ECIS 2014 Proceedings

COMPLEXITY IN IS PROGRAMS: A DELPHI STUDY

Everlin Piccinini
Georg-August Universität Göttingen, Göttingen, Niedersachen, Germany, epiccin@wiwi.uni-goettingen.de

Robert Gregory
IESE Business School, Barcelona, Barcelona, Spain, RWGregory@iese.edu

Jan Muntermann
University of Göttingen, Göttingen, Germany, muntermann@wiwi.uni-goettingen.de

Everlin Piccinini, Robert Gregory, and Jan Muntermann, 2014, "COMPLEXITY IN IS PROGRAMS: A DELPHI STUDY", Proceedings of the European Conference on Information Systems (ECIS) 2014, Tel Aviv, Israel, June 9-11, 2014, ISBN 978-0-9915567-0-0
http://aisel.aisnet.org/ecis2014/proceedings/track13/1

COMPLEXITY IN IS PROGRAMS: A DELPHI STUDY

Complete Research

Piccinini, Everlin, University of Göttingen, Göttingen, Germany, everlin.piccinini@wiwi.uni-goettingen.de

Gregory, Robert, IESE Business School – University of Navarra, Barcelona, Spain, rwgregory@iese.edu

Muntermann, Jan, University of Göttingen, Göttingen, Germany, muntermann@wiwi.uni-goettingen.de

Abstract

Information systems (IS) programs, i.e., temporary organizations consisting of a coherent group of interrelated projects that are jointly coordinated for achieving a common set of goals and objectives, are growing in importance for implementing strategic IT-driven organizational initiatives. Yet we hardly understand the specific challenges associated with them. In this paper, we draw connections between the nature of an IS program and the concept of complexity, arguing that a deeper understanding about the nature of complexity in IS programs is needed. Due to the lack of research in this area, we opted for an exploratory Delphi study. Our results point to three key themes: (1) the importance of organizational complexity in IS programs, (2) the role of organizational dynamic complexity, and (3) the challenges of coordinating and controlling IS programs. These findings offer a practically grounded foundation for exploring the specific challenges of managing IS programs.

Keywords: IT/IS Project Management, Program Management, Delphi Study, Complexity.

1 Introduction

Scholars in the field of information systems (IS) have been debating for quite some time how to transform organizations with the help of information technology (IT) (e.g., Sauer and Yetton, 1997; Teo et al., 1997). A key argument in this scholarly discourse is that value can be realized from the strategic implementation and use of IT if it is complemented by targeted changes in the organization's strategy, business processes, and structures and guided by the vision of increasing organizational competitiveness (Gregor et al., 2006). Typically, IS programs are designed and executed for implementing strategic IT-enabled organizational transformation initiatives (Ross et al., 2006). A program is a special kind of temporary organization that consists of a coherent group of interrelated projects that are jointly coordinated for achieving a common set of goals and objectives (Ferns, 1991). Frequently, a program is the selected organizational form when the goals and objectives are associated with an organizational strategy and their implementation is extremely difficult, involves a lot of change, and takes many years (Levin and Ward, 2011). Thus, program management is particularly pertinent to IS contexts, which frequently exhibit these characteristics. As a result, there is a great need for understanding the nature of IS programs as a foundation for examining IS program management about which we know very little.

IS programs involve a number of things that are very closely related to the phenomenon of complexity. In IS research the term complexity has been used to describe the challenges of large numbers (e.g., of technological components of an information system), variety (e.g., of relations and interdependencies between system components), and rate of change (e.g., changes made to system components over time) (Schneberger and McLean, 2003). Comparing these general facets of IS complexity to the nature of IS programs reveals their close connections. For example, the duration of an IS program normally spans a large number of years and, in addition, a large number of goals, stakeholders, affected users, etc., are involved (Pellegrinelli, 1997). Furthermore, an IS program typically consists of a variety of interdependent projects and associated goals, tasks, etc., that need to be coordinated as a whole (Ferns, 1991). Finally, the rate of change in an IS program is typically large, which is the natural consequence of the frequent involvement of a major business transformation (Gregor et al., 2006). Thus, in summary, the connection between complexity and IS programs is apparent, motivating this piece of research, which focuses on the nature of complexity in IS programs.

Prior IS research interested in the topic of complexity has either focused on as-is IS systems and architectures (e.g., Schneberger and McLean, 2003) or, if examining this phenomenon in a change context, focused on the project, as opposed to the program context. For example, Baccarini (1996) conceptualizes project complexity as the number and variety of technical and organizational elements of a project as well as the interdependencies between them. In a more comprehensive study, Xia and Lee (2005) characterize IS project complexity along two different dimensions: organizational/technological and structural/dynamic. Furthermore, the authors develop a list of complexity components that helps conceptualize and measure complexity in IS projects. These two examples (among others) illustrate the theoretical gap that we address with this paper, namely that, to the best of our knowledge, there has been no empirical study examining complexity in IS programs to date. We argue that there is a dedicated need to examine the nature of complexity in IS programs, complementing existing studies on complexity in IS projects, because IS programs differ from IS projects in many respects. For example, due to their strategic nature, IS programs are larger in scope than traditional IS projects and involve simultaneous change to a large percentage of an organization's IS (Ribbers and Schoo, 2002). As a result, IS program managers likely have to consider the full scope of IS complexity in an organization to a larger extent, whereas smaller projects only have to take a sub-portion of this complexity into account.

In order to address this theoretical gap, we decided to conduct an exploratory study of experts' opinions in the field of IS program management, i.e., a Delphi study, because we recognized that there were not many existing studies that we could build upon, motivating a research design that allowed us to capture knowledge-in-practice as a first step and foundation for building scientific knowledge. With our Delphi study, our goal was to answer the following research questions: (1) What is the nature of complexity in IS programs, i.e., what are the components of this complexity? (2) What is the relative importance of these components? The Delphi study was conducted by eliciting the opinion of 28 experts in IS programs in a process of brainstorming, selection and ranking. The experts contributed essential practitioner insights to help identify a comprehensive list of IS program complexity components and understand the relative importance of each component.

2 Theoretical Background

2.1 Introduction to organizational programs

There is an obvious close connection between the concept of a project and a program, most importantly, programs consist of projects. Furthermore, many management processes and techniques that we know very well from project management are also used in the context of a program. However, programs are different from projects in various ways (see Table 1). In IS we have a significant gap in knowledge about the phenomena of IS programs and IS program management. In fact, there are almost no studies available that we can build upon, which is why we draw upon the general program management literature in the following to discuss the uniqueness of programs.

Project	**Program**
• less strategic focus	• more strategic focus
• normally contains one set of goals	• normally contains multiple sets of interrelated goals
• relatively narrow scope of change	• relatively broad scope of change
• relatively stable and of short duration	• relatively unstable and of long duration

Table 1. Differences between project and program, adapted from Piccinini and Gregory (2013).

Programs are typically more strategic in nature than projects (Levin and Ward, 2011), i.e., they involve a greater variety of stakeholders with different interests and expectations to be coordinated and managed. A program typically includes a plurality of different goals and objectives, while a project is frequently limited to one specific (set of) objective(s) (Pellegrinelli, 1997). Moreover, the goals of programs are more strategic to the organization than projects, and their chief objective is to create business value through better organization and coordination of projects and their activities (Pellegrinelli, 1997). A further difference between a project and a program is that the latter is typically of longer duration and involves higher degrees of change than the former. This implies that due to the dynamics of their environments over a long period of time, programs frequently face a higher level of uncertainty in the execution of their tasks and objectives (Bechky, 2006). Additionally, there are some management processes that are specific to programs, such as coordinating scope definition of involved projects, coordinating project implementation planning, and controlling implementation activities simultaneously across project and program levels (Görög, 2011). In summary, the key unique characteristics of programs are their focus on executing organizational long-term objectives, the interdependencies that exist between a variety of interrelated projects they execute, and their scope of change that typically involves a major organizational transformation. Based on our understanding of IS contexts in which programs are executed (e.g., Gregor et al., 2006; Sauer and Yetton, 1997; Teo et al., 1997), we argue that these characteristics of programs in general also apply to IS programs.

2.2 Complexity in IS projects and programs

As stated before, programs usually involve transformative change, which prior research has attempted to understand from a complexity perspective (Burnes, 2005). In addition, the other key characteristics of programs mentioned above are also associated with complexity (e.g., the *variety* of components of a system (Schneberger and McLean, 2003)). However, despite this obvious connection between complexity and programs, we have a limited understanding of complexity in programs. Due to the lack of literature about this topic, we build upon prior related research on (IS) project complexity and IS complexity in general (see Table 2 for a summary of main studies in these areas).

Study	Context	Focus	Findings
Xia and Lee (2005)	IS project complexity	Identification of factors for conceptualizing and measuring complexity in IS projects	IS project complexity can be structured along four meta-categories: (1) organizational structural complexity, (2) organizational dynamic complexity, (3) technological structural complexity, and (4) technological dynamic complexity.
Baccarini (1996)	Project complexity	Conceptualization of project complexity	The number of organizational and technological elements of a project and the interdependencies between them are key aspects of project complexity.
Benbya and McKelvey (2006)	IS development complexity	Identification of sources of complexity related to IS development	A model presenting seven principles of 'adaptive success', which can enable the adaptation of IS development projects to changing stakeholder interests or environments.
Ribbers and Schoo (2002)	System implementation complexity	Analysis of how to cope with complex software implementations	Programs can be better managed by considering the complexity involved. System implementation complexity can be operationalized into variety, variability, and integration of elements of a system.
Schneberger and McLean (2003)	Computing complexity	Analysis of the number and variety of interrelated components and their combined rate of change	Complexity of the computing environment, measured in terms of the number and variety of components and their interactions, affects the difficulty of IS processes. The rate of system change considerably increases computing complexity.

Table 2. Overview of existing research on (IS) project complexity.

Finally, a common theme in prior IS research examining complexity is a focus on concepts such as variety, interdependencies (or interrelations, interactions), and change. Furthermore, both organizational and technological aspects of complexity in IS contexts have been studied. We argue that complexity in IS projects forms a subset of complexity in IS programs, with the implication that previously identified components of IS project complexity are potentially also relevant for understanding complexity in IS programs, but not exclusively. In other words, more research is needed to generate a full understanding of complexity in IS programs.

3 Research Approach

The primary aim of the research questions proposed in this study is to develop an accurate list of IS program complexity components and to determine which of these components are most relevant. One of the most meaningful sources for obtaining such information is through experts' knowledge in IS programs. Experts can contribute with their practitioner expertise and provide insights into issues not yet discussed in scientific literature. Therefore, we selected the Delphi method as the main research approach of this study. This method has been used frequently in IS research (e.g., Okoli and Pawlowski, 2004; Schmidt, 1997; Singh et al., 2009; Paré et al., 2013) and was designed to elicit the opinion of a panel of experts through a constant process of controlled feedback (Schmidt et al., 2001). The Delphi method is recommended when the problem is difficult to approach with explanatory methods and techniques and therefore profits from an exploratory approach that seeks to create consensus among a group of experts as a foundation and first step for further research (Singh et al., 2009). A great advantage of using the Delphi method is that “a panel study most appropriately answers the research questions, rather than any individual expert’s responses” (Okoli and Pawlowski, 2004, p. 18). Moreover, the panel size requirements of this method are modest, usually varying from 10 to 18 members, which facilitates the search for experts with deep understanding and knowledge about the study issues (Okoli and Pawlowski, 2004). The Delphi method suggests three distinct phases in data collection and analysis: (1) brainstorming, (2) selection, and (3) ranking of issues (Schmidt, 1997). Table 3 summarizes the tasks associated with each of these three phases.

Phase	Tasks	Outcomes
1st Brainstorming	• Each panelist provides his/her individual view of what is important in dealing with the given phenomenon • Research team consolidates the single responses into one list, removing duplicate items and unifying terminology • Consolidated list is sent to experts for validation • Researcher refines final version of consolidated list	• A list of 32 IS program complexity components
2nd Selection	• The panelists are asked to reduce the consolidated list of components by selecting the most important ones • Research team reduces the size of the list, maintaining the items selected by the majority of the panelists (>=70%)	• A consolidated list of 14 un-ranked IS complexity components
3rd Ranking	• Each panelist ranks the randomized list of components • Research team measures the degree of consensus of the panelists using Kendall’s coefficient of concordance (W) • Research team provides controlled feedback to panelists • Constant ranking is continued until the panelists reach either an acceptable consensus or stopping criteria are fulfilled	• A list of 14 ranked IS complexity components

Table 3. Delphi phases and associated tasks adapted from Schmidt (1997).

3.1 Panel selection

The Delphi method does not depend on a large statistical sample that attempts to represent any population, and therefore, it requires qualified experts with deep understandings of the study’s topic (Okoli and Pawlowski, 2004). In order to be a member of our Delphi panel, we required at least two years of experience in IS program management and participation in at least three IS programs, either as a company’s employee or consultant. Based on these selection criteria, an e-mail was sent to possible panelists explaining the subject of study, the procedures, and the commitment required for participation. After receiving the participation confirmation from the first experts, we asked them to help find additional panelists matching the selection criteria using the snowball sampling technique according to Singh et al. (2009). In this study, the Delphi panel was composed of 28 practitioners experienced in managing IS programs. The demographic data for the panel is provided in Table 4.

Characteristics	Panel profile (N=28)
Number of IS programs in which panelists participated	Mean: 13 programs (range: 3–30); 3–10 programs: 50%; 11–20 programs: 21%; > 20 programs: 29%
Number of years of IS program experience	Mean: 9 years (range: 2–26); 2–5 years: 38%; 6–10 years: 25%; 11–15 years: 23%; > 15 years: 14%
Nature of involvement in IS programs	As consultant: 61%; as employee: 39%
Type of industries panelists worked in IS programs	Banking & financial services: 96%; IT products & services: 32%; business consultancy: 28%; information & communications technologies: 28%
Panelists’ educational qualifications	Technical degree: 4%; bachelor degree: 4%; master degree: 79%; PhD degree: 13%

Table 4. Demographic profile of the Delphi panel.

The panelists had an average of 9 years of IS program experience, either as consultants or as employees. Although categorized as such, the panelists held a variety of job positions, including program manager, project manager, business manager, IT manager, business director, transformation expert, transformation director, business architect, business vice–president, business consultant, and technology consultant. All of the experts were located in Germany, which avoids consensus and/or context problems due to intercultural differences (Kasi et al., 2008). They have worked for small, medium and large organizations in different industries and participated on average in 13 IS programs. As demonstrated in Table 4, the profile of the experts selected indicates a great level of IS program experience for conducting this Delphi study.

3.2 Data collection and analysis methods

The data collection and analysis were carried out based on the three-phase process used by Schmidt (1997) (see Table 3). The Delphi study was designed and conducted entirely over the Internet through a standard online survey platform, while the feedback rounds were carried out via e-mail. After designing the online survey, we created a web link and in order to pre-test the online portal functions for the application of the Delphi method. Two Master students and two Ph.D. students volunteered to validate the instrument. Afterwards, we sent the web link to the panelists along with the definition of 'program' in order to ensure that they understood this concept and responded properly to the Delphi study according to their experiences in program and not project management.

In the first phase, we asked each of the 28 panelists to provide at least six components of complexity in IS programs and a brief description for each one, all panelists participated in phase 1. According to Singh et al. (2009), descriptions facilitate the understanding of the experts' answers, the process of elimination of duplicates, and the categorization of answers into meta-categories. The first phase resulted in a list of 182 items related to IS program complexity. Each expert listed 5 to 10 IS program complexity items. We consolidated and refined the list by removing identical answers, grouping similar answers together, and providing a consolidated description of the items (Schmidt, 1997). This process resulted in a consolidated list of 32 IS program complexity components, grouped into five meta-categories. Next, we sent this list to all respondents via e-mail in order to verify that their answers had been properly classified and that their ideas had been adequately represented. According to Schmidt (1997), "without this step, there is no basis to claim that a valid, consolidated list has been produced" (p. 769). After receiving feedback from the panelists, we revised the compiled list of IS program complexity components and the suggested changes were made accordingly.

In the second phase, we sent the consolidated and randomized list from phase 1 to the panellists, 27 panelists participated in phase 2. Following Schmidt's recommendations (1997), we asked the panelists to select the 20 components they considered most relevant in terms of driving complexity in IS programs. The selection of the top 20 components from each expert was reviewed, and only the components that were selected by the majority of the experts in the panel were retained for further investigation. Based on Singh et al. (2009), a cut-off value of 70% was chosen, giving a target range of 12–15 components for the successive ranking phase. At the end of the second phase, the panelists narrowed down the list of 32 components into a manageable list of 14 IS program complexity components.

In the third phase, we asked each of the panelists to rank the resulting 14 components of phase 2 in order of priority, 23 panelists participated in phase 3. The Delphi method aims at shaping a panel consensus about the relative importance of particular items and according to Okoli and Pawlowski (2004), the panelists are prone to reach a higher degree of consensus when the members understand one another's reasoning. Therefore, besides the ranking, the panelists were also asked to explain their reasoning behind the top 14 IS program complexity components they selected. In order to measure the degree of consensus among the panelists the Kendall's coefficient of concordance (W) was used, as suggested by Schmidt (1997). The Kendall's W indicates whether the consensus among the panelists has been reached, whether it increases, and its relative strength. The values of W range from 0 to 1, where numbers closer to 1 indicate a stronger consensus level (Schmidt, 1997).

The first round of phase 3 yielded a relatively weak consensus in terms of Kendall’s W (0.13). According to Singh et al. (2009), the ranking process should proceed until either “the coefficient of concordance indicated good consensus, or the level of consensus for the panel leveled-off in two successive rounds” (p. 415). Hence, we decided to conduct a second round of the third phase. In the second round, we asked the panelists to review their previous ranking based on the following information sent to each panelist via e-mail as a form of controlled feedback: the mean rank of each IS program complexity component, the ranking by that panelist for each IS program complexity component in the first round of the third phase, the panelist’s deviation from the group’s average ranking for each IS program complexity component, and a summarized list of the reason provided by the panellists as to why they considered a particular IS program complexity component more important than others (Singh et al., 2009).

Two of the 23 panelists that responded to the first round of phase 3 decided to reconsider their previous ranking, yielding a Kendall’s W of 0.15 (still a relatively weak consensus). A very important aspect of using the Delphi method is that “the researcher must know when to stop polling” (Schmidt, 1997, p. 764). The decision about whether another round of ranking must be conducted to reach a higher level of consensus should consider a trade-off between feasibility (i.e., the indulgence of the panel of experts, the time and resources required) and the possible gain that could be achieved (Schmidt, 1997; Singh et al., 2009). Following these stopping criteria, we decided not to proceed with a third round of ranking due to the drop-off of some panelists and the decrease in motivation among the panel of experts (perceived in the increasing number of reminders needed to obtain their responses). Furthermore, the results presented a leveling off of W, which, according to Schmidt (1997) indicates a “lack of progress from the previous round” (p. 771). Therefore, the polling should not continue, since further rounds could exert undue influence and provide artificial results (Schmidt, 1997).

4 Results

4.1 Components of IS program complexity

The first phase of the Delphi study resulted in 32 items related to the complexity in IS programs, which could be grouped into five distinct meta-categories. During the analysis of the consolidated list, we realized that the panelists’ answers could be sorted into the same previously mentioned meta-categories of Xia and Lee’s (2005) framework, i.e., organizational structural complexity (OSC), organizational dynamic complexity (ODC), technological structural complexity (TSC), and technological dynamic complexity (TDC). Therefore, we grouped related items accordingly. However, as some items did not fit into these meta-categories, and they were all strongly related to coordination and control issues in IS programs, we sorted them into a different meta-category named *coordination and control deficiencies* (CCD). This will be discussed in more detail below. The IS program complexity components found in the brainstorming phase are presented according to their corresponding meta-category in the following.

Construct	**Compiled components**
Meta-category: OSC	
Variety	• Variety and number of stakeholders with different perspectives, expectations, and interests • Variety and number of geographical locations • Variety and number of goals and objectives • Degree of cultural differences (i.e., national, organizational)

Interdependency	• Number of decision-making bodies with partially overlapping responsibilities • Number of interdependencies (e.g., resource, task) between different programs/projects
Meta-category: ODC	
Uncertainty	• Degree and frequency of change in scope, goals, budget, or duration during program execution • Degree of organizational change (e.g., processes, products, functions) • Degree and frequency of requirements change during program execution • Lack of payable and sufficient experience, expertise, and skills (business and IT) • Lack of outsourcing/offshoring readiness • Lack of business process and product standardization • Unwillingness to change
Ambiguity	• Unclear, ill-defined, and/or unrealistic scope, goals, target, budget, and/or duration • Unclear or ill-defined roles and responsibilities • Unclear or ill-defined requirements
Meta-category: TSC	
Variety	• Heterogeneity of IT architecture
Interdependency	• Number of functional interdependencies • Number of interfaces between IT architecture components • Fragmentation and disintegration of IT architecture
Meta-category: TDC	
Uncertainty	• Degree to which historically grown legacy systems are changed • Degree of technological change • Technological novelty and innovativeness • Criticality of the IS program/affected system • Non-adaptability and inflexibility of IT architecture
Meta-category: CCD	
Deficiencies	• Lack of strategic project portfolio management • Lack of focus on content • Lack of coordinating the social and technical elements of the system • Insufficient reliance on informal as opposed to formal communication • Information processing and communication problems • Inadequate division of scope into manageable pieces / levels of abstraction • Unclear or ill-defined program methodology

Table 5. Results of the brainstorming phase.

The newly identified meta-category of CCD can be better understood through the description of its components. In this category, one of the components mentioned most by the panelists was 'lack of strategic project portfolio management'. The panelists refer to this component as the need of a project portfolio approach in order to better manage the complexity that results from the interdependencies among projects. Projects interdependency is a component of IS program complexity and its degree of complexity increases with the lack of appropriate coordination methods to deal with it. Additionally, many panellists also mentioned 'information processing and communication problems', refering to ineffective and unstructured communication processes that keep stakeholders and program members unaligned and poorly informed about the program's current status, outcomes, and risks. Finally, the most frequently listed item of this meta-category was 'unclear or ill-defined program methodology', which was described as the lack of a solid and well-understood methodology with appropriate tools for planning, monitoring and controlling IS programs.

The main difference between the existing meta-categories of Xia and Lee's framework (OSC, ODC, TSC and TDC) and the newly identified meta-category of CCD lies in their constructs. The constructs of variety, interdependency, uncertainty and ambiguity are related to organizational and technological issues of a program and demand high program coordination as a solution approach to deal with their complexity (Bailey et al., 2010, Barki et al., 2001, McKeen et al., 1994). For example, the higher the task interdependency among projects, the more challenging the coordination of such projects will be, and therefore coordination needs to be efficient. However, if coordination strategies are not efficient, they cannot contribute as a solution approach to reduce the complexity of programs. Instead, they will be considered as additional factors that increase the overall complexity of programs. Thus, deficiencies of coordination and control are treated in the distinct meta-category of CCD. Based on the panelists' answers, deficiencies in coordination and control increase the complexity in IS programs, constituting a separate complexity meta-category that is related to, but not part of the organizational/technological structural/dynamic components of complexity.

4.2 Relative importance of IS program complexity components

As previously described, in the second phase of the Delphi study, the panelists reduced the consolidated list of 32 IS program complexity components, allowing us to obtain a manageable number of 14 challenges that could be ranked in the third phase of our study. Table 6 depicts the results of these two phases. We used the mean rank of round 2 (phase 3) to establish the final ranking of the 14 IS program complexity components.

The results of phase 2 indicate that the 14 items were selected by more than 70% of the panel of experts, and therefore they were retained for the subsequent ranking phase of this study. Moreover, through phase 2 we could also identify which of the meta-categories contribute most to the increase of complexity in IS programs. In the opinion of a panel of experts, complexity in IS programs appears to be more influenced by organizational rather than technological factors. Among the 14 most important IS programs complexity components, 10 of them are related to organizational issues. This indicates that although the technological meta-categories play a role in IS program complexity, technological elements in IS programs were not considered to cause complexity to the same degree as the elements from the organizational meta-categories. This can be understood, for example, due to the possible consideration of technological change and innovativeness as external factors that are "given", rather than elements that can be controlled or managed within the organization.

Meta-category	**IS program complexity components**	**Phase 2 Selection** % of panelists who selected this item	**Phase 3 – Ranking** Round 1 mean rank	**Phase 3 – Ranking** Round 2 mean rank	**Rank**
ODC	Unclear or ill-defined requirements	82	4.33	4.74	1
ODC	Degree and frequency of change in scope, goals, budget, or duration during program execution	74	6.56	5.61	2
ODC	Degree and frequency of requirements change during program execution	70	6.00	6.04	3
ODC	Unclear, ill-defined, and/or unrealistic scope, goals, target, budget, and/or duration	85	6.22	6.35	4
OSC	Number of interdependencies (e g , resource, task) between different programs/projects	89	7.11	6.83	5
CCD	Unclear or ill-defined program methodology	82	6.61	7.00	6
OSC	Variety and number of goals and objectives	70	7.72	7.35	7
OSC	Variety and number of stakeholders with different perspectives, expectations, and interests	93	7.71	7.48	8
ODC	Unclear or ill-defined roles and responsibilities	89	7.56	7.65	9
OSC	Number of decision-making bodies with partially overlapping responsibilities	89	7.61	8.04	10
TSC	Number of functional interdependencies	74	9.17	8.57	11
TSC	Heterogeneity of IT architecture	74	9.00	9.00	12
TSC	Number of interfaces between IT architecture components	82	9.22	9.87	13
TSC	Fragmentation and disintegration of IT architecture	74	10.11	10.48	14
	Kendall's coefficient (W)	**-**	**0.13**	**0.15**	

Table 6. Results of the selection and ranking phases of the Delphi study.

The results of phase 3 reinforce the observation made in phase 2: all organizational issues were ranked among the top 10 IS program complexity components, while all technological issues were ranked in the last positions. In the experts' reasoning for the ranking, they exemplify this by stating, "People-related factors are more important to be considered than technical constraints. Most of the complexity comes from the stakeholders and their priorities, interests, and the related changes to the program scope and targets."

Another interesting finding is that the meta-category ODC has four of its components at the top of the ranking. Hence, the panelists considered it to be the most relevant category in terms of IS program complexity. This can also be seen in the following comment: "Changes of goals or scope always lead to problems. On one hand, there is a lot of effort to be made and time to be spent in changes of plans, and on the other hand the mindset of people also has to be changed. It is not easy for program members to cope with changing of ideas previously defined."

Finally, another interesting overall result of our study is that a new meta-category emerged, i.e., CCD, which appears with one component in the final ranking, i.e., 'unclear or ill-defined program methodology'. This means that if a sound methodology is not in place in an IS program, complexity as perceived by IS program practitioners increases. This is illustrated by the following comments from our panelists: "A main driver for me is an inadequate methodology for setting up, planning, etc. which hinders one to avoid certain complexity factors", and "defined, communicated and approved methodology helps avoiding a lot of the impact factors for complexity."

5 Discussion of Findings

The purpose in this paper was to explore the specific nature of complexity in IS programs, while building upon prior research that has examined complexity in related contexts. Due to the definition of a program (see introduction) that involves the aspect of multiple involved projects, Xia and Lee's (2005) study served us in particular as a theoretical foundation. Our findings demonstrate that the meta-categories previously identified by Xia and Lee (2005) in the context of IS project complexity, i.e., OSC, ODC, TSC, and TDC, are also relevant for understanding complexity in IS programs. However, most of the individual components identified in this study were not included in previous studies of IS project complexity, and when taking a closer look at them, the unique characteristics of IS programs become salient. For example, in the meta-category OSC, the component 'number of interdependencies (e.g., resource, task) between different programs/projects' relates directly to one of the key unique characteristics of IS programs. Another related component that highlights particular challenges in an IS program context is 'variety of number of goals and objectives'. Even though an IS project may also involve a variety, our findings suggest that this component plays a more important role in an IS program in which several sets of goals and objectives of multiple projects must be coordinated in combination. In the meta-category ODC, only two out of ten identified components were previously examined in the context of IS project complexity (Xia and Lee, 2005), namely 'degree of organizational change' and 'degree and frequency of requirement change'. Our findings suggest that, as illustrated by the individual written feedback (see the excerpts provided in the previous section for example), these two change-related components are even more important in an IS program, which can be explained by the frequently involvement of organization transformation in IS programs. Furthermore, due to the strategic nature of IS programs (another key characteristic) requirements may change even more frequently than in a project context due to stronger external influences from the organization's environment. The degree of requirements change and the resulting organizational impact will arguably also be more relevant in a strategic program context. In addition to these two components, we also identified a large number of new ODC components that have not been examined in the context of IS projects.

Overall, our findings highlight the importance of organizational dynamic complexity (ODC), in IS programs. Illustrated both by the individually provided written feedback as well as the final ranking results from the third Delphi phase, ODC components appear to play the most important role in IS programs. These findings highlight the central challenge in IS programs of dealing with organizational dynamics, which we attribute directly to the known key characteristics of IS programs (Chapter 2.1). However, when looking at the final ranking results in general, it is very obvious how strongly the organizational complexity components were ranked in comparison with technological complexity components. Interestingly, all the components identified in the meta-categories TSC and TDC were also examined previously in the context of IS projects. This supports the claim that it is the organizational context in particular, not the technological context, that distinguishes an IS program from an IS project.

Besides this first key theme of the importance of organizational complexity in IS programs, ODC in particular, a second interesting theme is that a whole new meta-category emerged from our Delphi study, i.e., CCD (coordination and control deficiencies). Our overall interpretation of our data is that experts from our panel experienced significant challenges in coordinating and controlling IS programs, resulting in the perception that deficiencies in this regard relate directly to complexity in IS programs. For example, our panelists considered deficiencies concerning program methodology (planning, monitoring, controlling) as a key factor aggravating the complexity in IS programs. The reason for this is that a sound program methodology is needed in order to cope with known complexity drivers such as the variety of stakeholders, and consequently, deficiencies in this regard will lead directly to an increase in program complexity. Although future research is recommended to analyze this link more explicitly, our study suggests an overall finding that coordinating and controlling IS programs is extremely difficult. This finding is further emphasized by the written feedback provided individually by our panelists. This implies that the complexity of an IS program is increased not only by the given factors (such as variety of stakeholders, interdependencies, etc.), but also the by possibly limited means in which program members cope with complexity.

5.1 Insights for practice

This study presents valuable insights for practitioners. The list of 32 IS program complexity items identified by 28 experts clearly defines aspects and components of IS program complexity that should be taken into consideration and managed in the future. Furthermore, the ranked list of 14 IS program complexity components offers a more compact list containing only the most relevant components in the opinion of the panel of experts. This could allow practitioners to focus on the components that impact complexity the most. For example, the components related to the meta-category of ODC, which refers to the uncertainty and ambiguity present in the organizational elements of an IS program, were ranked by the experts as the most important components affecting the complexity in IS programs. Boehm (1991) argues that challenges must be identified before they can be addressed. The results obtained in this study can be used to assess and better deal with complexity in the initial planning phases of an IS program, facilitating the identification of key issues and making appropriate decisions. The impact of this study on IS program complexity is reflected in the participation of several experts in IS programs holding important positions in different companies.

5.2 Limitations and future research

Arguably the main limitation of our study concerns the weak level of consensus obtained in the results of the third phase of the Delphi study, although Schmidt (1997) outlines that “for panels of more than 10 experts, even very small values of W can be significant” (p. 771). We explain this result with two specific characteristics of our study. First, by the different nature of IS programs that the panelists have experienced in their organizations, because different organizations may define a distinct focus for their individual IS programs. For example, some organizations may have a greater focus on technological innovation, others in business processes reengineering, etc. Second, although the selected group of experts is homogeneous (i.e., they have the same nationality, similar educational background, and experience in program management), differences related to their current job positions and responsibilities (e.g., program managers, IT managers, business directors, business and technology consultants) might have led to divergent perspectives concerning the relative importance of each IS program complexity item.

The Delphi study served as a starting point to empirically conceptualize IS program complexity with important insights provided by practitioners. An important aspect for further inquiry would be to examine more intensively the relationships between the different complexity components identified in the Delphi study. Thereby, basing the examination of these relationships upon in-depth interviews with practitioners would be recommendable. Nevertheless, more research is needed with a particular focus on IS programs in order to better understand and cope with its complexity.

6 Conclusion

Our study emphasizes the importance of understanding and dealing with complexity in IS programs, which turns out to be a highly relevant, yet under-researched phenomenon. A complexity perspective on the phenomenon of IS programs reveals important differences between an IS project and an IS program and highlights key managerial challenges. The three key themes that we identified are (1) the relative importance of organizational complexity in IS programs, highlighting that technological complexity is dealt with more at the project levels and program managers have to focus particularly on organizational complexity drivers; (2) the special role of organizational dynamic complexity, highlighting the change-related challenges in IS programs driven by the external, strategic, and dynamic environment in which they usually operate; and (3) the challenges of coordinating and controlling IS programs, highlighting the connection between management deficiencies and perceptions of complexity in an IS program. In summary, we suggest that a complexity perspective on the phenomenon of IS programs may inform future empirical research on how to effectively manage IS programs. Such research is urgently needed, given the lack of studies in this area and the growing importance of IS programs in industry.

References

Baccarini, D. (1996). The Concept of Project Complexity-A Review. International Journal of Project Management, 14 (4), 201-204.

Bailey, E. D., Leonardi, M. P., and Chong, J. (2010). Minding the Gaps: Understanding Technology Interdependence and Coordination in Knowledge Work. Organization Science, 21 (3), 713-730.

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

Bechky, B.A. (2006). Gaffers, Gofers, and Grips: Role-Based Coordination in Temporary Organizations. Organization Science, 17 (1), 3-21.

Benbya, H., and McKelvey, B. (2006). Toward a Complexity Theory of Information Systems Development. Information Technology & People, 19 (1), 12-34.

Boehm, B.W. (1991).Software Risk Management:Principles and Practices.IEEE Software, 8(1), 32-41.

Burnes, B. (2005). Complexity Theories and Organizational Change. International Journal of Management Reviews, 7 (2), 73-90.

Ferns, D.C. (1991). Developments in Programme Management. International Journal of Project Management, 9 (3), 148-156.

Görög, M. (2011). Translating Single Project Management Knowledge to Project Programs. Project Management Journal, 42 (2), 17-31.

Gregor, S., Martin, M., Fernandez, W., Stern, S., and Vitale, M. (2006). The Transformational Dimension in the Realization of Business Value from Information Technology. The Journal of Strategic Information Systems, 15 (3), 249-270.

Kasi, V., Keil, M., Mathiassen, L., and Pedersen, K. (2008). The Post Mortem Paradox: A Delphi Study of IT Specialist Perceptions. European Journal of Information Systems, 17 (1), 62-78.

Levin, G., and Ward, J.L. (2011). Program Management Complexity: A Competency Model, Boca Raton: Auerbach Publications.

McKeen, D.J., Guimaraes, T., and Wetherbe, C. J. (1994). The Relationship between User Participation and User Satisfaction. An Investigation of Four Contingency Factors. MIS Quarterly, 18 (4), 427-451.

Okoli, C., and Pawlowski, S.D. (2004). The Delphi Method as a Research Tool: An Example, Design Considerations and Applications. Information & Management, 42 (1), 15-29.

Paré, J., Cameron, A., Poba-Nzaou, P., Templier, M. (2013). A Systematic Assessment of Rigor in Information Systems Ranking-Type Delphi Studies. Information & Management, 50 (5), 207-217.

Pellegrinelli, S. (1997). Programme Management: Organising Project-Based Change, International Journal of Project Management, 15 (3), 141-149.

Piccinini, E., Gregory, R. (2013).The Nature of Complexity in IS Projects and Programmes. Proceedings of the 21st European Conference on Information Systems, Utrecht, Netherlands

Ribbers, P.M.A., and Schoo, K.-C. (2002). Program Management and Complexity of ERP Implementations. Engineering Management Journal, 14 (2), 45-52.

Ross, J.W., Weill, P., and Robertson, D. (2006). Enterprise Architecture as Strategy: Creating a Foundation for Business Execution, Boston: Harvard Business Press.

Sauer, C., and Yetton, P.W. (1997). Steps to the Future: Fresh Thinking on the Management of IT-Based Organizational Transformation, San Francisco: Jossey-Bass.

Schmidt, C.R. (1997). Managing Delphi Surveys Using Nonparametric Statistical Techniques. Decision Sciences, 28 (3), 763-774.

Schmidt, R., Lyytinen, K., Keil, M., and Cule, P. (2001). Identifying Software Project Risks: An International Delphi Study. Journal of Management Information Systems, 17 (4), 5-36.

Schneberger, S.L., and McLean, E.R. (2003). The Complexity Cross: Implications for Practice. Communications of the ACM, 46 (9), 216-225.

Singh, R., Keil, M., and Kasi, V. (2009). Identifying and Overcoming the Challenges of Implementing a Project Management Office. European Journal of Information Systems, 18 (5), 409-427.

Teo, H.-H., Tan, B.C.Y., and Wei, K.-K. (1997). Organizational Transformation Using Electronic Data Interchange: The Case of TradeNet in Singapore. Journal of Management Information Systems, 13 (4), 139-165.

Xia, W., and Lee, G. (2005). Complexity of Information Systems Development Projects: Conceptualization and Measurement Development. Journal of Management Information Systems, 22 (1), 45-83.