could explore participants’ thoughts and interests in depth and, thus, uncover interesting findings (Doody & Noonan, 2013). We designed the guiding interview questions to cover the interdependencies between modules and control mechanisms. For example, we asked participants about the documents and software systems that they used to transfer the client requirements. The participants used BRSs as the main document in their ISD projects. Each module had one BRS that captured the client’s requirements for that particular module. Thus, in describing the interdependencies between requirements in the BRSs, participants indicated interdependencies between modules. Moreover, the project managers used the

BRSs as a formal control mode (i.e., to specify each module's expected outcomes). We asked the interview participants follow-up questions as appropriate during the emerging conversations. We designed the interview questions related to modularization, formal control, and informal clan control to focus on the entire project rather than a specific module. Thus, the interview data captured the individuals' opinions about the entire project. We audio-recorded all the interviews and transcribed them for subsequent data analysis. While conducting the interviews, we took additional notes (e.g., the documents respondents used and their primary purpose) whenever necessary. We supplemented the interview data with documents such as BRSs, test scenarios, and test case specifications. These documents increased the collected data's validity and reliability.

3.1 Data Analysis

We selected the ISD project as our unit of analysis. Our study satisfied the criteria for rigor: namely, construct validity, internal validity, external validity and reliability (see Appendix E). We analyzed the data in two phases: 1) within-case analysis and 2) cross-case analysis. We conducted the within-case analysis to become familiar with each case as a standalone entity. Furthermore, we conducted this analysis to identify unique patterns in each case. Subsequently, we conducted a cross-case analysis to investigate the similarities and differences between the cases.

We adopted a deductive approach (Thomas, 2006) to identify the level of interdependencies and control portfolios that each project employed. We identified characteristics and sample codes for interdependencies from studies such as MacCormack, Rusnak, and Baldwin (2007), Tiwana (2008), and Mani, Srikanth, and Bharadwaj (2014). We identified characteristics and sample codes for control configurations, control enactment, and control purpose from well-established control theory studies such as Choudhury and Sabherwal (2003), Kirsch et al. (2002), and Wiener et al. (2019). In Sections 4.1 to 4.6, we explain how we derived these characteristics and sample codes using examples.

Using the identified characteristics and sample codes as a guide, we assessed and reassessed the interview data and BRSs several times to identify the level of interdependencies, control configurations, control enactment, and control purpose. We moved back and forth between the transcripts, BRSs, and the literature to remain true to the extant theory while interpreting the data for emerging findings (Sarker & Sahay, 2003).

4 Results and Discussion

4.1 Establishing the Level of Interdependencies

In order to identify the control mechanisms that organizations require to effectively govern modularized ISD projects with varying degrees of interdependencies, one needs to recognize the level of interdependencies in each project. Accordingly, we prepared a coding guideline by including initial interdependency characteristics with guidance from prior studies (e.g., MacCormack et al., 2007; Tiwana, 2008; Mani et al., 2014). For example, when an ISD project properly modularizes requirements, a particular module should have less impact on other modules (Allen & Khoshgoftaar, 1999; Goulão, 2001). Therefore, we identified the "impact one module has on other modules" as one characteristic that we could use to identify the level of interdependencies between modules. During the coding process, we enhanced the coding guideline with newly identified characteristics. We included these characteristics in the coding guideline after carefully considering each concept's definition. We show the categorization standards that we used to categorize the level of interdependencies in each project in Appendix F.

Following the coding guideline, we categorized the level of interdependencies between modules in each project as high or low. We show the estimated level of interdependencies between the modules in each project in Table 1.

Table 1. The Level of Interdependencies between Modules in Each Project

Project	Discussion	Sample quotations	Level of interdependencies between modules
A	Business analysts removed some of the initial requirements from the BRSs since the initially agreed requirements **couldn't be implemented without implementing interdependent requirements.** [Project comprised modules that had a significant impact on other modules]*	*This requirement **cannot be implemented without that** [requirement] because it clashed with other requirement. So, a big requirement was removed* (Respondent 01).	Since the project comprised modules that had a significant impact on other modules and business analysts did not identify the modules that had an impact on other modules, we categorized the level of interdependencies as **high**.
	Due to the project's large scope, the consultants found it difficult to **identify the interdependencies** between requirements. [Business analysts did not identify the modules that had an impact on other modules]	*Most of the time, we also have some problem, because…the system was very large…. It is very **hard to identify what are the areas which have an impact**, because it is very large* (Respondent 01).	
B	The project managers assigned software engineers to modules, which had **dependencies**. The software engineer leads maintained an Excel sheet that included task interdependencies in the software engineering team. This sheet indicated various **interdependencies** between team members' tasks who worked on different modules. [Team members required significant input from other modules to complete their tasks]	*There are cross domain [module] **dependencies**. Now I'm 100% in development of particular area. There is another one who's handling a particular development area* (Respondent 05). *I just add **dependencies** task by task and I say you [project manager] can make a project plan using this thing I provide. There are ten tasks, one after the other* (Respondent 05).	Since team members required significant input from other modules to complete their tasks, we categorized the level of interdependencies as **high**.
C	The modules had interfaces between them. When team members had to integrate a new module to the existing modules, they had to consider the **interfaces** that connected the new module with the existing modules. [Interfaces connected different modules; when teams had to integrate a new module to existing modules, they had to consider the interfaces that connected the new module with the existing modules]	*We need to get the **interface** done and then test the requirement and the functionality* (Respondent 09).	When the teams had to integrate a new module to the existing modules, they had to consider the interfaces that connected the new module with the existing modules. Thus, we categorized the level of interdependencies as **low**.
D	The project's business analysts did not identify interdependencies between modules. As a result, some modules that **impacted** other modules. [Business analysts did not identify the modules that had an impact on other modules]	*If we are writing the specification, **we have to analyze the impact areas. Those areas were not clearly analyzed*** (Respondent 10).	Since business analysts did not identify the modules that had an impact on other modules, we categorized the level of interdependencies as **high**.
E	The project distributed the **responsibility** for individual modules among software engineers. [Teams worked on different modules]	*When a developer [software engineer] takes the **responsibility** of one component [module], he has the responsibility of changing the product document* (Respondent 13).	Since the team members worked on a different module and the project distributed their responsibility accordingly, we categorized the level of interdependencies as **low**.

Table 1. The Level of Interdependencies between Modules in Each Project

F	Team members worked on different modules and **took total ownership of their assigned modules**. [Teams worked on different modules]	*Everyone took [the]* ***total ownership of their components [modules] and the tasks*** (Respondent 16).	Team members worked on different modules and took ownership over their assigned modules. Thus, we categorized the level of interdependencies as **low**.
G	Team members worked on different modules and received **isolated work, which indicates** that they could work on different tasks independently. [Teams worked on isolated tasks that they had to complete independently]	*Most of the time they [team members get]* ***isolated work*** (Respondent 19).	Since the team members worked on isolated tasks that they had to complete independently, we categorized the level of interdependencies as **low**.
H	Functionalities of the project were **broken down into modules.** [Business analysts subdivided the IS solution into several modules]	*The full functionality is* ***broken down to components [modules]*** (Respondent 22).	Since the functionalities of the IS solution were subdivided into several modules and the team members were assigned accordingly, we categorized the level of interdependencies as **low**.
* Relevant categorization standards from Appendix F.			

4.2 Establishing the Level of Formal Controls

In order to identify whether a project executes formal controls when it contains a low level of interdependencies between modules, one needs to examine the level of formal control in them. Accordingly, we prepared a coding guideline to identify the level of formal outcome and formal behavior controls in each project. We adapted the coding guidelines from Kirsch (1997) and Choudhury and Sabherwal (2003). For example, Choudhury and Sabherwal (2003) note that one can identify "mechanisms to explicitly specify desired outcomes that were assessed later" (p. 301) as outcome controls. Therefore, we identified "the BRSs specified the expected outcomes in detail" as one characteristic of formal outcome controls (see Appendix F for the categorization standards). Following the coding guideline, we categorized the level of formal outcome and formal behavior controls in each project as high or low. We show the estimated level of formal controls in each project in Table 2.

Table 2. The Level of Formal Controls

Project	Discussion	Sample quotations	Level of formal controls
A	When modules were not feasible, software engineers had to identify alternative methods to accommodate the requirements of the particular modules. Due to the changes, business analysts had to **update the BRSs**. [Business analysts updated the BRSs several times, which indicated unstable expected outcomes]*	*Sometimes during developer discussions, they say this [module] is not feasible. During that discussion itself, you have to come up with [an] alternative to cater the functionality [module]. So after that what we [business analysts] did was, we* ***just updated the BRSs*** *and sent another version*. (Respondent 01)	The project's BRSs did not properly specify the IS solution's expected outcomes. Moreover, business analysts updated the BRSs several times during the project lifecycle. The developed IS solution did not completely align with the BRSs. Thus, we categorized the level of formal outcome control in project A as **low**.
	Team members **frequently updated BRSs**. [Business analysts updated the BRSs several times, which indicated unstable expected outcomes]	*They [software engineers] have to change certain things, because the document* ***[BRS] is changing, it is*** *changing continuously.* ***It is very frequently changing.*** (Respondent 02)	

Table 2. The Level of Formal Controls

	Signed off BRSs did not include **detailed information** about the project's functionalities. [The BRSs did not properly specify the expected outcomes]	*They [clients] signed off the business functionality basically. So, the* ***BRS don't have this is*** *exactly* ***how we are going to give [the IS solution] to you.*** (Respondent 01)	The software code includes the process that team members should follow to achieve the IS solution's expected outcomes (i.e., the behavior controls). Software engineers frequently updated the project's software code. Thus, we categorized the level of formal behavior control as **low**.
	Since team members maintained two separate BRSs: 1) external BRSs (which they sent to the client) and 2) internal BRSs (which they updated with implementation details), **the IS solution did not completely align with the signed-off BRSs**. [The developed information system solution did not completely align with the BRSs]	*Sign off…BRS are like one set, we have like a new set of BRSs, which are something* ***different from the signed off…BRSs*** (Respondent 01).	
	Software engineers had to **amend the software code** frequently. [Team members made several updates to the software code]	*They* ***[software engineers] have to change certain things****, because the document [BRS] is changing, it is changing continuously* (Respondent 02).	
B	The BRSs **lacked the required information** about the module interdependencies. Thus, during the project, team members had to update the BRSs to include such information. [The BRSs did not properly specify the expected outcomes]	*There can be* ***missing parts*** **[in the BRS],** *so there can be something [interdependencies] people [business analysts] can't identify. So, those should be sorted out iteratively and we should make sure [that] BRS are something very detailed, [where there is no] ambiguities about the content* (Respondent 05).	Since team members did not properly specify expected outcomes in the BRSs, we categorized the level of formal outcome control as **low**. Team members made changes to the software design. Thus, we categorized the level of formal behavior control as **low**.
	Since the BRSs contained **insufficient information**, the software engineering team requested additional information for the ISD process. [The BRSs did not properly specify the expected outcomes]	*We [software engineering team] need* ***additional information*** *when we are going to do the implementation* (Respondent 05).	
	Inaccurate modularization originated software **design changes**. [Team members made several updates to the software design]	*The thing is such a* ***design change,*** *actually before documenting also we…have to refer to the developer.... So, then only we start the documentation side.* (Respondent 06).	

Table 2. The Level of Formal Controls

C	The BRSs did not adequately specify information about the module interdependencies and the client business requirements. [The BRSs did not properly specify the expected outcomes]	*Most of the time, the spec **[BRS] carry out only high-level requirements**. If you take [BRSs in] another project very, very detailed than [our project] (Respondent 07). Talking about [our project] the specs [BRSs] are little bit loose. So, it is little bit tough to do a development. Mainly based on that specs* (Respondent 08).	Since the BRSs did not properly specify the expected outcomes and team members updated the BRSs several times during the project, we categorized the level of formal outcome control as **low**. Except for in critical deliveries, project managers did not closely monitor team members' progress. Thus, we categorized the level of formal behavior control as **low**.
	Business analysts accepted the software engineers' suggestions for achieving ISD requirements. Team members updated the BRSs after approval from the client. [Business analysts updated the BRSs several times, which indicated unstable expected outcomes]	*If they [software engineers] suggest new ways of doing it [ISD], we [business analysts] think it is good. [We] ask our client first whether they are okay with that, then we accept it, change the spec [BRS] as well* (Respondent 07).	
	Project managers sent daily follow-up emails only during critical ISD project deliveries. [Project managers did not closely monitor team members' progress]	*The deadline is closer and when we [are] having a critical delivery then they [project managers] will send the daily emails; other than that, no* (Respondent 08).	
D	Since the BRSs did not specify modules, team members could not completely understand the software requirements. [The BRSs did not properly specify the expected outcomes]	*Documents [BRSs] do not provide some example or **do not specify the areas [modules]**. Then, we [can] think in several ways. So, that is the main thing* (Respondent 10).	Since the BRSs did not properly specify the expected outcomes, we categorized the level of formal outcome control as **low**. Project managers did not closely monitor team members' progress. Thus, we categorized the level of formal behavior control as **low**.
	Project managers did not monitor the issues assigned for each team member. [Project managers did not closely monitor team members' progress]	*We [project managers] don't bother about **how many issues** against each person.* (Respondent 12).	
E	Team members did not amend initially agreed-on client functionalities. Therefore, the system functionalities and deliverables aligned with client expectations. [The developed information system solution aligned with the information specified in BRSs]	*The **client functionality [will] not change**. **Everything is matched with the client*** (Respondent 13).	Since the developed IS solution aligned with the information specified in BRSs, we categorized the level of formal outcome control as **high**. Project managers closely monitored team members' progress. Thus, we categorized the level of formal behavior control as **high**.
	Project managers constantly monitored team members' progress by contacting them and requesting updates on assigned tasks. [Project managers closely monitored team members' progress]	*They [project managers] know how [to] **monitor the progress** [and] **deadlines**. They contact us to **get updates**, what is the stage [of assigned tasks]* (Respondent 13).	

Table 2. The Level of Formal Controls

F	Team members highlighted the importance of having the IS solution closely align with the BRSs. [Team members had to strictly follow the BRSs and provide the exact requirements that the BRSs specified, The developed information system solution aligned with the information specified in BRSs]	*[We **follow the BRS** in] a one-to-one basis, (Respondent 16)* *The idea of the document **[BRS] has to be closely synchronized with the actual system** behaviors* (Respondent 16). *[At the] end of the day, the **document** have to…**reflect** exactly how the **system behaves*** (Respondent 16).	Team members had to strictly follow the BRSs and provide the exact requirements specified in the BRSs. The developed IS solution had to closely align with the information in BRSs. Team members made no or few updates to the BRSs. Moreover, project managers established project deadlines to ensure that the project achieved its expected outcomes on time. Thus, we categorized the level of formal outcome control as **high**. Software engineer leads conducted reviews to ensure that the team executed the project according to the project plan. Thus, we categorized the level of formal behavior control as **high**.
	Generally, the business analysts did not aggregate new ideas to the BRSs. [Business analysts made few updates to or did not update the BRSs at all, which indicated stable expected outcomes]	*They* [business analysts] ***do not aggregate new ideas to the BRSs**. It won't happen all the time, especially when it comes to GUIs* [graphical user interfaces] *and reference data areas, it is not happening* (Respondent 17).	
	Team members completed the project on time. [he project had deadlines to ensure that team members achieved the expected outcomes on time]	*The **time** we [were] supposed to go live, we have **met that** [deadline and we] did not have to postpone [the deadline]* (Respondent 16).	
	The software engineer leads reviewed each individual software engineer's tasks to ensure that the tasks aligned with the total project plan. [Project managers conducted project reviews and project meetings to ensure that team members executed the project according to the project plan]	*The development leads do the **reviews** to make sure the time that we need to complete the project* [is] *not too high. They* [development leads] *make sure we do not have redundant tests, which would push the project timeline too far beyond the accepted delivery date* (Respondent 16).	
	Project managers conducted weekly meetings to monitor team members' progress and to ensure they executed project according to the project plan. [Project managers conducted project reviews and project meetings to ensure that team members executed the project according to the project plan]	*"We have at least weekly meetings to make sure everyone is on the same page"* (Respondent 16).	

Table 2. The Level of Formal Controls

G	Software engineers did not request BRS updates, which indicates that the software engineers were **satisfied with the details about expected outcomes specified** in the BRSs. [The BRSs specified the expected outcomes in detail]	*Unlike in other projects, the development leads reviewed the requirement and they gave some insight, whether it is possible. Based on the knowledge they* [tech leads had], *they gave some inputs and when it came to our* [software engineers] *level it was up to some level of acceptance. We **didn't have to go and change requirement** and so on. Because, they* [tech leads] *involved in the initial stages* (Respondent 19).	Project managers established project deadlines to ensure that the project achieved expected outcomes on time. Due to tight schedules, team members could not document all the decisions that they took during the project. However, they wrote the IS solution's expected outcomes in detail in the BRSs. Therefore, we categorized the level of formal outcome control as **high**. Since project managers closely monitored team members' progress, we categorized the level of formal behavior control as **high**.
	Due to **tight schedules,** team members could not document each and every decision. While they confirmed major decisions via emails or documentation, they did not document some non-critical decisions. [The project had deadlines to ensure that team members achieved the expected outcomes on time]	*Because of the…**very tight schedules** sometimes we can't document each and every decision. Such things happen…we can't document each and everything. If is a major decision, we would document it and send out emails [to] capture it* (Respondent 19).	
	Intermediate supervisors in the project **tracked team members' work** on a daily basis, which shows that they conducted daily project reviews to ensure team members conducted the project as per the project plan. [Project managers closely monitored team members' progress]	*There are lots of people managing* [the project]*. There is a layer* [on] *top of us to manage it. So, …our immediate supervisors **track each and everyday percentage of work done**…and what is not done and maybe they are the people who report to PMs [project managers]* (Respondent 19).	
H	Sometimes the requirements included in BRSs lacked clarity. [The BRSs did not properly specify the expected outcomes]	*Sometime the requirement things are not clear* (Respondent 22).	Since team members mentioned that the requirements specified on the BRSs lacked clarity, we categorized the level of formal outcome control as **low**. Since team members made several updates to software code and project plans, we categorized the level of formal behavior control as **low**.
	When the business analysts and the software support team requested **software code changes**, the software engineers updated the code accordingly. Since the software engineers had a significant workload, they did not document the code changes accurately, which created issues later in the ISD lifecycle in that other software engineers found it difficult to identify the code changes. [Team members made several updates to the software code]	*When the support and the BA [business analyst] guys ask us [software engineers] to change **[the code], we are changing**. But the thing is, since we [have] lots of work; we are not focused on the documentations. It is a problem here…. We have to ask senior or another developer in another project or we have spent lot of time studying [the] code, what is this doing, what is this functionality, what is the use of this functionality* (Respondent 22).	
	Project managers **updated the project's plans** twice a week. [Team members made several updates to the project plans]	*We have the **changes happening in the plan** every twice a week because…we have the new changes coming into the plan* (Respondent 21).	
* Relevant categorization standards from Appendix F.			

4.3 Establishing the Level of Informal Clan Controls

In order to identify whether a project executes informal clan controls when it contains a high level of interdependencies between modules, one needs to examine the level of informal clan control in them. Accordingly, we prepared coding guidelines to examine the level of informal clan control that we adapted from Kirsch (1997) (see Appendix F for the categorization standards). Following the coding guidelines, we categorized the level of informal clan controls in each project as high or low (see Table 3).

Table 3. The Level of Informal Clan Controls

Proj.	Discussion	Sample quotations	Level of informal clan controls
A	Team members in the project displayed low team spirit. [The team displayed low team spirit]*	*When compare [this project with] other projects I have worked,* ***teamwork is low*** (Respondent 03). *[In this project], the* ***team spirit was bit like less or minimum*** **(Respondent 01).**	Since respondents who worked in the project mentioned that the team members lacked team spirit, we categorized the level of informal clan control as low.
B	Since the team had high team spirit, they could deal with unplanned situations. [The team displayed high team spirit]	*Actually, that was the main reason why this project went live. Because, everything didn't happen in the proper, official, standard or expected way. We are just [dealing with] change all the time. But, people* ***had the team spirit*** *that is why the project plan went live* (Respondent 05).	Since team members had good interactions and collaborations and a high team spirit, we categorized the level of informal clan controls as **high**.
	The business analysts **discussed** the design changes with the software engineering team, even before documenting the changes. They updated documents after the software engineers agreed on the changes. [Team members had good interactions, relationships, and collaborations]	*The thing is such a design change, actually before documenting also we... have to* ***refer to the developer****.... So, then only we start the documentation side* (Respondent 06).	
C	Although business analysts modularized the requirements, the software engineers had to **interact** with the business analysts to get more information that the ISD required. [Team members provided suggestions or clarifications to other team members]	*We...have to* ***interact*** *with the BAs* [business analysts] *a lot, and get clarification and all* (Respondent 08).	Since the team members had good relationships and the team members interacted with each other often, we categorized the level of informal clan controls as **high**.
	The business analysts and the software engineers had a good relationship. [Team members had good relationships]	*We have* ***good relationship*** *[between] the developers and BAs [business analysts], not an issue there* (Respondent 08).	

Table 3. The Level of Informal Clan Controls

D	The business analysts had to **interact** with the other team members in order to finish the BRSs on time. [Team members had good interactions]	*We have a good **interaction**. If we [business analysts] have to provide the document on time, we have to talk about it* (Respondent 10).	Since the team members had good interactions and displayed a high team spirit, we identified the level of informal clan control as **high**.
	When the team had to finish some tasks, all the team members worked extra hours. [The team displayed high team spirit]	*No one tried to do their own work and leave office and so on. **Everyone worked** even late hours, I can remember* (Respondent 11). *Basically, everyone in the team is in that **same mind set**, they… work and they go and play* (Respondent 12).	
E	In situations where a particular team member could not complete their tasks on time, other team members **supported the member in completing the tasks**. [When some team members required support to complete the assigned tasks, other team members were willing to provide the required support]	*In [our project] we have a **very good team**. The team is much bigger now…. Normally everybody…knew about each other strengths and weaknesses as well. Therefore, we don't have much problem to work late night or weekends or problems don't occur. Somebody fails to do [a] certain delivery or something; we [were able to] **manage [it using] the available team members*** (Respondent 13).	When some team members required support to complete the assigned tasks, other team members willingly provided support. Therefore, we identified the level of informal clan controls as **high**.
	When required, team members helped one another to complete their tasks. [The team displayed high team spirit]	*Question: It means the [team] members…think about the other people and work as the team. They help each other.* *Answer: Yes…. It is like [a] **family** once we figure [it] out* (Respondent 13).	
F	Team members mentioned that the project had a good team spirit. [The team displayed high team spirit]	*Because it is a single team who handle all this and they have been together for some time, the **team spirit** is there.* (Respondent 17).	Since the team members mentioned that the project had a good team spirit, we categorized the level of informal clan controls as **high**.
G	Since team leaders modularized the project appropriately, the software engineers **rarely had collaborations** with the business analysts. [Team members did not collaborate often]	***Only few changes we [software engineers] have to go and ask** [from the business analysts], can we do this change? Because, in the client [requirement gathering phase] already they [business analysts] have done that* (Respondent 19).	Since the team members mentioned that they rarely collaborated with one another, we categorized the level of informal clan controls as **low**.
H	Team members collaborated often when they wanted to identify a solution to an issue. [Team members had good collaborations]	***Full team sit together and talk** "is this the correct way to do it?"* (Respondent 21). *First we **engage with the team** and find the solution* (Respondent 21). *If there is a problem, they* [team members] *are…in the same flow. It is just a matter of going there or picking up the phone and **calls them*** (Respondent 21).	Since team members collaborated well, we classified the level of informal clan control as **high**.
* Relevant categorization standards from Appendix F.			

4.4 Establishing the Level of Informal Self-controls

We adapted the coding guidelines for informal self-controls from Choudhury and Sabherwal (2003). For example, Choudhury and Sabherwal (2003) identified self-control mechanisms as: 1) mechanisms that can encourage or motivate team members to exercise greater self-control and 2) mechanisms that help enhance team members' ability to exercise control. Therefore, when preparing the coding guideline, we considered "mechanisms encouraged and motivated team members to make their own decisions and solutions" as one characteristic of informal self-controls. We show the estimated level of informal self-controls in each project in Table 4.

Table 4. The Level of Informal Self-controls

Project	Discussion	Sample quotations	Level of self-controls
A	Since the team members lacked awareness about the exact information on the client functionalities, the team members had the **freedom** to elaborate on the functionalities. [Team members had the freedom to innovate and execute their tasks]*	*We have to come up with the solutions and we have to elaborate on the functionalities and one other thing I want to emphasis is when we started the project, we didn't know anything about their operations. For example, you can have a functionality, but you don't know how they currently operate, then the **scope of the functionality becomes very open**. Since we don't know the exact way that they are doing that functionality, **we just try to come up with several alternatives** which were never used by the client* (Respondent 1).	As team members mentioned that they had the freedom to elaborate the functionalities and the ability to prioritize their own tasks, we categorized the level of informal self-controls as **high**.
	Team members could **prioritize their own tasks** as required. [Team members had the freedom to innovate and execute their tasks]*	*I [always] had the list of tasks I should do immediately. I always **priorities** it* (Respondent 1).	
B	The team members willingly spent **additional time** on research and development activities. [Team members used their personal time to complete their tasks]	*Actually, the whole team of the…project is young team. They can spend **additional time** on research and try to do come up with new things* (Respondent 6).	As the team members willingly spent additional time on research and development activities and could innovate and suggest new ideas during the ISD projects, we categorized the level of self-controls as **high**.
	Team members could **innovate and suggest new ideas** as a result of their **past experience** in stock exchange and software as a service model. [Team members had the freedom to innovate and execute their tasks]	*We also knew the stock exchange domain and software as a service model. So, what we did was a result of our **past experience*** (Respondent 5).	
C	As the BRSs only included high-level information about software functionalities, team members could **suggest new ideas**. [Team members had the freedom to innovate and execute their tasks]	*They [team members] have the **freedom** there, as the spec is at high level. So, they actually get quite a freedom there* (Respondent 7).	As the team members had the freedom to elaborate the functionalities and provide alternative methods to cater new client requirements, we categorized the level of informal self-controls as **high**.
	During software development, the team members **suggested** several alternative methods to cater to new client requirements. [Team members had the freedom to innovate and execute their tasks]	*Normally we are **going to give alternative ways** that can handle the new requirements to the client* (Respondent 11).	

Table 4. The Level of Informal Self-controls

D	As the BRSs did not include detailed information about the client requirements, team members had the **freedom to suggest ideas**. [Team members had the freedom to innovate and execute their tasks]	*Sometimes for the documents currently the thing is we are not providing some examples, then we have to **think in several ways*** (Respondent 10).	As the team members had the freedom to suggest ideas and the ability to participate in decision-making processes, we categorized the level of informal self-controls as **high**.
	Project managers encouraged team members to **participate in the decision-making process**. [Mechanisms encouraged and motivated team members to make their own decisions and solutions]	*We get the developers to the audience of that **decision-making process**; we are definitely getting an architect or the Tech lead to that audience as well. Because those are the guys who have the maximum knowledge regarding [the technical requirements of software development]* (Respondent 11).	
E	In situations where the client highlighted the need to amend the software development procedures, team members could think innovatively and **suggest alternative software development procedures**. [Team members had the freedom to innovate and execute their tasks]	*They [client] might suggest that including the field may have an impact on the backward compatibility, **we can do it another way**, use existing fees add some values, change value something like that* (Respondent 15).	As the team members could think innovatively and suggest alternative software development procedures and mechanisms encouraged them to take ownership of the tasks assigned to them, we categorized the level of informal self-controls as **high**.
	When project managers assigned software project components to software engineers, the managers also gave them the ability to take **full responsibility for the component**. [Mechanisms encouraged and motivated team members to make their own decisions and solutions]	*When a developer takes the **responsibility of one component,** he has these responsibilities of the changes in the product document and he should [ensure] how the design is in that component and it is set* (Respondent (13).	
F	Everyone in the team took ownership for the assigned tasks and cared about the software product's reputation, which indicates that team members had the ability to **take responsibility** for the tasks assigned to them. [Team members had the freedom to innovate and execute their tasks]	*All of them were very keen on making sure the system we built and its reputation. Everyone took the **total ownership** of their components and tasks* (Respondent 16).	As the team members had the ability to take the responsibility for the assigned tasks and the freedom to execute their tasks, we categorized the level of informal self-controls as **high**.
	When issues in the project arose, software engineers, business analysts, and consultants had the **flexibility to discuss** the issues with team members. [Team members had the freedom to innovate and execute their tasks]	*So, the **flexibility** [was there] with all the development members, with BAs and consultants* (Respondent 16).	

Table 4. The Level of Informal Self-controls

G	Most of the time, the project's business analysts could **suggest** the methods to cater to client requirements. [Mechanisms encouraged and motivated team members to make their own decisions and solutions]	*Most of the time they would say yea, **this is the best way to do it** and they would come to a conclusion and sometime if they can't, they would consult [the client]* (Respondent 19).	As the team members could suggest and innovate on methods to cater to client requirements, we categorized the level of informal self-controls as **high**.
	The team had a balance between senior and junior staff. While the senior staff had extensive knowledge, the junior staff were tech-savvy and had innovation capabilities. These factors indicate that project managers managed the project in a way that provided team members the **ability to innovate and execute their tasks**. [Team members had the freedom to innovate and execute their tasks]	*Mainly the senior staff has a good knowledge in term of business domain and technical domain knowledge and may exceed the knowledge of BAs [business analysts] sometimes. They have been doing [software development] for more than 10 years. But the others are fairly new and the new people are capable of technical stuff, finding **new things** and like that* (Respondent 19).	
H	When software development issues arose, team members had the ability to collaboratively **find the solutions** for those issues. [Mechanisms encouraged and motivated team members to make their own decisions and solutions]	*So first we engage with the team and **find a solution*** (Respondent 21).	As the team members could collaboratively find the solutions for software development issues and take their own decisions, we categorized the level of informal self-controls **high**.
	Team members had the ability to check the project plan and decide the tasks that they are happy to complete during a day. This indicates that team members had the ability to take **own decisions** about the daily tasks to be completed. [Mechanisms encouraged and motivated team members to make their own decisions and solutions]	*I'm reading the project plan and I'm noting down in another document what are the **task I have to do day**. So, after end of the day, I will check whether I did those things* (Respondent 22).	
* Relevant categorization standards from Appendix F.			

4.5 Establishing Control Enactment based on Control Styles

The two types of control enactment include: 1) the authoritative control style (i.e., unilateral style with limited controller-controllee interaction) and 2) the enabling control style (i.e., bilateral style with frequent interactions between controller and controlees) (Wiener et al., 2019). As project managers (PM) govern the ISD projects, they constitute the controllers, whereas the remaining team members constitute the controllees. The authoritative or enabling control style occurs due to: 1) PMs' characteristics (e.g., their flexibility), 2) team characteristics (e.g., how well team members communicate), and 3) project characteristics (e.g., the project's structure). Thus, we identified the control style in each project by considering those three characteristics (see Appendix F for the categorization standards). For example, in well-structured projects that lack team discussions and communications, project managers can govern the project through authoritative control style whereby they do not have to constantly engage with the team. Moreover, project managers who lack flexibility and strictly follow the project plan indicate that they manage the project with authoritative control style. We show the control style that we identified in each project in Table 5.

Table 5. The Control Styles

Project	PM characteristics	Project characteristics	Team characteristics	Summary
A	The project manager changed during the project. The project managers updated the project plan as needed. For example, one project manager mentioned: "[From] time to time you have to change the plan. I had to take it from previous project manager, [the project plan version I received] was something like version 20. So, the plan has changed more than 20 times.".	The project lacked structure. For example, team members made many updates to the BRSs. As specified fund-processing BRS's revision history, on 5 December, 2011, the business analysts modified 116 specification points, made amendments to 73 points, and clarified a further 29 points. Moreover, on 13 August, 2012, team members updated the fund-processing BRS by: 1) adding or deleting 85 specification points, 2) amending 33 specification points, and 3) clarifying a further 31 specification points.	Team members did not sufficiently communicate with one another and lacked team spirit. For example, respondent 1 mentioned: "I think it was always a game of like passing the ball. I think it is mostly because of the pressure and the stress the team was going through, because we had unrealistic deadlines. Because of the stress, the workload and the pressure the team members are going through, the team spirit was like really less.".	Team members did not sufficiently communicate with one another and lacked team spirit due to time pressures. However, considering the number of BRS updates, the project clearly lacked structure. Thus, comparing the three characteristics (i.e., PMs' characteristics, team characteristics, and project characteristics), we concluded that project managers mainly governed the project through the enabling control style.
B	Only the project managers had access to project plans. When the team leads sent the project updates using Excel sheets, the project managers updated project plans accordingly. Respondent 4 mentioned: "Only the project managers have project plans. [There are] Excel sheets where the team leads [used to] send the update to the project manager. So, PM update the project plan accordingly.".	The project plans frequently changed. For example, respondent 5 mentioned: "Everything didn't happen in the proper, official, standard or expected way. We are just [dealing with] changes all the time.".	Team members communicated with one another often. For example, respondent 6 mentioned: "Even after [sharing] the document, we have a discussion with the developers as well. So, they have the time to go through the document, if there are any clarification required, we have session where we describe the functionality.".	The project lacked structure, and the project managers frequently updated the project plans. Moreover, team members communicated with one another often. Thus, we concluded that project managers governed the project through the control style.
C	The project managers prepared the project plans using Microsoft Project with help from team leads. Thereafter, the project managers verified the project's status on a weekly basis. The project manager (i.e., Respondent 9) mentioned: "Basically what we do is use MS project and plan together and weekly basis track how the process is".	Project managers assigned team members to other projects at the same time. As a result, the team members had to move between projects on a daily basis, which led to time overruns. For example, respondent 8 mentioned: "Currently we work on this project [today] and tomorrow another one.… Usually PM keeps a buffer."	Team members interacted with one another quite often. For example, respondent 8 mentioned: "We...have to interact with the BAs [business analysts] a lot and get clarification and all".	The project managers prepared the project plans in collaboration with the team leads and the project had time overruns. Moreover, team members constantly communicated with one another. Thus, we concluded that project managers governed the project through the enabling control style.

D	The project managers followed structured procedures in managing the project. The project manager (i.e., respondent 12) mentioned: “We have specific procedures to capture the client requirements”.	The project was well structured and stable. For example, respondent 10 mentioned: “It is a very stable project, very less issues come from them [the clients] and we only get most of the time change requests for either regulation change for them or some enhancement kind of a thing”.	The team members interacted with one another well. For example, respondent 10 mentioned: “We have a good interaction”.	Although the team members interacted well, the project managers followed a structured procedure in managing the project. Thus, comparing the three characteristics (i.e., PMs’ characteristics, team characteristics, and project characteristics), we concluded that the project managers governed the project through the authoritative style.
E	The project managers followed structured processes in managing the project. For example, the project manager (i.e., respondent 14) mentioned: “There are certain process people have to follow with regards to fixing issues in new development, meeting timeline, following protocol etc.”.	The project included a 20% to 30% buffer in its timeline to deal with unplanned issues. Most of the time, team members could manage their tasks within the allowed timeframe or within the buffer time period. Respondent 13 mentioned: “We don’t get such difficulty. When we give the estimated time, we add normally 20 to 30 percent buffer for testing. If something goes wrong, we add the buffer. Normally [it is a] problem…stopper.”.	The project managers ensured that team members conversed with one another sufficiently so that they knew about client requirements and implementation procedures. Respondent 14 mentioned: “We make sure the relevant discussions are held before the implementation and internally make sure all the team are in the same [page] BAs [business analysts], QAs [quality assurance engineers] and developers.”	The project managers followed a structured procedure in managing the project and guided the team discussions. Thus, we concluded that the project managers governed the project through the authoritative style.
F	The project managers managed the project by strictly following the contracts and specifications. The project manager (i.e., respondent 17) stated: “The project is governed by the contract.” Respondent 17 continued, “Basically during the implementing phase, the requirement methods [are] based on product specifications”.	Team members could complete the project on time. Respondent 16 stated: “The time we [were] supposed to go live, we have met that [deadline and we] did not have to postpone [the deadline]”.	The team members communicated with other team members about the implementation procedures depending on the criticality and impact of the activities they completed. Respondent 18 stated: “I just have to do this probably depending on the critically and impact of what am I doing, I might consult other people as well”.	As the project managers managed the project following structured guidelines and the team members communicated with one another depending on the criticality and impact of the activities they completed, we concluded that the project managers governed the project through the authoritative style.

G	Several team members strictly monitored the project's progress. Respondent 19 mentioned: "There are lots of people managing [the project]. There is a layer [on] top of us to manage it. So, …our immediate supervisors track each and everyday percentage of work done…and what is not done and maybe they are the people who report to PMs.".	As project managers structured the project well, team members made few updated to the BRSs. Respondent 19 stated: "We didn't have to go and change requirement and so on. Because they [tech leads] involved in the initial stages.".	The software engineers rarely collaborated with the business analysts for requirement changes. For example: respondent 19 stated: "Only few changes we [software engineers] have to go and ask [from the business analysts], can we do this change? Because, in the client [requirement gathering phase] already they [business analysts] have done that.".	As project managers managed the project following structured guidelines and the team members rarely had to request changes to business requirements, we concluded that project managers governed the project through the authoritative style.
H	The project managers worked with team members when they wanted to find the best method to manage the project's evolving situation. The project manager (i.e., respondent 21) stated: "First we engage with the team and find the solution".	Team members frequently updated the project plans. For example, respondent 21 stated: "We have the changes happening in the plan every twice a week because…we have the new changes coming into the plan".	Team members collaborated often when they wanted to identify a solution for an issue. Respondent 21 mentioned: "Full team sit together and talk 'is this the correct way to do it?'".	As team members frequently updated project plans and the project managers collaborated with team members to find the best method to manage evolving situations in the project, we concluded that the project managers governed the project through the authoritative style.

4.6 Establishing the Control Purpose

As per Wiener et al. (2019), the two types of control purpose include: 1) a value-appropriation purpose (harness value) and 2) a value-creation purpose (maximizing value). When a project has a value-appropriation purpose, project managers focus on monitoring and supervising team members to achieve previously identified and agreed goals. In contrast, when a project has a value-creation purpose, team members coordinate in such a way that enables innovation and value creation.

The eight projected we examine in this paper constitute ISD projects. When the projects began, the client organizations and vendors' senior executives and line-of-business managers discussed the project scope and objectives. Once they agreed on the initial parameters and the project scope, they assembled a team with project managers, consultants, business analysts, technology leads, software engineers, system support engineers ,and quality assurance managers that they assigned to the project. The consultants, business analysts, and project managers interacted with the client representatives in order to further specify the requirements, establish project timelines, and develop the project budget. The project management team also interacted with all vendor team members to ensure that they executed the project according to the derived project plan.

For example, the vendor and client organizations involved in project A signed a contract to develop a stock-trading system with 11 modules. The two organizations agreed on the BRS written for each module. These BRSs included detailed information about the project's expected outcomes. We show an example specification review / BRS review schedule document in Table 6.

All eight projects had a final goal to achieve their clients' expected outcomes as per the signed contracts and BRSs. Thus, we concluded that the organizations executed the projects with a value-appropriation purpose (i.e., to harnessing value via achieving the expected and agreed-on outcomes rather than to innovate and maximize outcomes).

Table 6. Example of a Specification Review / BRS Review Schedule Document

Spec / BRS ref	Specification / BRS	Review (client)	Review incorporation (vendor)	Second review (client)	Review incorporation (vendor)	Sign-off (client)
Volume 01	Client registration	Done	Done	17-Mar-11	23/24-Mar-11	28-Mar-11
Volume 03	Trade processing	Done	Done	23-Mar-11	25-Mar-11	29-Mar-11
Volume 06	Brokerages, taxes, and charges	Done	Done	21-Mar-11	28-Mar-11	30-Mar-11
Volume 10	Users and user management	Done	Done	17-Mar-11	23-Mar-11	25-Mar-11
Volume 02	Master data	Done	Done	25-Mar-11	29-Mar-11	31-Mar-11
Volume 04	Fund processing	30-Mar-11	1-Apr-11	5-Apr-11	6-Apr-11	8-Apr-11
Volume 05	Stock processing	31-Mar-11	5-Apr-11	6-Apr-11	7-Apr-11	8-Apr-11
Volume 07	General accounting and journal entries	1-Apr-11	4-Apr-11	7-Apr-11	8-Apr-11	11-Apr-11
Volume 11	Depository participant module	4-Apr-11	5-Apr-11	7-Apr-11	11-Apr-11	12-Apr-11
Volume 02	Master data	5-Apr-11	6-Apr-11	8-Apr-11	11-Apr-11	13-Apr-11
Volume 08	Derivatives (futures and options)	11-Apr-11	15-Apr-11	18-Apr-11	20-Apr-11	22-Apr-11
Volume 09	Initial Public offering and manual funds processing	12-Apr-11	18-Apr-11	19-Apr-11	20-Apr-11	22-Apr-11
Volume 02	Master data	15-Apr-11	19-Apr-11	20-Apr-11	22-Apr-11	25-Apr-11

4.7 Summary of the Results

We summarize our findings in Table 7 (see end of section). The values for the second column (level of interdependencies) come from Table 1, for the third and fourth columns (level of formal outcome controls / formal behavior controls) from Table 2, for the fifth and sixth columns (level of informal clan controls / informal self-controls) from Tables 3 and 4, and the seventh column (control enactment) from Table 5. We discuss the findings in more detail in Sections 4.7.1 to 4.7.3.

4.7.1 The Link between Interdependencies and Formal Outcome and Formal Behavior Controls

As projects E, F and G had a low level of interdependencies between intra-project modules, project managers governed them with a high level of outcome and behavior controls. Although projects C and H had a low level of interdependencies between intra-project modules, other factors such as flexible project practices and volatile client requirements that caused fewer outcome and behavior controls in those projects. The BRS in project C included only high-level information about the intra-project module interdependencies and the project requirements. Moreover, the business analysts updated the BRSs with information such as suggestions from the software engineers. These findings indicate that project C followed a flexible project practice in that software engineers had the ability to provide suggestions to enhance the BRSs. In project H, the business analysts requested software code changes due to volatile client requirements. Thus, we can conclude that project managers have a tendency to govern projects with a low level of intra-project module interdependencies with a high level of outcome and behavior controls. However, it will be challenging to govern the projects with a low level of intra-project module interdependencies with a high level of outcome and behavior controls when the projects have flexible project practices and volatile client requirements.

Since projects A, B and D had a high level of interdependencies between intra-project modules, they had a low level of outcome and behavior controls. While team members made frequent updates to project A's BRSs, they updated the software designs in project B. Since the business analysts could not identify the intra-project module interdependencies accurately, project D's BRSs lacked information about the module interdependencies. Thus, we can conclude that project managers govern projects with a low level of intra-project module interdependencies with a low level of outcome and behavior controls.

4.7.2 The Link between Interdependencies and Informal Clan and Informal Self-controls

As the projects B and D had a high level of intra-project module interdependencies, project managers governed them through a high level of informal clan controls. Although project A had a high level of intra-project module interdependencies, it had a low level of informal clan controls due to issues such as time pressure. Thus, we can conclude that although project managers have a tendency to govern the projects with a high level of intra-project module interdependencies with a high level of clan controls, project managers will find project governance difficult when team members face time pressures.

Although projects C, E, F, and H had a low level of intra-project module interdependencies, project managers governed them with a high level of informal clan control because these projects' team members had to collaborate with the business analysts in order to obtain clarifications about the client requirements and the module interdependencies. In contrast, as the software engineering team participated in the modularization process in project G, they understood the client's requirements and interdependencies well. Therefore, the software engineers did not need to collaborate with the business analysts that much. Thus, we can conclude that project managers can govern projects with a low level of intra-project module interdependencies with a low level of informal clan controls when all the teams (e.g., business analyst team, software engineering team, and quality assurance team) participate in the modularization process early in the project lifecycle.

Regardless of the level of intra-project module interdependencies between modules, all eight projects had a high level of informal self-controls. Thus, we can conclude that the level of intra-project module interdependencies does not have an impact on the level of informal self-controls.

4.7.3 The Link between Interdependencies, Control Enactment and Control Purpose

Project managers governed projects A and B, which had a high level of intra-project module interdependencies, with an enabling style as both projects lacked adequate structure. However, project managers governed project D with an authoritative style although the project had a high level of interdependencies because it lacked structure. Thus, we can conclude that project managers govern projects with a high level of intra-project module interdependencies with an enabling control style when they lack the adequate structure.

Project managers governed the projects with a low level of intra-project module interdependencies (i.e., projects E, F, G, and H) with an authoritative control style as those projects lacked structure. Although project C had a low level of intra-project module interdependencies, project managers governed the project with an enabling control style as the project's team members worked on multiple projects at the same time. Therefore, the project managers needed to interact frequently with the team and update project plans as required. Thus, we can conclude that project managers can govern projects with a low level of intra-project module interdependencies with an authoritative control style except in the projects where they assign team members to multiple projects simultaneously.

As all the projects constituted ISD projects, they all had a value-appropriation purpose rather than value-creation purpose. Thus, we could not establish a link between the level of intra-project module interdependencies and control purpose.

Table 7. Summary of Findings

Project	Level of interdependencies	Control configuration				Control enactment	Control purpose
		Level of formal outcome controls	L.O formal behavior controls	Level of informal clan controls	Level of informal self-controls		
A	High	Low	Low	Low	High	Enabling	Value appropriation
B	High	Low	Low	High	High	Enabling	Value appropriation
C	Low	Low	Low	High	High	Enabling	Value appropriation
D	High	Low	Low	High	High	Authoritative	Value appropriation
E	Low	High	High	High	High	Authoritative	Value appropriation
F	Low	High	High	High	High	Authoritative	Value appropriation
G	Low	High	High	Low	High	Authoritative	Value appropriation
H	Low	Low	Low	High	High	Authoritative	Value appropriation

5 Implications, Limitations, and Future Research

5.1 Implications for Research

Providing a theoretical extension to control theory, we highlight the need to consider interdependencies between intra-project modules when managing ISD projects. Kirsch (2004) points out that ISD projects' different stages may include task interdependencies, which can lead to changes in the control mechanisms that project managers choose. According to our findings, when projects had a low level of interdependencies between intra-project modules, project managers tended to use formal outcome and formal behavior control mechanisms to govern team members. Although some projects had a low level of interdependencies between intra-project modules, other factors such as project practices and volatile client requirements may minimize the level of formal outcome and formal behavior controls in projects. Our findings highlight that project managers governed the projects with a high level of intra-project module interdependencies with enabling control styles when the projects lacked structure. Project managers can govern projects with a low level of intra-project module interdependencies with authoritative control styles except in the projects where they assign team members to multiple projects simultaneously. This finding extends the previous findings on relationships between module interdependencies and project control (Kirsch, 2004; Ouchi & Maguire, 1975). Researchers have mainly used control theory to examine the governance between a client and a vendor in ISD outsourcing projects (e.g., see Tiwana, 2008; Rustagi et al., 2008). We extend control theory by explaining how one can apply it to examine how project managers govern intra-project module interdependencies in ISD projects. Moreover, little research has applied recent concepts related to control (e.g., control styles and control purpose) to understand how to govern projects (Wiener et al., 2019). We extend control theory by applying control styles and control purpose to real-world scenarios.

Kirsch (1996) recommends organizations use clan control in order to increase individuals' commitment to teams. Although the control literature discusses clan control, it does not discuss governing multiple clans in a single ISD project. Our results indicate that ISD projects comprise several teams with different team goals and objectives, which creates multiple clans in a single project. Moreover, these teams work on different intra-project modules. Therefore, when the projects include intra-project module interdependencies, project managers should ensure that they appropriately implement clan control in teams (e.g., in the business analyst team and in the software engineering team) and between teams that they assign to different modules (e.g., between the business analyst team and the software engineering team). As expected, in the projects we examined, project managers scarcely used informal clan control in projects due to low interdependencies between intra-project modules. Even though some projects had low interdependencies between intra-project modules, the projects required clan control in order to gain sufficient information about client requirements.

5.2 Implications for Practice

Our results have the potential to influence ISD project governance in practice. We found that, when projects had a low level of interdependencies between intra-project modules, project managers tended to use formal outcome and formal behavior controls to govern team members. Although projects had a low level of interdependencies between intra-project modules, project managers required informal clan control to ensure that team members understood the client requirements accurately. Project managers could govern projects with a high level of intra-project module interdependencies with the enabling control style when the projects lack structure. They can govern projects with a low level of intra-project module interdependencies with the authoritative control style except in projects in which they assign team members to multiple projects simultaneously. Therefore, project managers should properly understand control mechanisms and control styles to govern modularized ISD projects with varying degrees of intra-project module interdependencies. Because project managers assigned team members to different modules in some cases, the projects had multiple clans. However, interdependencies between intra-project modules and unanticipated situations in projects such as schedule overruns can occur. Thus, project managers should ensure all project teams work collaboratively as a single team.

5.3 Limitations and Future Research

Though this study makes several theoretical and practical contributions, it has some limitations that we need to acknowledge. First, although we followed procedures to ensure internal and external validity, our subjectivity in data analysis might lead to subjective results. Second, the study's sample comprised eight projects selected from a single ISD organization, which might preclude generalizability. Thus, other studies should use samples from multiple organizations in order to enrich the insights we obtained about the effective control mechanisms for modularized ISD projects. Furthermore, a longitudinal study on modularized ISD project control could yield further insights about the phenomena observed.

6 Conclusions

In this study, we examine the control mechanisms that project managers use to govern modularized ISD projects with varying degrees of intra-project module interdependencies. We selected control theory as the situating theoretical lens as it helps one understand and resolve in complex ISD projects. We identified that a low level of interdependencies between intra-project modules lead to a high level of formal outcome and formal behavior controls in ISD projects. Although some projects include a low level of interdependencies between intra-project modules, project practices and volatile client requirements may minimize the level of formal outcome and formal behavior controls in such projects. Even though a low level of interdependencies between intra-project modules minimize the need for informal clan control, project managers may need to implement such a control mechanism to ensure team members understand the project requirements. Projects with a high level of interdependencies between intra-project modules may have a high level of informal clan controls. We found some situations where the projects with a high level of interdependencies between intra-project modules had a low level of informal clan control, which occurred due to time pressures. Project managers may govern ISD projects with a high level of intra-project module interdependencies with an enabling control style when the projects lack structure. Project managers can effectively govern projects with a low level of intra-project module interdependencies though an authoritative control style except in projects where they assign team members to multiple projects simultaneously.

References

Abubakre, M. A., Ravishankar, M. N., & Coombs, C. R. (2015). The role of formal controls in facilitating information system diffusion. *Information & Management, 52*(5), 599-609.

Al-Otaiby, T. N., AlSherif, M., & Bond, W. P. (2005). Toward software requirements modularization using hierarchical clustering techniques. In *Proceedings of the meeting of the Southeast Regional Conference.*

Allen, E. B., & Khoshgoftaar, T. M. (1999). Measuring coupling and cohesion: An information-theory approach. In *Proceedings of the meeting of the Sixth International Software Metrics Symposium.*

Andres, H. P. (2002). A comparison of face-to-face and virtual software development teams. *Team Performance Management, 8*(1/2), 39-48.

Baghizadeh, Z., Cecez-Kecmanovic, D., & Schlagwein, D. (2019). Review and critique of the information systems development project failure literature: An argument for exploring information systems development project distress. *Journal of Information Technology*, 35(2), 123-142.

Benbasat, I., Goldstein, D. K., & Mead, M. (1987). The case research strategy in studies of information systems. *MIS Quarterly, 11*(3), 369-386.

Cai, Y., & Huynh, S. (2007). An evolution model for software modularity assessment. In *Proceedings of the 5th International Workshop on Software Quality.*

Capretz, L. F., & Ahmed, F. (2010). Making sense of software development and personality types. *IT Professional Magazine, 12*(1), 6-13.

Cataldo, M. (2007). *Dependencies in geographically distributed software development: Overcoming the limits of modularity*. Pittsburgh, PA: Carnegie Mellon University.

Cataldo, M., Bass, M., Herbsleb, J. D., & Bass, L. (2007). On coordination mechanisms in global software development. In *Proceedings of the IEEE International Conference on Global Software Engineering.*

Cataldo, M., Herbsleb, J. D., & Carley, K. M. (2008). Socio-technical congruence: A framework for assessing the impact of technical and work dependencies on software development productivity. In *Proceedings of the ACM-IEEE International Symposium on Empirical Software Engineering and Measurement.*

Cataldo, M., & Nambiar, S. (2012). The impact of geographic distribution and the nature of technical coupling on the quality of global software development projects. *Journal of Software: Evolution and Process, 24*(2), 153-168.

Choudhury, V., & Sabherwal, R. (2003). Portfolios of control in outsourced software development projects. *Information Systems Research, 14*(3), 291-314.

Conley, C. A., & Sproull, L. (2009). Easier said than done: An empirical investigation of software design and quality in open source software development. In *Proceedings of the Hawaii International Conference on System Sciences.*

Cram, A., Brohman, K., Chan, Y., & Gallupe, B. (2016a). Information systems control alignment: Complementary and conflicting systems development controls. *Information & Management, 53*(2), 183-196.

Cram, A., Brohman, K., & Gallupe, B. (2016b). Hitting a moving target: A process model of information systems control change. *Information Systems Journal, 26*(3), 195-226.

Cram, A., Brohman, K., & Gallupe, B. (2016c). Information systems control: A review and framework for emerging Information Systems processes. *Journal of the Association for Information Systems, 17*(4), 216-266.

Cram, A., & Brohman, M. K. (2013). Controlling information systems development: A new typology for an evolving field. *Information Systems Journal, 23*(2), 137-154.

Doody, O., & Noonan, M. (2013). Preparing and conducting interviews to collect data. *Nurse Researcher, 20*(5), 1-10.

Eisenhardt, K. M. (1985). Control: Organizational and economic approaches. *Management Science, 31*(2), 134-149.

Gershenson, J., Prasad, G., & Zhang, Y. (2003). Product modularity: Definitions and benefits. *Journal of Engineering Design, 14*(3), 295-313.

Ghobadi, S. (2015). What drives knowledge sharing in software development teams: A literature review and classification framework. *Information & Management, 52*(1), 82-97.

Ghobadi, S., & D'Ambra, J. (2013). Modeling high-quality knowledge sharing in cross-functional software development teams. *Information Processing & Management, 49*(1), 138-157.

Goldbach, T., Benlian, A., & Buxmann, P. (2018). Differential effects of formal and self-control in mobile platform ecosystems: Multi-method findings on third-party developers' continuance intentions and application quality. *Information & Management, 55*(3), 271-284.

Gomes, P. J., & Joglekar, N. R. (2008). Linking modularity with problem solving and coordination efforts. *Managerial and Decision Economics, 29*(5), 443-457.

Gopal, A., & Gosain, S. (2010). The role of organizational controls and boundary spanning in software development outsourcing: Implications for project performance. *Information Systems Research, 21*(4), 960-982.

Goulão, M. (2001). Coupling and cohesion as modularization drivers: Are we being over-persuaded? In *Proceedings of the 5th European Conference on Software Maintenance and Reengineering.*

Henderson, J. C., & Lee, S. (1992). Managing I/S design teams: A control theories perspective. *Management Science, 38*(6), 757-777.

Hitz, M., & Montazeri, B. (1995). Measuring coupling and cohesion in object-oriented systems. In *Proceedings of the International Symposium on Applied Corporate Computing.*

Hoegl, M., Parboteeah, P. K., & Gemuenden, H. G. (2003). When teamwork really matters: Task innovativeness as a moderator of the teamwork-performance relationship in software development projects. *Journal of Engineering and Technology Management, 20*(4), 281-302.

Hofmann, H. F., & Lehner, F. (2001). Requirements engineering as a success factor in software projects. *IEEE Software, 18*(4), 58-66.

Holmqvist, T., & Persson, M. (2004). Modularization-not only a product issue. In *Proceedings of the 7th Workshop on Product Structuring–Product Platform Development.*

Jaworski, B. J. (1988). Toward a theory of marketing control: Environmental context, control types, and consequences. *Journal of Marketing, 52*(3), 23-39.

Jaworski, B. J., & MacInnis, D. J. (1989). Marketing jobs and management controls: Toward a framework. *Journal of Marketing Research, 26*(4), 406-419.

Kim, S. K., & Tiwana, A. (2016). Chicken or egg? Sequential complementarity among salesforce control mechanisms. *Journal of the Academy of Marketing Science, 44*(3), 316-333.

Kirsch, L. J. (1996). The management of complex tasks in organizations: Controlling the systems development process. *Organization Science, 7*(1), 1-21.

Kirsch, L. J. (1997). Portfolios of control modes and IS project management. *Information Systems Research, 8*(3), 215-239.

Kirsch, L. J. (2004). Deploying common systems globally: The dynamics of control. *Information Systems Research, 15*(4), 374-395.

Kirsch, L. J., Sambamurthy, V., Dong-Gil, K., & Purvis, R. L. (2002). Controlling information systems development projects: The view from the client. *Management Science, 48*(4), 484-498.

Kraut, R. E., & Streeter, L. A. (1995). Coordination in software development. *Communications of the ACM, 38*(3), 69-81.

Kumar, N., Stern, L. W., & Anderson, J. C. (1993). Conducting interorganizational research using key informants. *Academy of Management Journal, 36*(6), 1633-1651.

Kwong, C. K., Mu, L. F., Tang, J. F., & Luo, X. G. (2010). Optimization of software components selection for component-based software system development. *Computers & Industrial Engineering, 58*(4), 618-624.

Levesque, L. L., Wilson, J. M., & Wholey, D. R. (2001). Cognitive divergence and shared mental models in software development project teams. *Journal of Organizational Behavior, 22*(2), 135-144.

Lindberg, A., Berente, N., Gaskin, J., & Lyytinen, K. (2016). Coordinating interdependencies in online communities: A study of an open source software project. *Information Systems Research, 27*(4), 751-772.

MacCormack, A., Rusnak, J., & Baldwin, C. Y. (2007). The impact of component modularity on design evolution: Evidence from the software industry. *Harvard Business School Technology & Operations Management*. Retrieved from https://ssrn.com/abstract=1071720

Mani, D., Srikanth, K., & Bharadwaj, A. (2014). Efficacy of R&D work in offshore captive centers: An empirical study of task characteristics, coordination mechanisms, and performance. *Information Systems Research, 25*(4), 846-864.

Manz, C. C., & Angle, H. (1986). Can group self-management mean a loss of personal control: Triangulating a paradox. *Group & Organization Studies 11*(4), 309-339.

Mao, J.-Y., Lee, J.-N., & Deng, C.-P. (2008). Vendors' perspectives on trust and control in offshore information systems outsourcing. *Information & Management, 45*(7), 482-492.

Maruping, L. M., Venkatesh, V., & Agarwal, R. (2009). A control theory perspective on agile methodology use and changing user requirements. *Information Systems Research, 20*(3), 377-399.

Mikkola, J. H., & Skjøtt-Larsen, T. (2004). Supply-chain integration: Implications for mass customization, modularization and postponement strategies. *Production Planning & Control, 15*(4), 352-361.

Minichiello, V., Aroni, R., Timewell, E., & Alexander, L. (1995). *In-depth interviewing: Principles, techniques, analysis* (2nd ed.). Melbourne, Australia: Pearson.

Mirani, R. (2007). Procedural coordination and offshored software tasks: Lessons from two case studies. *Information & Management, 44*(2), 216-230.

Narduzzo, A., & Rossi, A. (2003). *Modular design and the development of complex artifacts: Lessons from free/open source software Quaderno* (Vol. 80). Trento, Italy: University of Trento.

Nidumolu, S. R., & Subramani, M. R. (2003). The matrix of control: Combining process and structure approaches to managing software development. *Journal of Management Information Systems, 20*(3), 159-196.

Nuwangi, S. M., Sedera, D., Srivastava, S. C., & Murphy, G. (2013). Intra-organizational information asymmetry in offshore ISD outsourcing. *VINE: The Journal of Information and Knowledge Management Systems, 44*(1), 94-120.

Ouchi, W. G. (1978). The transmission of control through organizational hierarchy. *Academy of Management Journal, 21*(2), 173-192.

Ouchi, W. G. (1979). A conceptual framework for the design of organizational control mechanisms. *Management Science, 25*(9), 833-848.

Ouchi, W. G. (1980). Markets, bureaucracies, and clans. *Administrative Science Quarterly, 25*(1), 129-141.

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

Patton, M. Q. (2002). *Qualitative research and evaluation methods*. Thousand Oaks, CA: Sage.

Rao, M. T., Brown, C. V., & Perkins, W. C. (2007). Host country resource availability and Information System control mechanisms in multinational corporations: An empirical test of resource dependence theory. *Journal of Management Information Systems, 23*(4), 11-28.

Ravishankar, M. N., & Pan, S. L. (2013). Examining the influence of modularity and knowledge management (KM) on dynamic capabilities: Insights from a call center. *International Journal of Information Management, 33*(1), 147-159.

Remus, U., & Wiener, M. (2012). The amount of control in offshore software development projects. *Journal of Global Information Management, 20*(4), 1-26.

Rustagi, S., King, W. R., & Kirsch, L. J. (2008). Predictors of formal control usage in IT outsourcing partnerships. *Information Systems Research, 19*(2), 126-143.

Sanchez, R. (1995). Strategic flexibility in product competition. *Strategic Management Journal, 16*, 135-159.

Sanchez, R., & Mahoney, J. T. (1996). Modularity, flexibility, and knowledge management in product and organization design. *Strategic Management Journal, 17*, 63-76.

Sarker, S., & Sahay, S. (2003). Understanding virtual team development: An interpretive study. *Journal of the Association for Information Systems, 4*(1), 1-36.

Sethi, K., Yuanfang, C., Wong, S., Garcia, A., & Sant'Anna, C. (2009). From retrospect to prospect: Assessing modularity and stability from software architecture. In *Proceedings of the Joint Working IEEE/IFIP Conference on Software Architecture 2009 & European Conference on Software Architecture.*

Snell, S. A. (1992). Control theory in strategic human resource management: The mediating effect of administrative information. *Academy of Management Journal, 35*(2), 292-327.

Sosa, M. E., Eppinger, S. D., & Rowles, C. M. (2004). The misalignment of product architecture and organizational structure in complex product development. *Management Science, 50*(12), 1674-1689.

Srivastava, S. C., & Shainesh, G. (2015). Bridging the service divide through digitally enabled service innovations: Evidence from Indian healthcare service providers. *MIS Quarterly, 39*(1), 245-267.

Taube-Schock, C., Walker, R. J., & Witten, I. H. (2011). Can we avoid high coupling? In *Proceedings of the 25th European Conference on Object-oriented Programming.*

Thomas, D. R. (2006). A general inductive approach for analyzing qualitative evaluation data. *American Journal of Evaluation, 27*(2), 237-246.

Tiwana, A. (2008). Does technological modularity substitute for control? A study of alliance performance in software outsourcing. *Strategic Management Journal, 29*(7), 769-780.

Tiwana, A. (2015). Evolutionary competition in platform ecosystems. *Information Systems Research, 26*(2), 266-281.

Wiener, M., Mähring, M., Remus, U., & Saunders, C. (2016). Control configuration and control enactment in Information Systems projects: Review and expanded theoretical framework. *MIS Quarterly, 40*(3), 741-774.

Wiener, M., Mähring, M., Remus, U., Saunders, C., & Cram, W. A. (2019). Moving IS project control research into the digital era: The “why” of control and the concept of control purpose. *Information Systems Research, 30*(4), 1387-1401.

Wiener, M., Remus, U., Heumann, J., & Mähring, M. (2015). The effective promotion of informal control in information systems offshoring projects. *European Journal of Information Systems, 24*(6), 569-587.

Yin, R. K. (2003). *Case study research: Design and methods* (3rd ed.). Thousand Oaks, CA: Sage.

Zimmermann, A., & Ravishankar, M. (2014). Knowledge transfer in IT offshoring relationships: The roles of social capital, efficacy and outcome expectations. *Information Systems Journal, 24*(2), 167-202.

Appendix A: Overview of Existing Research

Table A1. Overview of Existing Research that Integrates Interdependencies, Modularization, Project Control, and Governance in the ISD Project Context

Study	Context	Method	Theory	Focus	Main findings
Tiwana (2015)	Platform markets	Quantitative method Data collection in three phases over five years (2009 to 2013) from 342 independent extension developers	Control theory	The interplay between control and extension modularization	Complementarity between input controls and extension modularization increases project performance.
Tiwana (2008)	Software outsourcing projects	Quantitative method Data from 120 software outsourcing alliances	Control theory	To identify whether technological modularity substitutes for control	Process control, outcome control, and modularity independently enhance alliance performance. Modularity and control are imperfect substitutes; modularity minimizes the effect that process control has on alliance performance but does not minimize the effect that outcome control has on performance.
Cataldo (2007)	Geographically distributed software development projects	Quantitative method Four stage study using software source code files and modification requests	Socio-technical congruence	To identify the work dependencies and to investigate the impact of work dependencies on the development process	Defines a method to measure socio-technical congruence (i.e., the relationship between work dependencies and coordination patterns).
Lindberg, Berente, Gaskin, & Lyytinen (2016)	Open source software projects	Qualitative method Using the available data within the repository	Coordination	To explore how open source software project manages unresolved development interdependencies	Explained how development and developer interdependencies are associated with different routines (i.e., activity sequences).
Cataldo, Herbsleb, & Carley (2008)	Software development projects	Quantitative method Data from the first four releases of a large distributed system development project	Socio-technical congruence	To explain the impact of technical and work dependencies on software development productivity	Identifying an accurate set of product dependencies determines the relevant work dependencies and, thus, ultimately minimizes the time to resolve software modification requests.
Gomes & Joglekar (2008)	Software development projects	Quantitative method 71 tasks carried out in 11 software development projects	Information processing view of product development and transaction cost economics	To identify the effect of task modularity on transactional efficiency and software project completion time	Task modularity increases the transactional efficacy. Increase in modularity decreases the development time and coordination effort.

Table A1. Overview of Existing Research that Integrates Interdependencies, Modularization, Project Control, and Governance in the ISD Project Context

Conley & Sproull (2009)	Open source software development projects	Quantitative method Data collected from 203 software releases in 46 open source projects hosted on SourceForge.net	Software modularity	To identify the relationship between software design modularity and software quality	Software modularity reduces the software complexity, increases the number of static software bugs, and has a mixed relationship with the percentage of software bugs closed.
Narduzzo & Rossi (2003)	Open source software projects	Qualitative method Case studies and indirect observations	Software modularity	To identify the relationships between organizational structure and architecture design in modular projects and to explore how the modularity copes with unexpected interdependencies	Increasing the degree of modularity and violating the information hiding principle are important in open source software projects.

Appendix B: Project Descriptions

Table B1. Projects

Project name	Project description
A	The project developed a post-trade application that provided clearing and settlement for trades after execution. The software application's functionalities included trade processing, user management, general accounting, and journal entries. Moreover, the software application comprised complex trade processing methods that integrated highly with clearing and settlement procedures. The client resided in in the Asian region.
B	The project developed a real-time clearing system that managed post-trade activities. This project, which the latest technology enriched, provided successful solutions to overcome the limitations that current traditional clearing systems face. This project comprised three major functionalities: 1) clearing (matching, recording, and transaction processing), 2) settlement (trade settlement), and 3) risk management (managing the risks of market participants). The client company resided in in Europe.
C	This project addresses a wide range of business needs including connectivity and order management. It can handle a variety of firms' trading business; covering front office, middle office and back office functionality. Key characteristics of the information system solution include: connectivity hub and post-trade risk management. The project comprised around 40 clients from all over the world.
D	This project developed a tested, real-time, and transparent trading platform. Furthermore, it facilitated robust assaying and warehousing facilities to execute trades. The software solution include features such as hedging; risk management, and clearing and settlement. The client resided in in the Asian region.
E	This project developed tools to detect irregular trading behaviors. The system's key functionalities included analyzing real-time/offline transaction data, providing alerts, and managing cases. This project comprised several clients including clients from Asian and African regions.
F	The project integrated several trading platforms. It provides smooth transition for trading systems. This project developed a platform for a client that began operation in the 1980s. This client provided capital market solutions for several instrument trading.
G	This project developed a post-trade application. The post-trade application included clearing and settling executed trades. The application had several users with different authorization levels; the registry owner constituted the main user. The system had different users (e.g., brokers and custodian banks). The client company resided in the Asian region.
H	This project developed a highly critical application as it is a client-focused solution and integrates several modules. One could adjust the application to trade any product in any type of market. It displayed reliability and flexibility and met the clients' specific needs. This project comprised multiple clients from all over the world.

Appendix C: Participant Information

ISD projects generally comprise a business analyst team, software engineering team, quality assurance team and a project management team. While the business analysts document the BRS as per the client requirements, the software engineering team develops ISD solutions according to the BRS. The software quality assurance team tests the ISD solutions to identify non-compliance issues. The project management team ensures that team members execute the project according to the project plan. We interviewed a project manager, a team member from the business analyst team (business analyst, senior business analyst, consultant or senior consultant) and a team member from the software engineering team (software engineer, senior software engineer or technical lead). Informant 2 was the project manager for both Project A and Project G.

Table C1. Participant Information

Project	Participant ID	Designation	Years of experience in the ISD industry	Years of experience in the company
A	01	Senior business analyst	4	4
	02	Project manager	15	2
	03	Specialist software engineer	8	8
B	04	Director business operations (post-trade)	11	10
	05	Senior tech lead	7	7
	06	Senior business analyst	4	4
C	07	Business analyst	4	3
	08	Principal software engineer	5	4
	09	Junior project manager	4	3
D	10	Business analyst	8	8
	11	Technical lead	9	9
	12	Associate project manager	10	10
E	13	Senior software engineer	8	8
	14	Junior project manager	3	3
	15	Senior business analyst	3*	3*
F	16	Technical lead	12	9
	17	Project manager—level II	9	9
	18	Senior business analyst	6	4
G	19	Specialist software engineer	8	8
	20	Consultant	12	10
	02**	Project manager	15	2
H	21	Project manager—level I	6*	5*
	22	Senior software engineer	3*	3*
	23	Senior business analyst	3*	3*

* Verified through LinkedIn
** Informant 2 was the project manager for both Project A and Project G.

Appendix D: Interview Questions

1) Can you please describe project X[1]?
2) What sort of issues do you encounter in your project?
3) Can you describe the documents and software systems that your team uses to transfer the client requirements?
4) To what level do you document the client requirements?
5) Are there any other documents and software systems that your team uses as contracts between client and you?
6) Can you please describe the penalties, rewards and time allocations of your project?
7) Can you discuss the issues you face when you are controlling a project?
8) What knowledge is required for your team members to develop products that satisfy client requirements?
9) What knowledge do team members have about the contracts and requirement documents?
10) To what level do you follow the document during day-to-day activities?
11) How do you describe the behavior of your project team members? Are they flexible to provide more information than what is mentioned in the requirement documents?
12) Can you please describe the team spirit, shared values and beliefs of the team?
13) To what extent do you amend the requirement documents, time estimations and project templates according to the requests from the team members?
14) How does your team manage day-to-day operations and quick decisions?

[1] To maintain confidentiality, we disguise the projects' names.

Appendix E: Validity and Reliability

Table E1. Validity and Reliability Tests

Adapted from Yin (2003)			
Tests	**Description**	**Case study tactic**	**How we applied the case study tactic in our study**
Construct validity	Identify correct operational measures for the concepts	Collect evidence from multiple sources	We collected data from: interviews, official websites, and different documents (BRSs, test scenarios, test cases, and design documents).
		Have key informants review our draft reports	We asked key participants and peers to review our draft case study reports.
		Maintain an evidence chain	We maintained the evidence chain for the whole case study (including case study report and case study questions).
Internal validity	Establish causal relationship	Pattern matching	We followed the pattern-matching analysis technique to analyze data.
External validity	Identify whether one can generalize the findings	Use replication logic for the multiple case studies	We used replication logic to analyze data.
Reliability	Demonstrate that one can repeat the study's operations and achieve the same results	Use case study protocol	We designed case study protocol, which included describing the case study, data-collection procedures, case study questions, and the sample codes. We designed the interview protocol to broadly understand the phenomenon.
		Use case study database	We collected and organized data from various resources such as official websites and documents including BRSs and design specifications.
Biased results	Interview data can be subjective; therefore, obtaining results from one person can generate inaccurate and bias results	Interview many participants from each project	We interviewed three participants, (i.e., one participant from project management team, one participant from business analyst team, and one participant from software engineering team) from each project. We gained results by comparing and contrasting all three members' opinions, which also minimized the bias in the results.

Appendix F: Categorization Standards

Table F1. Categorization Standards

Level of interdependencies: high
Project comprised modules that had a significant impact on other modules.
Team members required significant input from other modules to complete their tasks.
Business Analysts could not subdivide functionalities of the IS solution into modules.
A change or update in one module had a significant impact on other modules.
Business Analysts did not identify the modules that had an impact on other modules.
Level of interdependencies: low
Business Analysts subdivided the IS solution into several modules.
The IS solution comprised modules that had fewer impact on other modules.
A change or update in one module impacted the other modules less.
Teams worked on different modules.
Teams worked on isolated tasks that they had to complete independently.
Interfaces connected different modules; when teams had to integrate a new module to existing modules, they had to consider the interfaces that connected the new module with the existing modules.
Formal outcome control: high
The BRSs specified the expected outcomes in detail
Business Analysts made few updates to or did not update the BRSs at all, which indicated stable expected outcomes.
The developed information system solution aligned with the information specified in BRSs
Team members had to strictly follow the BRSs and provide the exact requirements that the BRSs specified.
The project had deadlines to ensure that team members achieved the expected outcomes on time.
Formal outcome control: low
The BRSs did not properly specify the expected outcomes.
The developed information system solution did not completely align with the BRSs.
Business Analysts updated the BRSs several times, which indicated unstable expected outcomes.
The team members did not have to strictly follow the BRSs.
The project did not have deadlines to ensure that team members achieved the expected outcomes on time.
Formal behavior control: high
Team members made few or no updates to the software design and software code (i.e., the procedures that team members should follow to achieve the expected outcomes). Lack of updates to the software design and code indicated stable the formal behavior controls.
The project comprised detailed project plans
Project managers did not frequently update project plans
Project managers conducted project reviews and project meetings to ensure that team members executed the project according to the project plan.
Project managers closely monitored team members’ progress
Formal behavior control: low
Team members made several updates to the project plans, software design, and software code
The project did not comprise detailed project plans
Team members did not have regular project reviews and project meetings
Project managers did not closely monitor team members’ progress
Informal clan control: high
The team shared the same values and beliefs
The team displayed high team spirit

Table F1. Categorization Standards

Team members had good interactions, relationships, and collaborations
Team members provided suggestions or clarifications to other team members
When some team members required support to complete the assigned tasks, other team members were willing to provide the required support.
Informal clan control: low
The team displayed low team spirit
Team members did not interact frequently
Team members did not collaborate often
Team members had weak relationships
Team members did not provide suggestions or clarifications for other team members' tasks
When a team member required support to complete the assigned tasks, other team members were not willing to provide the required support.
Informal self-control: high
Mechanisms encouraged and motivated team members to make their own decisions and solutions
Team members used their personal time to complete their tasks
Team members had the freedom to innovate and execute their tasks
Informal self-control: low
No/few mechanisms encouraged and motivated team members to make their own decisions and solutions
Team members did not use their personal time to complete their tasks
Team members did not have the freedom / lacked the freedom to innovate and execute their tasks
Control enactment: unilateral
Limited interactions between a controller (i.e., project manager) and controlee (i.e., team members)
PMs' characteristics: the project managers were not flexible / they did not frequently update the project plans
Project characteristics: project managers and the team leaders structured the project well and, thus, allowed fewer updates (we identified a well-structured project when BRS had fewer updates / no design changes / team members did not receive several unplanned tasks / team members did not have to frequently update the tasks / Project managers did not change or disturb team members' tasks / teams completed the project on time as planned)
Team characteristics: team members communicate with one another less frequently
Control enactment: bilateral
Frequent interactions between a controller (i.e., project manager) and controlee (i.e., team members)
PMs' characteristics: the project managers were flexible / they frequently updated the project plans
Project characteristics: project manager did not structure the project well and, thus, allowed many updates (we identified a poorly structured project when BRS had many updates / several design changes / team members received several unplanned tasks / team members had to frequently update the tasks / project managers changed or disturbed team members' planned tasks / teams completed the project on time as planned)
Team characteristics: team members communicated with one another more frequently

About the Authors

Maduka Subasinghage is a lecturer at Auckland University of Technology, New Zealand. She completed her PhD studies at the Queensland University of Technology, Australia. Her research interests include knowledge management, governance, ISD and outsourcing. Her research has been published in *Communications of the AIS*, *Journal of Knowledge Management*, *Enterprise Information Systems Journal*, *VINE: The Journal of Information and Knowledge Management Systems,* and in the proceedings of international conferences such as International Conference on Information Systems (ICIS) and European Conference on Information Systems (ECIS).

Darshana Sedera is a Professor of Information Systems and serves as the Associate Dean of Research at the Southern Cross University's Faculty of Business, Arts and Law. He has published his research in the information systems field in over 200 publications in major refereed journals and conferences. He researches of the role of technology in innovation, entrepreneurship and business transformation. His publications have appeared in *Journal of the AIS*, *Journal of the Strategic Information Systems*, *Information and Management*, and *Communications of the AIS*.

Shirish C. Srivastava is Professor and GS1 France Chair in "Digital Content for Omni Channel" at HEC Paris. Prior to joining HEC, he has lectured at the School of Business, National University of Singapore and holds a Ph.D. from the same university. His research interests include e-government, services sourcing, technology enabled innovation, artificial intelligence, opensource, and social media strategy. His research has been published in several top-tier journals such as *MIS Quarterly (MISQ), Information Systems Research (ISR), Journal of Management Information Systems (JMIS), and* others. He has also won multiple awards, including awards for best papers at different forums such as Academy of Management, International Conference on Information Systems, Association for Computing Machinery, and Society for Information Management. He currently serves as the senior editor at the *Journal of the Association for Information Systems (JAIS)* and *European Journal of Information Systems (EJIS)*.