

Association for Information Systems

AIS Electronic Library (AISeL)

International Research Workshop on IT Project
Management 2019

International Research Workshop on IT Project
Management (IRWITPM)

12-13-2019

How Management Control Portfolios are Developed for Modularized ISD Projects

Subasinghage Maduka Nuwangi

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

Darshana Sedera

Monash University, Australia, darshana.sedera@gmail.com

Shirish C. Srivastava

HEC School of Management, srivastava@hec.fr

Follow this and additional works at: <https://aisel.aisnet.org/irwitpm2019>

Recommended Citation

Nuwangi, Subasinghage Maduka; Sedera, Darshana; and Srivastava, Shirish C., "How Management Control Portfolios are Developed for Modularized ISD Projects" (2019). *International Research Workshop on IT Project Management 2019*. 9.

<https://aisel.aisnet.org/irwitpm2019/9>

This material is brought to you by the International Research Workshop on IT Project Management (IRWITPM) at AIS Electronic Library (AISeL). It has been accepted for inclusion in International Research Workshop on IT Project Management 2019 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

How Management Control Portfolios are Developed for Modularized ISD Projects

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

Darshana Sedera
Swinburne Business School
Swinburne University of Technology
Melbourne, Australia
darshana.sedera@gmail.com

Shirish C. Srivastava
HEC
Paris, France
srivastava@hec.fr

ABSTRACT

Information Systems Development projects often modularize the work by decomposing complex tasks to enable better management and control. While the objectives are noble, modularization itself can introduce interdependencies. Using the control theory perspective and leveraging case study research approach, we examine eight projects to unearth the four types of interdependencies in modularized Information Systems Development projects. Next, for the four types of interdependencies derived through the study, corresponding control portfolio is developed, making theoretical contributions and recommendations for practice.

Keywords

Interdependencies, Modularization, Control Theory, Information Systems Development Projects, Requirements

INTRODUCTION

Information Systems Development (ISD) organizations often conduct ‘modularization’ to simplify and execute complex tasks (Cataldo et al. 2008). Modularization minimizes the coordination costs, transaction costs and production costs in projects (Tanriverdi et al. 2007). Rooted in product development practice (Gerdin 2005), modularization entails splitting a whole product (task) into manageable chunks (modules), wherein each module is first developed independently and then recombined. In contrast to product development context, ISD project modules have multifarious inherent interdependencies (Taube-Schock et al. 2011; Nuwangi 2016). Managing such interdependencies between modules within ISD projects remains a significant challenge for organizations (Cataldo et al. 2009; Nuwangi et al. 2019).

The current study specifically focuses on the interdependencies between modules at the requirements elicitation phase¹, where the client requirements are gathered and decomposed (Kirkman 1998). Therein, similar requirements are grouped together as ‘requirement modules,’ which are then described in the Business Requirement Specifications (BRSs). Once derived, BRS in each module becomes the salient guide for the software developers, consultants and module managers. They use the BRSs to guide development activities, gauge progress and device management interventions. Moreover, BRSs also provide the essential boundaries of each module, outlining the points of interdependencies between modules. Past researchers (Cataldo et al. 2007; Cataldo et al. 2009) too have argued that managing interdependencies is an important aspect in project performance. However, requirement interdependencies in ISD projects are not well understood.

When examining how requirement interdependencies as stated in BRSs can be better managed, we take recourse in the control theory (Eisenhardt 1985; Kirsch 1997), which explains the use of a portfolio of controls to manage projects. Portfolio of controls consists of a mix of formal and informal controls (Choudhury and Sabherwal 2003; Kim and

¹ We note that ISD projects can derive modules at various phases of the ISD process. The requirements elicitation phase takes a special significance given that subsequent modules (e.g. testing modules) are most likely to be developed using modules derived in this phase.

Tiwana 2016; Wiener et al. 2016). While formal controls focus on managing employees through performance evaluation strategies, informal controls are based on social or people-based strategies (Kirsch 1997). We posit that different control mechanisms may be required for managing different types of requirement interdependencies in modularized ISD projects. Researchers have presumed a link between interdependencies and project controls (Sosa, Eppinger, & Rowles, 2004; Tiwana, 2008b). However, previous literature has rarely explored the linkage between types of requirement interdependencies and controls in ISD projects. As such, this research addresses the following two research question: “how modularized ISD projects employ various control mechanisms to effectively manage different types of requirement interdependencies?”. A deep understanding of the interdependencies between modules and how control portfolios can be developed to manage such interdependencies is a crucial contribution to the body of knowledge. Moreover, as Tiwana (2008b) argued, projects must be managed with careful attention to the interdependencies between modules, as the type of interdependency dictates which control modes need to be adopted. As interdependence between modules varies, SD projects should also change their management control portfolios (Mani et al. 2014).

This research utilizes abductive research approach to determine the types of requirement interdependencies and respective control portfolios. For this, we draw upon eight (8) projects in an organization that develops capital market solutions. Qualitative interview data and project management documentations were collected from the case organization to obtain an in-depth understanding of the phenomenon. The paper proceeds in the following manner. The next section discusses the theoretical background of the research. The subsequent section explains the research methodology, data collection and initial observations. This section is followed by the interdependency types and control portfolios section. Finally, the paper discusses research implications, limitations and future research.

THEORETICAL BACKGROUND

This study employs notions of two theoretical perspectives to build its findings: 1) the interdependencies (Pee et al. 2010; Pohl 1996) and 2) the control theory (Eisenhardt 1985; Kirsch 1997).

Interdependencies between Modules

Interdependencies between modules dictate how modules depend upon each other (Gerdin 2005; Macintosh and Daft 1987). Interdependencies range from autonomy (from the lowest form of pooling interdependency), to linkage to serial (for sequential interdependence), to project and mutual adjustment among specialized components (Gerdin 2005; Macintosh and Daft 1987). We focus on requirement interdependencies in ISD projects. As per Zhang et al. (2014), the types of requirement interdependencies suggested by Pohl (1996) and Dahlstedt and Persson (2005) can be considered as a general representation of requirement interdependency types suggested by the previous literature. However, research that identifies requirement interdependency types based on the direction (i.e., unidirectional and bidirectional) and temporal characteristics (i.e. temporary or permanent) remains scarce. Thus, this research focuses on identifying requirement interdependency types based on the direction and temporal characteristics. Research (Goknil et al. 2014; Salado and Nilchiani 2016) has discussed the importance of managing requirement interdependencies. The assessment of interdependencies is made using two important aspects established in research: (i) coupling, and (ii) cohesion (Adamo-Villani et al. 2009). Coupling refers to the interdependencies of one module with other modules and cohesion refers to the internal interdependencies of a module (Kwong et al. 2010). In this paper, we only focus on coupling, i.e. interdependencies between requirement modules. We use control theory to explain how a control portfolio can be developed to manage requirement interdependencies.

Control Theory

Kirsch (1997) defined control as “encompassing all attempts to ensure individuals in organizations act in a manner that is consistent with meeting organizational goals and objectives.” Formal and informal controls which are commonly considered in the control theory literature can assist complex project management (Mähring et al. 2017; Wiener et al. 2016). Formal controls involve managing employees through mechanisms such as performance evaluations, in which either the outcomes or behaviors of employees are measured, evaluated and rewarded. Formal controls are further subdivided into outcome-based and behavior-based control modes, where outcome-based modes specify the expected outcomes of projects and behavior-based modes influence behaviors (Eisenhardt 1985). Informal controls like social and people-based strategies are also commonly employed in ISD projects. Informal controls consist of clan controls and self-controls. Ouchi (1978) explained clan control as promoting common values and beliefs within a clan, which is defined as a group of individuals who share a set of common goals. Self-control occurs when the

employees control their own actions. While we acknowledge the importance of self-control in projects, it was excluded from this research due to two reasons: 1) self-control can increase self-reported bias (Schmeichel and Zell 2007), and 2) requirement modules were assigned to a group of team members. Rather than focusing on self-controls, this research focused on identifying controls for ISD teams.

When there are minimal interdependencies between modules, processes can be standardized and outcomes can be well predicted (Gerdin 2005). Thus, projects with minimal interdependencies can be better managed by outcome and behavior controls (Remus and Wiener 2012). However, interdependencies between modules are required to maintain the integrative nature warranted in contemporary software. According to Sosa et al. (2004), minimum interdependencies between modules create boundaries between teams, thereby increasing communication barriers between them. This occurs because interdependencies are not well understood. When project managers have a proper understanding of the interdependencies between modules, they can establish appropriate communication mechanisms to ensure that the project meets expected goals (Mani et al. 2014). Building on this background, this study provides a theoretical explanation for selecting appropriate control portfolios as per the types of requirement interdependencies.

RESEARCH METHODOLOGY, DATA COLLECTION AND INITIAL OBSERVATIONS

This study has a qualitative research design. The qualitative case study approach is recognized as appropriate for research that explores complex environments and contemporary events (Benbasat et al. 1987). This research followed an abductive research approach (Mantere and Ketokivi 2013; Timmermans and Tavory 2012). An abductive approach is suitable to explore the relationships between modularization and project controls by moving between data and literature. The abductive process commenced by studying existing literature on control theory and interdependencies between modules. Then, the data analysis process was started. There were interactive forward and backward moments between data and literature, attempting to identify various interdependency types and respective controls.

The case organization was selected using the criterion sampling strategy (Patton 2002). Three conditions formed the benchmark criteria for the selection of the ISD organization. First, the organization must modularize business requirements. Second, the organization must be involved in multiple ISD projects to provide flexibility in data collection. Third, the organization must be sufficiently large, with a standard hierarchy of employment to enable collection of data from team members at different levels of employment. Following the application of these criteria, StockEX was selected as the case organization. StockEX was established 10 years ago and has 300 staff members. The company specializes in the development of capital market solutions that include the functionalities of multiple trading methods and multiple asset classes such as equities, securities lending and borrowing². The reasonably narrow focus of the company is beneficial to the research as it allows extraneous factors (e.g. organizational control and cultural issues) to be avoided. Eight (8) ISD projects within company StockEX were selected for the data collection. See appendix A for a summary of project descriptions. Selection of the projects followed the opportunistic/emergent sampling strategy. The selected projects were similar in terms of the industry sector (ISD projects), type of software developed (capital market solutions), and project stage (completed) but varied in terms of clients (from different countries) and team members (different personnel). The selected projects were already completed, which enabled the respondents to discuss the interdependencies and control mechanisms of the entire project. Although the projects were completed, the team members were providing after delivery services and managing change requests of the clients. Therefore, the team members had a recent memory of the projects. The projects followed the incremental and iterative development method (Kneuper 2018). The incremental and iterative life cycle has been defined as “a project life cycle where the project scope is generally determined early in the project life cycle, but time and cost estimates are routinely modified as the project team understanding of the product increases. Iterations develop the product through a series of repeated cycles, while increments successively add to the functionality of the product” (Kneuper 2018, p. 90). Moreover, the projects followed ‘release train’ approach, where information system development tasks were divided into multiple iterations with short time frames, where at the end of each iteration some proportion of the information system was delivered to the client (Boehm, 1988).

Twenty-three (23)³ semi-structured interviews, each lasting approximately 30 minutes, were conducted with employees from 8 ISD projects. See appendix B for participant information. The participants consisted of one participant from the consultant team, one participant from the software engineering team, and one participant from

² For a wider discussion of the above concepts see for example Senarath and Copp (2015) and Senarath (2016).

³ Informant 2 was the project manager for both Project A and Project G.

the project management team. The sampling was non-probabilistic and purposive and employed the 'snowball' technique (Minichiello et al. 1995). All the interviews were audio-recorded and transcribed for subsequent data analysis, yielding approximately 145 pages of transcription. The interview data was supplemented by project management documents such as BRSs, change request documents, project descriptions, flow charts and design documents. Forty-three BRSs were collected: project A (14), project B (21), and project G (8). Three project description documents were collected: project C (1), project E (1), and project H (1). Nine change request documents were collected from project H and seven software design documents were collected from project C. Those documents were used as a means of data triangulation when establishing interdependencies and control portfolios. Since BRSs were written to explain the requirement modules, analysis of BRSs is considered as the best method to explore the interdependencies between modules. Moreover, BRSs explain the expected outcomes and behaviors (i.e., rules, procedures, and processes required to achieve the outcomes) of each requirement module. Therefore, BRSs were read in detail to identify the interdependencies between modules and formal controls in each project. The information extracted from BRSs was supported by interview data. In projects where the BRSs were not available, the analysis mainly focused on interview data. Consistent with Yin (2003), the 'ISD project' was selected as the unit of analysis. Data analysis included; (1) within-case analysis and (2) cross-case analysis. The overall aim of within-case analysis was to become familiar with each case as a stand-alone entity. Cross-case analysis was conducted to investigate the similarities and differences between the cases.

Identifying the interdependencies

During the data analysis process, we identified that some modules acted as the foundation for a substantial proportion of other modules, creating an interdependency between modules. For companies that develop capital market solutions, the trade processing module was recognized as the core or foundational module. Trade processing module created the foundation for interdependent modules such as fund processing, stock processing, general accounting and journal entries, clearing and settlement, trade reporting, trade validation and enrichment, market trade input, and trade details modules. In some projects, trade validation and enrichment, trade inputs and trade details were managed by separate modules rather than the trade processing module. However, the trade processing module served as the foundation module.

Relationships between modules

We discovered that in some projects the relationships between the foundation module and the dependent modules were maintained throughout the project. For example, Project G focused on developing a post-trade solution for the clearing and settlement of the executed trades. Throughout the project, the dependent modules such as the clearing and settlement and the cash management modules were executed per the requirements of the foundation module, i.e. the trade processing module. All trades entered through the trade processing module were settled through clearing and settlement module. This indicates a permanent connection between the trade processing and the clearing and settlement modules. However, in some projects, interdependencies between the foundation module and the dependent modules were established only when it was required. For example, Project E involved in the development of software solutions to detect irregular trading behaviors. The relationships between the trade processing and the dependent modules were established only when it was required to detect irregular trading behaviors.

Scope of the modules

The BRSs written for each module of the ISD projects consisted of information about the modules, including the scope, the flow of information between the modules and how different modules were related. Interfaces between modules explained how a particular module interacts with other modules. Interfaces between modules were not properly defined in some projects. In project D, the interfaces through which modules interacted were loosely defined in the BRSs.

Evidence of controls

We found evidence of simultaneous execution of multiple controls in projects. All the projects used BRSs to explain the expected outcomes and behaviors (i.e., the rules, procedures and processes required to achieve the outcomes) of each module. Thus, the BRSs were used to identify the outcome and behavior controls of projects. We identified that while there were some main control modes in projects, other control modes were used only in situations when they were necessary. For example, BRSs of project G consisted of in-detail information about expected outcomes and

behaviors. Since team members completed their tasks mainly based on BRSs, outcome and behavior controls can be considered as main controls. Clan controls were used only in situations where they were necessary.

Interdependencies and relevant control portfolios

During cross-case analysis, the characteristics of interdependencies and control modes of each individual project were grouped and compared with other projects. We used control theory perspectives and the literature on interdependencies as sensitizing devices. We identified two types of relationships between the foundation module and dependent modules; (i) unidirectional and (ii) bidirectional. The unidirectional relationships occurred when the dependent modules receive data from the foundation module, but the foundation module was not dependent on data received from the dependent modules. In bidirectional relationships, the foundation module and dependent modules were dependent on each other's data. There were permanent, temporary and unknown relationships between the foundation module and dependent modules. Permanent relationships were observed when dependent modules were executed per the requirements of the foundation module throughout the project. Temporary relationships were established between the foundation module and dependent modules only when it was required. Unknown relationships were formed when the relationships between the foundation module and dependent modules were undefined. The BRSs of some modules clearly stated the scope and interfaces between modules. However, the BRSs of some modules did not include clear information about the scope and interfaces. Further, we identified main and auxiliary controls in projects. Main controls were used to govern the modules, whereas auxiliary controls were used only in situations where they were necessary.

Then, we compared the characteristics of interdependencies and mapped them with control modes. By doing so, we were able to identify four types of interdependencies and appropriate control modes which govern these interdependencies.

INTERDEPENDENCY TYPES AND CONTROL PORTFOLIOS

While we acknowledge that the four types of interdependencies may co-exist in ISD projects and that the corresponding control portfolios too may have an overlapping role across the interdependencies, we derived four types of interdependencies and its corresponding control portfolio. During the project lifecycle, interdependency types may also evolve from one to another based on the project requirements.

Unidirectional Permanent Interdependency

The interdependency between two modules was considered a Unidirectional Permanent Interdependency, which occurred when there were unidirectional, permanent links between the foundation module and interdependent modules (see figure 1). While the dependent modules received data from the foundation module, the foundation module was not dependent on data received from the dependent modules. The scope and the interfaces of the interdependency were known. In Unidirectional Permanent Interdependency, the formal controls were established without negotiation between the modules and were the preferred mode to manage them. The role of clan control was less in projects where Unidirectional Permanent Interdependencies were observed.



Figure 1: Unidirectional Permanent Interdependency

Unidirectional Permanent Interdependency was observed in project G, which was focused on developing a post-trade solution for the clearing and settlement of executed trades. The solution provided the ability to execute manually-entered trades as well as market trades. In the process of clearing and settlement, the trade owners and relevant accounts were updated. Unidirectional Permanent Interdependency was observed between the trade processing and clearing and settlement and cash management modules. The trade processing module consisted of information required to process a trade whereas the cash management module contained details on calculating the fees and other taxation on a trade. The clearing and settlement module explained the processes for managing risks between a trade taking place and being settled and the exchange of cash and assets between buyers and sellers following a trade. When a trade is processed by the trade processing module, the cash management module and the clearing and settlement module manage the calculation of fees and taxes, and trade settlements, respectively. Therefore, the two modules—clearing and settlement, and cash management—were dependent on the information generated by the foundation module—the trade processing module. Moreover, it was identified that there were only a few updates to the trade processing module, indicating that the relationship between the trade processing module and dependent modules was

permanent. Since the dependent modules simply used data from the foundation module, formal controls were established without negotiation between the modules. As a result, there was a minimum level of clan controls in the project.

Unidirectional Temporary Interdependency

The interdependency between two modules was considered a Unidirectional Temporary Interdependency when there were unidirectional, temporary relationships between the foundation and dependent modules (see Figure 2). In a Unidirectional Temporary Interdependency, the scope and interfaces of the interdependency were known. As such, formal controls were established for the known interactions, while clan control was also used to govern the modules.



Figure 2: Unidirectional Temporary Interdependency

Unidirectional Temporary Interdependency was observed in projects E and F, which focused on developing software solutions for detecting irregular trading behaviors. Since the modules of the projects were executed per the outputs of the trade processing module, the relationships between the foundation module (i.e., trade processing module) and dependent modules such as the business intelligence module and the alert and case management module were unidirectional. Connections between the trade processing and dependent modules were established only when it was required to detect irregular trading behaviors. In projects E and F, the scope and interfaces between modules were known by the team members. Therefore, team members were able to take full responsibility of modules assigned to them. Since formal controls were established for the known interactions, the developed software solution could be closely aligned with BRSs. When there were issues in project F, software engineers, business analysts, and consultants had the flexibility to discuss the issues with team members. The issues were then communicated to the client for further feedback.

Bidirectional Permanent Interdependency

The interdependency between two modules was considered a Bidirectional Permanent Interdependency when there were permanent links between two modules and when each module was dependent partially on the other for deriving data (see Figure 3). The scope of the interdependency was pre-established; however, the interfaces through which they interact were loosely defined. Therefore, the use of formal controls was largely ineffective. Since the interfaces were not well-established, establishing these interfaces was done through clan control.



Figure 3: Bidirectional Permanent Interdependency

Bidirectional Permanent Interdependency was observed in projects B, C, D, and H. In project B, the trade validation and enrichment, market trade input and trade details, and trade processing BRSs had permanent interdependencies with each other. The trade validation and enrichment module consisted of information on, 1) trade validation to ensure that all required information is present in a trade, and 2) trade enrichment to include any missing information such as settlement details and settlement dates. The market trade input and trade details module consisted of information on capturing trades from various markets and converting the received trades to internal formats to maintain all information required to manage the trade through the lifecycle of clearing and settlement. The trade processing module consisted of information on viewing trade information, the trade update process, and trade allocation. The trade validation and enrichment BRS mentioned, *“This specification will cover the validation and enrichment activities. Upon completion of these activities the trade will be deemed to have been accepted into the system”* [trade validation and enrichment BRS]. The trade processing BRS specified, *“If a trade passes the initial basic validations (as discussed in the trade validations and enrichment specification) it will be considered to be in a pending (initial validation) status”* [trade processing BRS]. This indicates that these BRSs were dependent on each other for deriving data and execution information. Project C contained of a Bidirectional Permanent Interdependency, with formal bidirectional links between modules. This project focused on developing an automated order handling system, in which matching buy and sell orders are automatically processed. Since project C addressed a wide range of business needs, including connectivity and order management, updating the BRS of the foundation module was necessary as per the requirements of the dependent modules. Moreover, when team members wanted to integrate a new module with

existing modules, they had to develop interfaces between the modules. It was difficult to establish formal controls in project C. Respondent 07 from project C discussed, “Most of the time, the spec [BRS] carries out only high-level requirements”. Module interfaces were established through clan control. As per respondent 08 from project C, “We have to interact with the BAs [business analysts] a lot and get clarification.”

Bidirectional Unknown Interdependency

The interdependency between two modules was considered a Bidirectional Unknown Interdependency when there were undefined links between two modules and each module was dependent partially or fully on the other for deriving data (see Figure 4). Bidirectional Unknown Interdependencies were observed in projects at their early stages, especially when agreeing with specifications and deliverables. It was naturally hard to employ formal controls as the evolution of specifications makes it impossible to develop concrete targets. It was difficult to establish clan controls in projects with Bidirectional Unknown Interdependency. This was because the evolution of specifications created unmanageable targets for team members, which ultimately minimized team spirit.



Figure 4: Bidirectional Unknown Interdependency

Bidirectional Unknown Interdependency was observed in project A, where there were no formal links between modules, yet modules were dependent partially or fully on other modules. Interdependencies were observed between trade processing, fund processing and stock processing modules. When a trade was processed, the relevant fund accounts were updated by the ‘fund processing’ module whereas the relevant stock accounts were updated by the ‘stock processing module’. As specified in the revision history of the fund processing BRS, on December 5, 2011, the accounting structure of project A had to be removed as the structure could not be implemented without implementing another interdependent module. Since the trade processing BRS had interdependencies with fund processing BRS, it was necessary to update trade processing BRS as well. It was difficult to establish formal controls in project A. The evolution of specifications created unmanageable targets for the team members, leading to a minimum level of team spirit. Respondent 01 mentioned, “Team spirit was really lower [than other projects].”

Figure 5, which was developed to represent the four types of interdependencies and the relevant control modes, cross references the results. See Table 2 for detailed data analysis results.

Interdependency			Control Theory View			
Interdependency Types and Characteristics			Graphical Representation	Formal-outcome control	Formal-behaviour control	Informal-clan control
Types	Characteristics					
Unidirectional Permanent Dependency	Direction of dependency	Uni-directed		Main	Main	Auxiliary
	Type of the relationship	Permanent Scope and the interfaces between the modules were known				
Unidirectional Temporary Dependency	Direction of dependency	Uni-directed		Main	Main	Main
	Type of the relationship	Temporary Scope and the interfaces between the modules were known				
Bidirectional Permanent Dependency	Direction of dependency	Bi-directed		Auxiliary	Auxiliary	Main
	Type of the relationship	Permanent Scope of the modules were pre-established Interfaces through which modules interact were loosely defined				
Bidirectional Unknown Dependency	Direction of dependency	Bi-directed		Auxiliary	Auxiliary	Auxiliary
	Type of the relationship	Unknown Undefined Scope and the interfaces between modules were unknown				

Figure 5: Types of Interdependencies and Relevant Control Modes

Table 1: Data Analysis Results

Interdependencies & Controls		Representative quotes / BRS details
Unidirectional Permanent Interdependency	Characteristics <i>[Direction of interdependency]</i> Unidirectional <i>[Type of Relationship]</i> Permanent The scope and interfaces of the modules were known	Project G focused on developing a post-trade solution for the clearing and settlement of executed trades. Clearing, settlement, and cash management were conducted as per the functionalities of the trade processing module. Therefore, dependent modules such as cash management and clearing and settlement were dependent on information generated by the trade processing module. Moreover, it was identified that there were only a few updates to the trade processing module. The relationship between the trade processing module and dependent modules was permanent. In other words, throughout the project, the dependent modules were executed per the requirements of the trade processing module. “The purpose [code: scope known] of the clearing system is to get pre-matched [...] trades [code: unidirectional] ready for settlement, to calculate settlement obligations and, when required, to apply these settlement obligations [code: scope known].” [Project G, clearing and settlement BRS] “The clearing system will receive trades [code: unidirectional] in all forms of instruments executed on the exchange or regulated public market.” [Project G, clearing and settlement BRS] “Similar to shares, cash accounts are also maintained for each transaction [code: permanent] in the system.” [Project G, cash management BRS] “All trades must be settled against a specific investor account [code: permanent]. Investor account information will not be entered at the time of the trade but will have to be allocated by the broker as part of the clearing process.” [Project G, clearing & settlement BRS]
	Controls Formal controls established without negotiation Main controls outcome controls behavior controls Auxiliary controls clan controls	“We didn’t have to go and change the requirement, [which was written in BRS] [code: formal controls, without negotiation].” [Respondent 19, Project G] “For only few changes, we [software engineers] have to go and ask [from the business analysts], can we can do this change? [code: formal controls, without negotiation].” [Respondent 19, Project G] BRSs explained the expected outcomes and behaviors (i.e., required rules, procedures, and processes to achieve the outcomes) of each module. There were minimum updates to requirements specified in BRSs. Since team members completed their tasks mainly based on BRSs, outcome and behavior controls were identified as main controls. Since the team members rarely collaborated, clan controls were identified as auxiliary controls.
Unidirectional Temporary Interdependency	Characteristics <i>[Direction of interdependency]</i> Unidirectional <i>[Type of Relationship]</i> Temporary The scope and interfaces between modules were known	Projects E and F involved the development of software solutions to detect irregular trading behavior. Since the solutions were focused on detecting irregular trading behavior, the modules of the projects were executed per the outputs of the trade processing module (i.e., the foundation module). Therefore, the link between the foundation module and dependent modules was unidirectional. A relationship between the foundation module and dependent modules was established only when it was required to detect irregular trading behavior. Therefore, the relationship was a temporary connection between the foundation module and dependent modules. The scope of the foundation module and dependent modules was known. While the scope of the trade processing module was to execute the trades, other modules were focused on detecting irregular trade behaviors. “Manipulative trading behavior such as front running, insider trading, wash sales, layering the book, and marking the close are readily detected [code: scope known].” [Project E, project description document, p. 2]

	<p>Controls Formal controls were established for the known interactions</p> <p>Main controls: outcome controls behavior controls clan controls</p>	<p>“Instances of potential market abuse such as insider trading and market manipulation are reported to the [] for further investigation and possible enforcement action [code: scope known].” [Project F, project information document, p. 5]</p> <p>“The idea of the document [BRS] has to be closely synchronized with the actual system behaviors [code: formal controls, known interactions].” [Respondent 16, Project F]</p> <p>“[We follow the BRS in] a one-to-one basis [code: known interactions, formal controls]”. [Respondent 16, Project F]</p> <p>“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 [code: clan controls].” [Respondent 13, Project E]</p> <p>Since team members completed their tasks mainly based on BRSs, outcome and behavior controls were identified as main controls. Moreover, the team members worked together as a family and thus clan controls were also identified as a main control.</p>
<p>Bidirectional Permanent Interdependency</p>	<p>Characteristics [Direction of interdependency] Bidirectional</p> <p>[Type of Relationship] Permanent</p> <p>The Scope of the modules was pre-established</p> <p>Interfaces through which modules interact were loosely defined</p>	<p>While project C addressed a wide range of business needs, including connectivity and order management, project H focused on developing a platform that integrates modules. Project B developed a real-time clearing system that manages post-trade activities. Project D was focused on developing a software solution to provide specialized facilities for commodity markets such as metal (e.g., gold, silver and lead) and energy (e.g., natural gas and crude oil).</p> <p>Since projects C and H integrated modules, it was necessary to update the foundation module (i.e., trade processing module) per the requirements of the dependent modules. Similarly, the foundation modules of projects B and D were also updated during project execution. Although project B was a clearing system, some functionalities of the foundation module were required to be updated as a result of the interdependencies. Therefore, the relationship between the foundation module and the dependent modules was bidirectional.</p> <p>In projects C, H, B and D, the scope of the modules was pre-established; however, the interfaces through which they interacted were loosely defined. “These statistics will account for trades received only via the direct market source. It will not consider trades reported from [...] trading or manually entered/uploaded by an administrative user [code: scope pre-established].” [Project B, market trade input and trade details BRS], “The provision of such data is considered to be out of scope [code: scope pre-established].” [project H, statistics BRS]</p> <p>In project D, team members had a proper understanding of the interdependencies between modules, but the interfaces of the interdependencies were not clear to them. “If we are writing the specification [BRS], we have to analyze the impact areas. Those areas were not clearly analyzed.” [code: interdependency, interfaces loosely defined] [Respondent 10, Project D] “We need to get the interface done and then test the requirement and the functionality.” [code: interfaces are loosely defined] [Respondent 09, Project C]</p> <p>“Once a trade is captured, prior to accepting the trade [code: bidirectional, permanent] (and posting to trading/settlement accounts), the following activities need to take place [...] This specification [validation and enrichment BRS] will cover these validation and enrichment activities [code: scope of module pre-established]. Upon completion of these activities, the trade will be deemed to have been accepted [by the trade processing BRS] [code: bidirectional] into the system.” [Project B, trade validation and enrichment BRS]</p> <p>“Failed & rejected status trades will be referred back to gateway by communicating with the trading source [code: bidirectional].” [Project B, market trade input & trade details BRS]</p>

	<p>Controls Difficult to establish formal controls</p> <p>Auxiliary controls: outcome controls behavior controls Module interfaces were established through clan controls</p> <p>Main controls: clan controls</p>	<p>“Most of the time, the spec [BRS] carries out only high-level requirements [code: difficult to establish formal controls].” [Respondent 07, Project C]</p> <p>“Talking about [our project], the specs [BRSs] are little bit loose. So, it is little bit tough to do development based mainly on those specs” [code: difficult to establish formal controls].” [Respondent 08, Project C]</p> <p>“We have changes happening in the plan twice a week [...] we have new changes coming in to the plan [code: difficult to establish formal controls].” [Respondent 21, Project H]</p> <p>“We have to interact with the BAs [business analysts] a lot and get clarification [code: clan controls].” [Respondent 08, Project C]</p> <p>“First we engage with the team [code: clan controls] and find the solution.” [Respondent 21, Project H]</p> <p>Since the BRSs did not include sufficient information on requirements, team members had to work as a team to overcome issues during software development. Therefore, clan control was identified as a main control whereas outcome and behavior controls were identified as auxiliary controls.</p>
<p>Bidirectional Unknown Interdependency</p>	<p>Characteristics [Direction of interdependency] Bidirectional</p> <p>[Type of Relationship] Unknown Undefined The scope and interfaces between modules were unknown</p>	<p>The accounting structure of the fund processing BRS of project A could not be implemented as a result of interdependencies with other modules. Therefore, business analysts had to remove the accounting structure functionality from the fund processing BRS during the project life cycle. Since the trade processing BRS had interdependencies with the fund processing BRS, business analysts had to update the trade processing BRS as well. This indicates that the foundation module (i.e., trade processing module) and dependent module (i.e., fund processing module) rely partially or fully on the other for deriving data.</p> <p>The trade processing BRS was updated, including ‘strategy order ID’, which was required to apply the correct brokerage scheme specified in the brokerages taxes and charges BRS. This indicated that the foundation module (i.e., the trade processing module) was updated per the requirements of other modules. The trade processing BRS was updated by including the following statement; “The Strategy Order ID will be used only as an indicator to identify the strategy orders separately and to apply the correct brokerage scheme. Please refer to Volume 06, the Brokerages Taxes and Charges BRS for this functionality [code: bidirectional].” [project A, Trade processing BRS]</p> <p>“The whole account postings section was removed since the same had been explained in the Trade Processing spec” [code: scope of the module was unknown].” [project A, Fund processing BRS]</p> <p>“We had to remove the functionality completely [code: undefined, scope of the module was unknown].” [Respondent 01, Project A]</p>
	<p>Controls Difficult to establish outcome and behavior controls Difficult to establish clan controls</p> <p>Auxiliary controls: outcome controls behavior controls clan controls</p>	<p>“They [software engineers] have to change certain things because the document [BRS] is changing continuously [code: difficult to establish formal controls].” [Respondent 02, Project A]</p> <p>“They [clients] basically signed off on the business functionality. So, the BRS didn’t specify exactly how we are going to give [the software solution] to you [code: difficult to establish formal controls].” [Respondent 01, Project A] Most of the time, when we [consultants] suggest a problem or suggest a solution, [software engineers mention that] we can’t do this [code: difficult to establish clan controls].” [Respondent 01, Project A]</p> <p>“Team spirit was really lower [than other projects] [code: minimum clan controls].” [Respondent 1, Project A]</p> <p>Since the BRSs were frequently updated, it was difficult to establish outcome and behavior controls. The evolution of specifications created unmanageable targets for team members, thereby minimizing team spirit. Therefore, outcome, behavior, and clan controls were identified as auxiliary controls.</p>

IMPLICATIONS, LIMITATIONS AND FUTURE RESEARCH

The objective of this study was to examine how interdependent modularized SD projects are managed using a portfolio of controls. We employed eight projects selected from a single ISD organization that specializes in developing capital market solutions. The study sought evidence from 23 participants and used documents such as BRSs to draw further insights. During data analysis, four types of module interdependencies were identified: (i) Unidirectional Permanent Interdependency; (ii) Unidirectional Temporary Interdependency; (iii) Bidirectional Permanent Interdependency; and (iv) Bidirectional Unknown Interdependency. Each of the four interdependencies was discussed and the study demonstrated a rationale for using the relevant control modes based on the type of interdependency.

First, this is one of the first studies to highlight the importance of considering interdependencies between modules when managing ISD projects. Our findings add further substance to the claims presented in Tiwana (2008), who explained how minimizing interdependencies between outsourced systems and other software applications is a substitute for controls. While such studies made a substantial contribution to our knowledge, a theoretical explanation for how modularized ISD projects can be controlled using a portfolio of controls remains absent. As such, we extend past research (Choudhury and Sabherwal 2003; Kirsch 1997; Tiwana 2008) by highlighting the importance of selecting appropriate control portfolios based on the types of requirement interdependencies between modules in ISD projects.

Second, the study findings extend the applicability of the notion of the control portfolio explained in the control theory by providing a nuanced view of the requirements analysis phase of ISD projects. While the existence and importance of a portfolio of controls have been discussed in the prior literature, research on the requirements analysis stage is scant. For example, Kirsch (2004) pointed out that the different stages of ISD projects may include task interdependencies, leading to changes in the choice of control modes. We discuss how ISD projects have implemented outcome, behavior and clan controls based on interdependencies between requirement modules.

Third, this study develops four types of module interdependencies to extend research on modularity. Although previous research has discussed types of requirement interdependencies (e.g. Dahlstedt and Persson (2005), Pohl (1996)), research that identifies requirement interdependency types based on the direction (i.e., unidirectional and bidirectional) and temporal characteristics (i.e. temporary or permanent) remains scarce. The new requirement interdependencies we introduced provide a nuanced view that will enable future research to derive more insights through a better understanding of the nature and complexity of interdependencies in contemporary ISD projects.

For practitioners, this study offers several insights into better management of ISD projects. First, the study demonstrates the impact of the modularization of requirements, which is commonly employed in ISD companies. More specifically, the study findings highlight the impact that interdependencies between modules have on project controls. As such, it demonstrates that better communication is required between business analysts, software engineers and project managers when deriving requirement modules - a task which was traditionally conducted only by the business analysts.

Second, practitioners can benefit from our findings regarding the control portfolios employed in each project. More specifically, the study demonstrates the vital role BRSs play in project control. If developed with clear instructions observing interdependencies, BRSs can serve as a structured formal control mechanism for all modules. However, given the inevitable interdependencies between modules, managers must employ a portfolio of controls to govern projects.

Third, the study provides four interdependencies that practitioners can use to assess the main interdependency type in a project and then develop the control portfolio. The study characterizes the interdependencies and provides means of how they can be assessed by observing the information flows between modules. Such a structured approach will undoubtedly benefit managers in managing future ISD projects.

Though the study makes several theoretical and practical contributions, it has a few limitations that need to be acknowledged. First, although procedures for internal and external validity were followed, the subjectivity of data collection and analysis may be an issue. Moreover, the data was collected from a single organization. In future studies of this nature, generalizability can be improved by using a larger sample of projects, perhaps representing multiple organizations. Second, since self-reported self-control can be biased, self-control was excluded from this research. Future research can examine self-control if there is the possibility to avoid potential bias. Third, although we have

collected the data from a single organization, the organization structural characteristics and cultural issues may have an impact on project controls, which we have not considered in this study. Future research can explore the impact of organization structural characteristics and cultural issues on project controls. Forth, there can be other interdependency types in ISD projects. For example, the classification used in this study is currently based on a 2x3 grid: unidirectional/bidirectional X open/temporary/permanent. Yet, during the data analysis we account only four forms. Future research could explore the existence of other forms of interdependencies in ISD projects. Finally, a longitudinal study on managing interdependencies in modularized ISD projects could yield further insights into the phenomenon being observed. For example, sequential and parallel development of modules in ISD projects and the impact of project maturity level on controls could be observed through a longitudinal study.

REFERENCES

- Adamo-Villani, N., Antsaklis, P. J., Aragon, C. R., Bagheri, N., Baiden, G., Balasubramanian, P., ...Zak, S. H. 2009. "Handbook of Automation ", S.Y. Nof (ed.). Springer.
- Ahlstedt, A. and Persson, A. (2005). *Requirements Interdependencies: State of the Art and Future Challenges*. In: Aurum A. and Wohlin C., ed., *Engineering and Managing Software Requirements*. [online] Springer Science & Business Media.
- Benbasat, I., Goldstein, D. K., and Mead, M. 1987. "The Case Research Strategy in Studies of Information Systems," *MIS Quarterly* (11:3), pp. 369-386.
- Cataldo, M., Bass, M., Herbsleb, J. D., and Bass, L. 2007. "On Coordination Mechanisms in Global Software Development," in: *IEEE International Conference on Global Software Engineering*. IEEE, pp. 71-80.
- Cataldo, M., Herbsleb, J. D., and Carley, K. M. 2008. "Socio-Technical Congruence: A Framework for Assessing the Impact of Technical and Work Dependencies on Software Development Productivity," in: *ACM-IEEE International Symposium on Empirical Software Engineering and Measurement*. Kaiserslautern, Germany: ACM, pp. 2-11.
- Cataldo, M., Mockus, A., Roberts, J. A., and Herbsleb, J. D. 2009. "Software Dependencies, Work Dependencies, and Their Impact on Failures," *IEEE Transactions on Software Engineering* (35:6), pp. 864-878.
- Choudhury, V., and Sabherwal, R. 2003. "Portfolios of Control in Outsourced Software Development Projects," *Information Systems Research* (14:3), pp. 291-314.
- Eisenhardt, K. M. 1985. "Control: Organizational and Economic Approaches," *Management Science* (31:2), pp. 134-149.
- Gerdin, J. 2005. "The Impact of Departmental Interdependencies and Management Accounting System Use on Subunit Performance," *European Accounting Review* (14:2), pp. 297-327.
- Goknil, A., Kurtev, I., van den Berg, K., and Spijkerman, W. 2014. "Change Impact Analysis for Requirements: A Metamodeling Approach," *Information and Software Technology* (56:8), pp. 950-972.
- Kim, S. K., and Tiwana, A. 2016. "Chicken or Egg? Sequential Complementarity among Salesforce Control Mechanisms," *Journal of the Academy of Marketing Science* (44:3), pp. 316-333.
- Kirkman, D. P. 1998. "Requirement Decomposition and Traceability," *Requirements Engineering* (3:2), pp. 107-114.
- Kirsch, L. J. 1997. "Portfolios of Control Modes and Is Project Management," *Information Systems Research* (8:3), pp. 215-239.
- Kirsch, L. J. 2004. "Deploying Common Systems Globally: The Dynamics of Control," *Information Systems Research* (15:4), pp. 374-395.
- Kneuper, R. 2018. "Software Processes and Life Cycle Models: An Introduction to Modelling, Using and Managing Agile, Plan-Driven and Hybrid Processes." Switzerland: Springer, pp. 1-344.
- Kwong, C. K., Mu, L. F., Tang, J. F., and Luo, X. G. 2010. "Optimization of Software Components Selection for Component-Based Software System Development," *Computers & Industrial Engineering* (58:4), pp. 618-624.
- Macintosh, N. B., and Daft, R. L. 1987. "Management Control Systems and Departmental Interdependencies: An Empirical Study," *Accounting, Organizations and Society* (12:1), pp. 49-61.
- Mähring, M., Wiener, M., and Remus, U. 2017. "Getting the Control Across: Control Transmission in Information Systems Offshoring Projects," *Information Systems Journal*, pp. 1-21.

- Mani, D., Srikanth, K., and 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), pp. 846-864.
- Mantere, S., and Ketokivi, M. 2013. "Reasoning in Organization Science," *Academy of management review* (38:1), pp. 70-89.
- Minichiello, V., Aroni, R., Timewell, E., and Alexander, L. 1995. *In-Depth Interviewing: Principles, Techniques, Analysis*, (2nd ed.). Melbourne, Australia: Pearson Education Australia.
- Nuwangi, S. M. 2016. "The Impact of Modularisation on Information System Development Outsourcing Project Control," in: *Information Systems*. Queensland University of Technology, Australia p. 259.
- Nuwangi, S. M., Sedera, D., and Srivastava, S. C. 2019. "Control Mechanisms for Managing Modularized ISD projects," *Pacific Asia Conference on Information Systems*, Xi'an, China: Association for Information Systems AIS Electronic Library (AISeL), pp. 1-14.
- Ouchi, W. G. 1978. "The Transmission of Control through Organizational Hierarchy," *Academy of Management Journal* (21:2), pp. 173-192.
- Patton, M. Q. 2002. *Qualitative Research and Evaluation Methods*. Thousand Oaks, Calif: Sage.
- Pee, L. G., Kankanhalli, A., and Kim, H.-W. 2010. "Knowledge Sharing in Information Systems Development: A Social Interdependence Perspective," *Journal of the Association for Information Systems* (11:10), pp. 550 - 575.
- Pohl, K. 1996. *Process-Centered Requirements Engineering*. New York, USA: John Wiley & Sons, Inc.
- Remus, U., and Wiener, M. 2012. "The Amount of Control in Offshore Software Development Projects," *Journal of Global Information Management* (20:4), pp. 1-26.
- Salado, A., and Nilchiani, R. 2016. "The Concept of Order of Conflict in Requirements Engineering," *IEEE Systems Journal* (10:1), pp. 25-35.
- Schmeichel, B. J., and Zell, A. 2007. "Trait Self-Control Predicts Performance on Behavioral Tests of Self-Control," *Journal of Personality* (75:4), pp. 743-756.
- Senarath, S., and Copp, R. 2015. "Credit Default Swaps and the Global Financial Crisis: Reframing Credit Default Swaps as Quasi-Insurance," *Global Economy and Finance Journal* (8:1), pp. 135-149.
- Senarath, S. 2016. "Not So 'Bankruptcy-Remote': An Insight into Sri Lankan Securitization Practices in a Post_Gfc Context," in: *Multidisciplinary Academic Conference on Management, Marketing and Economics*. pp. 53-60.
- Sosa, M. E., Eppinger, S. D., and Rowles, C. M. 2004. "The Misalignment of Product Architecture and Organizational Structure in Complex Product Development," *Management Science* (50:12), pp. 1674-1689.
- Tanriverdi, H., Konana, P., and Ge, L. 2007. "The Choice of Sourcing Mechanisms for Business Processes," *Information Systems Research* (18:3), pp. 280-299.
- Taube-Schock, C., Walker, R. J., and Witten, I. H. 2011. "Can We Avoid High Coupling?," in *Ecoop 2011 – Object-Oriented Programming*. Springer, p. 657.
- Timmermans, S., and Tavory, I. 2012. "Theory Construction in Qualitative Research: From Grounded Theory to Abductive Analysis," *Sociological theory* (30:3), pp. 167-186.
- Tiwana, A. 2008. "Does Technological Modularity Substitute for Control? A Study of Alliance Performance in Software Outsourcing," *Strategic Management Journal* (29:7), pp. 769-780.
- Wiener, M., Mähring, M., Remus, U., and Saunders, C. 2016. "Control Configuration and Control Enactment in Information Systems Projects: Review and Expanded Theoretical Framework," *MIS Quarterly* (40:3), pp. 741-774.
- Yin, R. K. 2003. *Case Study Research: Design and Methods*, (3rd ed.). Thousand Oaks, CA: Sage Publications.
- Zhang, H., Li, J., Zhu, L., Jeffery, R., Liu, Y., Wang, Q., and Li, M. 2014. "Investigating Dependencies in Software Requirements for Change Propagation Analysis," *Information and Software Technology* (56:1), pp. 40-53.

APPENDIX A

Project Name	Description of the Project
A	The purpose of this project is to develop a post-trade application, which provides clearing and settlement for the trades after execution. Functionalities of the software application include trade processing, user management, general accounting and journal entries. Moreover, the software application consists of trade processing methods which are integrated with clearing and settlement procedures. The client is situated in the Asian region.

Project Name	Description of the Project
B	This is a real-time clearing system that manages post-trade activities. This project provides solutions for the limitations of current traditional clearing systems. This project consists of three major functionalities: 1) clearing (the process of matching, recording and transaction processing); 2) settlement (trade settlement); and 3) risk management (managing the risks of market participants). The client company is situated 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 software solution include: connectivity hub and post-trade risk management. The project consists of around 40 clients all over the world including clients from North America and Asia.
D	This project involves developing a tested, real-time and transparent trading platform. Furthermore, it facilitates robust assaying and warehousing facilities to execute the trades. The key features of software solution include: hedging; risk management and clearing and settlement. The client is situated in the Asian region.
E	This project involves the development of tools to detect irregular trading behaviors. Key functionalities of the software include: analysis of real-time / offline transaction data; provide alert and case-management. This project consists of several clients including clients from Asian and African regions.
F	This is a project to develop a software solution to detect irregular trading behaviors. This project is focused on developing a platform for a client, who was formed in 1980s. This client provides capital market solutions for several instrument trading. The derivatives offered by the client include bonds, indices and commodities.
G	This project includes the development of a post-trade application. The post-trade application includes clearing and settlement of the executed trades. There are several users (e.g. registry owner, brokers and custodian banks) with different authorization levels; registry owner is the main user. Client company is situated in the Asian region.
H	The application developed in this project can be adjusted to trade varied products in different types of markets. It is a reliable and flexible solution, which can be adjusted to meet client's specific needs. This project consists of multiple clients from all over the world, including clients from Europe.

APPENDIX B

Project	Participant ID	Designation	Years of experience in 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

Project	Participant ID	Designation	Years of experience in ISD industry	Years of experience in the company
	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.