THE DUAL ROLE OF IS SPECIFICITY IN GOVERNING SOFTWARE AS A SERVICE

Completed Research Paper

Till J. Winkler
Institute of Information Systems
Humboldt-Universität zu Berlin
Spandauer Str. 1, Berlin, Germany
till.winkler@wiwi.hu-berlin.de

Alexander Benlian
Chair of Inf. Syst. & E-Services
Technische Universität Darmstadt
Hochschulstr. 1, Darmstadt, Germany
benlian@ise.tu-darmstadt.de

Abstract

This study addresses the theoretically neglected role of information system (IS) specificity for application governance, by referring to the allocation of application-related decision authority and task responsibility between business and IT units. Based on the premise of organizational and technical 'embedding', and employing a transaction cost theoretic (TCT) lens, we develop the idea that customization and greater functional specificity of an IS lead to both more business unit governance through higher human asset specificity and more IT unit governance through higher technical specificity. Survey data from 76 organizations using different types of Software as a Service provide support for these ideas. Our results unveil a new dualism for explaining IT governance phenomena on the application level. Furthermore, we demonstrate a relevant appropriation of the frequency construct from TCT to this context. Besides practical implications, we outline contributions to IT governance, transaction cost, and IT artifact theories in the IS field.

Keywords: IT governance, Transaction cost theory, Information system specificity, Software as a Service, Partial least squares, Moderated mediation model
Introduction

Organizations today need to manage expanding information technology (IT) landscapes and a multitude of internally and externally operated business applications (O’Brien 2005). The way that each of these subsystems is governed may certainly depend on the type of application itself. For example in large firms, ERP systems are often operated and managed by centralized competence centers within the IT organization (Kremmergaard and Rose 2002; Miller et al. 2004), as opposed to emerging social software tools that tend to be managed with very strong business ownership (Deans 2011). Defining such governance arrangements between business and IT stakeholders is a common challenge for organizations, especially given the increasing amount of services that are provided through third-parties. Although the IS literature has widely recognized the importance of governance arrangements for the overall IT function (e.g., Brown and Magill 1994; Sambamurthy and Zmud 1999; Weill and Ross 2004a), few works have addressed governance arrangements specifically on the application level. This is remarkable considering that IT artifacts and their structural and social embeddedness in organizational contexts are often seen at the core of the IS discipline (e.g., King and Lyytinen 2006; Orlikowski and Iacono 2001).

Notwithstanding, the mere knowledge of the software’s category and a vague sense of its functionality are certainly not sufficient to advance towards a general understanding of governance patterns across different application types. This is mainly because applications of the same ‘type’ may still cover different functionalities (e.g., an ERP system may include some functionalities of a CRM), and in case two applications would exhibit a comparable functionality set, they can still be utilized by organizations in very different ways (Jasperson et al. 2005; Strong and Volkoff 2010). The misfits arising between the (often rather generic) functionality set of an enterprise application and the (concrete) organizational context are commonly addressed either through changes of the software (i.e., customizations) or changes to the organization itself (Sia and Soh 2007; Soh et al. 2000; Strong and Volkoff 2010) during the process of organizational ‘embedding’ (Orlikowski and Iacono 2001, p. 126). Thus, the relevant question from a research perspective is, what are the pertinent and salient properties of the entire information system that lead to different application management and governance outcomes during this process?

Transaction cost theory has evolved as a valuable lens to capture the specifics of enterprise applications—foremost in IS outsourcing contexts—by postulating an important, if not essential, characteristic of an IT artifact: asset specificity (Williamson 1979; 1981; 1985; 1991; 1996). According to this view, for example, enterprise applications that are highly specific to a company’s processes are less likely to be outsourced (e.g., Aubert et al. 2004; Benlian et al. 2009; Dibbern et al. 2005; Loebbecke and Huyskens 2006). However, the IS literature has conceptualized and measured this construct in different ways and attained mixed explanatory value from it. Especially the influence on the decision to outsource remains inconclusive in many (i.e., more than half) of the past IS studies (Karimi-Alaghehband et al. 2011; Lacity et al. 2011). Therefore some authors have demanded better appropriations of transaction cost theory (Karimi-Alaghehband et al. 2011) or even called for theories that are endogenous to IS to explain outsourcing governance phenomena (Lacity et al. 2011).

In this work, we build on the extant literature and extend the notion of asset specificity to embrace specificity characteristics of ‘ensemble’ information systems (IS), i.e. of the application including its organizational and technological context (Orlikowski and Iacono 2001, p. 125). We conceptualize IS specificity by three components (functional specificity, human asset specificity and technical specificity) and study the impact of these components on application-level governance. More precisely, we focus on the horizontal distribution of decision authority and task responsibility between business and IT units as a fundamental dimension of application governance arrangements.

---

1 With the ‘type of application’ we refer to the bundle of enterprise functions that are supported by a software application. These software types are often clustered and denominated by acronyms such as ERP (Enterprise Resource Planning), CRM (Customer Relationship Management), SCM (Supply Chain Management), HRM (Human Resource Management), MES (Manufacturing Execution Systems), CCC (Content, Communications and Collaboration Suites), etc.
As the question of application governance arrangements becomes more prevalent with increasing application outsourcing to/and sourcing from external providers (Brown 2003, p. 202), we draw on survey data from 76 organizations that use different types of software as a service (SaaS). The results of our analysis unveil the dual effects that different components of IS specificity have on application governance arrangements for SaaS. We show that greater technical embedding (technical specificity) is associated with stronger IT authority, while greater organizational embedding (human asset specificity) correlates with stronger ownership by business units. A mediation model as well as an extended model (including the influence exerted by usage characteristics) support the significance of this and other transaction cost theoretic constructs in an application governance context.

In the remainder of this paper we first explain our theoretical lens (§2), followed by the research model (§3), the description of the methodology (§4), the statistical analysis (§5) and a discussion of theoretical and practical implications (§6).

**Theoretical Lens**

Transaction Cost Theory (TCT) puts transactions, defined as the ultimate unit of economic activity (Commons 1931), and the cost for planning, adapting, and monitoring this activity into the focus to explain the existence of institutions, whether they are bureaucracies, markets or other organizational forms (Williamson 1985). Williamson uses the term transaction cost synonymously with governance cost and coordination cost and delineates these from the costs of production. According to this view, the boundary of an organization (or an organizational subunit) extends to the point where total costs for hierarchically coordinating transactions exceed the costs for coordinating via the market. Variables that influence transaction costs have been defined as asset specificity, uncertainty and the frequency of transactions, whereas asset specificity is deemed the most important (Williamson 1985, p. 52). For example, when the required assets are idiosyncratic (i.e., highly specific) and frequently required, the costs for coordinating these transactions internally, as well as uncertainty surrounding the transaction, are presumably lower than when a firm needs to find, contract out and perform the transaction with a partner on the external market. Based on TCT, organization theorists have also viewed firm boundaries as a continuous choice that may lead to diverse hybrid forms, i.e. arrangements between the polar market and hierarchy extremes (e.g., profit centers, Hennart 1993; Williamson 1991).

Assuming that boundary choices are a parallel notion to allocations of authority, TCT may also inform governance arrangements between business and IT units. Or, in the words of Whetten et al. (2009), we may “borrow” TCT as a theoretical lens and transfer it to the context of IT governance phenomena between business and IT units (cross-context) on the subsystem level of single applications (cross-level). According to this view, business units decide whether to enter into a ‘contract’ with IT units (i.e., to use the firm-internal market) or govern application operations in a hierarchical form (i.e., business unit internally) depending on application-related transaction costs. Analogously to Lacity et al. (2011, p. 146), we argue that the notion of a ‘transaction’, however, may be less adequate for the IS domain, since it implies that these internal contracts have defined entry and exit points. Information systems, in contrast, are characterized by changing requirements and a continuous evolution (e.g., through updates and enhancements) rather than discrete transactions with a certain frequency and uncertainty. This leads us to make several appropriations when casting TCT into application governance problems.

First, costs of application-related ‘transactions’ can be best understood as coordination cost, referring to the continuous efforts for planning, adapting, and monitoring application operation—as opposed to ‘production’ cost, e.g. for personnel, software licenses and operating technological infrastructure. In line with TCT, organizations seek for the best governance mode of applications to economize on total cost. Arguably, coordination costs are lower under direct coordination within business units (i.e. hierarchical coordination), while production is more efficiently coordinated by a (centralized) IT unit, due to the

---

2 SaaS refers to the use of business applications over the Internet from an external provider and is commonly seen as the highest layer in Cloud computing (e.g., Cusumano 2010). Several authors argue that SaaS and Cloud computing also pose new management challenges for business and IT units (e.g., Bento and Bento 2011; Khajeh-Hosseini et al. 2010; Winkler et al. 2011).
possibility to leverage internal economies of scale effects (e.g., through a better utilization of IT resources across the enterprise). However, when operating applications at an external party—as it is the case for SaaS—, the rationale of leveraging internal production cost advantages becomes much weaker, since the economies of scale accrue on the provider side. That is, production costs (expressed by the fees of the provider) are largely independent from the internal coordination form so that the question on application governance becomes solely dependent on the costs of coordination.

Second, the frequency of transactions, often defined as “the repetitiveness of a certain type of transaction” (Karimi-Alaghehband et al. 2011) appears less applicable to the context of long-lasting information systems. When introducing the frequency construct, Williamson originally pointed at the amount of “buyer activity in the market” (Williamson 1979, p. 247), i.e. in our context the amount of business activity in information systems use. This property, we argue, may be captured better by the scope of use of an application within the wider organization. According to TCT, the scope of use then moderates the relationship of asset specificity on application coordination cost (Karimi-Alaghehband et al. 2011; Williamson 1981). That is, given a certain degree of specificity, more business activity (i.e., a greater scope of use) will also increase the cost for coordinating the management of the application. Thus, scope of use is also expected to play a reinforcing role for the effects of specificity on application governance.

Finally, as a third appropriation, we assume that in an intra-organizational context, behavioral uncertainty defined as “strategic non-disclosure, disguise or distortion of information” (Williamson 1985, p. 57) plays a minor role compared to contractual situations with external, i.e., potentially unknown, agents on the market. Moreover, meta studies in an IS outsourcing context (Karimi-Alaghehband et al. 2011) as well as in the broader strategic management literature (David and Han 2004) document a general explanatory weakness of this variable, which corroborates the argument to omit this variable in our theoretical lens. The appropriation of the core construct of TCT, asset specificity, will be treated in the following section, after we conceptualize appropriate dimensions of application governance.

Research Model

This work aims to investigate the relationship of IS specificity on application governance arrangements. We derive appropriate constructs from the extant literature, conceptualize their meaning in the context of the governance of SaaS and hypothesize relevant relationships. The overall research model is depicted in Figure 1.

Application Governance

The notion of ‘IT governance’ has been shaped by researchers and practitioners and evolved over the last two decades. Departing from the discussions on the centralization versus decentralization of the IT function, the early IT governance literature has primarily focused on the distribution of IT decision rights between divisional and corporate IT units (e.g., Brown 1997; Brown and Magill 1994; Sambamurthy and Zmud 1999; Weill and Ross 2004a). Later—also influenced by the increasing pervasiveness of ‘best-practice’ governance frameworks such as ITIL and COBIT—authors emphasized the importance of complementary mechanisms to implement such governance arrangements, such as certain roles, defined processes, and service level arrangements (e.g., Van Grembergen 2004; Weill and Ross 2004a). While the past literature has primarily viewed IT as a bulk function, recently also more fine-grained conceptualizations of governance arrangements have been proposed, such as for data governance (Khatri and Brown 2010), development project governance (Tiwana 2009), infrastructure sourcing governance (Xue et al. 2011), and application governance (Winkler et al. 2011).

Analogously to IT governance, application governance is concerned with the design of decision rights and complementary mechanisms to deliver expected value from the use of business applications. One of the most fundamental dimensions of application governance is certainly the degree to which application-related decisions are shared horizontally within the organization, i.e. between business and IT units.
This dimension is fundamental in the sense that organizations would probably first assign the major accountabilities for managing a business application, before deciding on detailed governance processes, responsibilities, performance measurement and charging schemes. Conceivably, such internal agreements may be especially valuable also in an outsourcing context, where the “view of only two stakeholders—the client firm and the supplier firm—is too simplistic” (Lacity et al. 2011, p. 149).

The agency theoretic imperative to separate decision control from decision implementation (Fama and Jensen 1983) suggests the existence of two classes of application-related decision rights, which have been introduced elsewhere as decision authority and task responsibility (Winkler et al. 2011). This is, the principal makes control decisions regarding the management of an application which are to be implemented by an agent (Tiwana 2009). For example, as common governance frameworks suggest, application change decisions might be agreed by a change advisory board (comprising business and IT stakeholders) and implemented by a change manager on IT side (TSO 2011). In our theoretical model, however, we consider a bundle of decision rights as well as a bundle of actors (i.e., business and IT units). Thus, in the given example, we might find mixed decision authority on control level and greater IT authority on implementation level. Therefore, these dimensions can both be conceptualized as a continuum with greater ownership by one or the other side, rather than a discrete set of allocations.

Generally, there may be a certain delta between decision authority and task responsibility (e.g., business units are likely to have more decision control rights, while IT units possess more implementation decision rights). However, we argue, that these dimensions are still highly correlated (i.e., the delta is not likely to change substantially in magnitude or sign across cases), so that both dimensions together ultimately reflect an overall application governance dimension. Weill and Ross (2004b) make a similar argument. They define seven rather sophisticated patterns for distributing IT decision rights horizontally and vertically within an organization (e.g., business monarchies, IT monarchies, duopolies, federal, feudal and anarchy patterns). However, finally they conclude that all of these patterns can be ultimately arranged on a single centralization/decentralization scale. Thus, we may conceptualize the two subdimensions as first-order factors of an overall second-order application governance dimension (Edwards 2001, p. 146).

---

3 Note, that this dimension delineates from the vertical distribution or decision rights, e.g. across C-level, senior management, middle management and staff roles (Weill and Ross 2004b).
Decision Authority

Decision authority captures the distribution of application-related decision control rights between business and IT stakeholders. Over the lifecycle of a business application—i.e., from the requirements analysis, design and/or vendor selection, building and/or deployment, operation, optimization and phase out (van der Pols 2004)—a multiplicity of single decisions (e.g., on the initial investment) and recurrent decisions (e.g., on regular change approvals) accrue. For example, (Weill and Ross 2004a; 2004b) use six classes of overall IT decision rights that refer to IT principles, IT infrastructure, IT architecture, business application needs, as well as IT investment and prioritization. Arguably, from these classes the latter three appear most applicable to the application level (moreover in a SaaS sourcing context). Similar to the overall application governance dimension, we argue that the allocation of such decision rights is strongly correlated, so that they can be ultimately reduced to a single dimension. Analogously, the allocation of any application-related decision control right thus ultimately becomes a (good or bad) indicator for the decision authority subdimension.

Task Responsibility

Task responsibility refers to implementing application-related decision rights, or simply, to the execution level. Thus, this dimension is concerned with who is doing the requirements analysis, the design and/or vendor selection, the building and/or deployment, operation, optimization and phase out (van der Pols 2004). Arguably, for conventional IT applications (meaning those that are operated in-house), the scope of task responsibility is fairly wide and may include application development and integration, infrastructure operation as well as user support tasks. However, for externally operated applications—such as SaaS—few activities remain with the client organization. Susarla et al. (2010, p. 93), for example, propose a task structure for SaaS CRM services that consists of seven tasks, out of which only two (‘mapping SaaS to business requirements’, and ‘streamlining customer-facing activities’) are performed by the client organization. In our lifecycle framework these categories would largely relate to requirements analysis for initial deployment and ongoing application optimization (i.e., implementing changes). Furthermore, (Winkler et al. 2011) point out that for many SaaS offerings end-user support is an activity that remains with the client organization, and which is not allocated to IT units ‘by default’. Analogously to decision authority, we argue that such task responsibilities are likely to be strongly correlated. Therefore, any allocation of any single task responsibility would as well represent an indicator for this subdimension.

Information System Specificity

In line with Williamson’s (1996, p. 105) definition of asset specificity, we define IS specificity as the degree to which an IS can be redeployed to alternative user organizations without sacrifice of its productive value. When developing the TCT, Williamson emphasized that specificity can derive from various categories of assets, foremost the geographical location of an investment (site specificity), the employees’ knowledge, expertise and learning (human asset specificity) and the specialization of equipment and tools (physical asset specificity) (Williamson 1979; 1981).

IS researchers have mostly aggregated different facets to a single dimension, when applying TCT to study IS phenomena (see Karimi-Alaghehband et al. 2011 for a systematic review). For example, in a study on SaaS adoption, Benlian et al. (2009) operationalize application specificity as the “degree that specific applications can be customized, integrated, and modularized prior to and in the outsourcing relationship”. Dibbern et al. (2005) focus on human resource specificity to study a similar phenomenon and operationalize this construct by four subdimensions (unique business knowledge, unique software knowledge, social collaboration IS/user, social collaboration within IS). In a study on ‘netsourcing’, Loebbecke and Huyskens (2006) consider items related to technical specificity, site specificity, and human capital specificity. Aubert et al. (2004) study outsourcing of diverse IT functions and possibly use one of the broadest operationalization of asset specificity with 24 indicators relating to categories such as “client investment, human resource (HR) specificity, HR hiring delay, HR trainings delay, supplier investment, [and] structural liaison devices”.

6 Thirty Third International Conference on Information Systems, Orlando 2012
The explanatory value that IS researchers have obtained from this variable, however, remains mixed. More precisely, meta-reviews demonstrate that in more than half of the studies the construct does not produce the hypothesized relationships (Karimi-Alagheband et al. 2011; Lacity et al. 2011). In the given examples, Benlian et al. (2009) as well as Loebbecke and Huyskens (2006) find no significant relationships, Dibbern et al. (2005) obtain partly significant relationships (the transaction cost hypothesis, however, is not significant with t=1.34), and Aubert et al. (2004) find a significant opposite hypothesized relationship regarding their multi-faceted construct. The authors partly explain the low impact of asset specificity by the “definition and operationalization” of the variable (Loebbecke and Huyskens 2006, p. 421) and “measurement problems” (Aubert et al. 2004, p. 929). We put forward that IS specificity can be decomposed into three major factors referring to functional, human, and technical categories. This conceptualization is based on the idea that an IS is gradually ‘made specific’ while it is embedded in its organizational and technical context, so that the facets of IS specificity mutually influence each other.

Functional Specificity

The core role of an enterprise IS is to provide certain functionalities that typically map to one or several business processes of an organization (O’Brien 2005). Since organizations are inherently different, packaged applications, such as ERP, typically do not perfectly match the organizations’ process, data and user requirements (Sia and Soh 2007; Soh et al. 2000; Strong and Volkoff 2010). Strong and Volkoff (2010) emphasize that two different kinds of misfits can arise: deficiencies which result from required functionality that is actually missing, and impositions that arise from additional functionality that organizations implement to enable new work practices (‘best practices’). The former (deficiencies) are typically overcome by adapting the application to the organization, which can vary strongly in the degree and effort put into it. (Glass 1998) differentiates customization (or configuration), extension, and modification. We posit that the degree of adaptation to company specific requirements directly expresses the degree of specificity of the overall IS inasmuch as a highly customized system according to our definition can be less easily redeployed to an alternative user organization without sacrifice of its productive value.

Important to note, customizations are not limited to conventionally deployed IS. Although SaaS applications inherently comprise a very standardized set of functionalities, vendors increasingly provide ways to enable self-service adaptation and extension, e.g. through web-based configuration (instead of editing local configuration files), use of components from platform markets (instead of locally installing add-ons) and integration through standard application programming interfaces (APIs) (Bezemer and Zaidman 2010; Sun et al. 2008; Xin and Levina 2008). For example—just like in any other software implementation project—SaaS consultants typically write up a ‘requirements specification’ document, which can take few days or up to several months depending on the specific case (Winkler and Günther 2012). We argue that this degree of functional specificity has both organizational and technical implications. As our first hypotheses, we pose:

**H1:** Higher functional specificity of an IS leads to greater human asset specificity, and

**H2:** Higher functional specificity of an IS leads to greater technical specificity.

Human Asset Specificity

According to Williamson (1981, p. 555), human asset specificity is associated with an organization’s work routines and primarily “arises from learning by doing”. For example, although programming skills may be improved by practice, they are still regarded as less specific inasmuch as they are equally valued by the current and potential other employers. In contrast, the knowledge of a firm’s business process is regarded as highly specific as it cannot be instantly transferred to other organizations. Functional specificity may lead to greater human asset specificity (H1), since the work routines that are created in overcoming information system misfits, i.e. through both customizing the application to own routines and enabling new routines, will also require greater firm-specific knowledge. In other words, in the process of specifying and customizing new software to an organization, those additional functionalities that are added to the software also require greater knowledge and skills from users. This knowledge comprises
“knowledge of unique business processes and application software that is specifically customized to a company” (Dibbern et al. 2005). Such skill requirements could be reflected, for example, in the amount of user trainings given and the need to specialize and educate staff for managing this application.

The degree of human asset specificity of an information system (i.e., the application including its users and personnel responsible for its management), in turn, is likely to affect the mode of application governance. The more company-specific knowledge is incorporated in an information system, the more involvement of those organizational units is expected that make use of the information system—which are commonly the business units. Or, from a TCT view, highly specific knowledge needs to be “embedded in a protective governance structure” to economize on transaction costs for acquiring and maintaining this knowledge (Williamson 1981, p. 563). Altogether, this argument implies that more customized and functionally specific applications cause higher IS human asset specificity and therefore more business involvement in application-related decision authority and task responsibility. Accordingly we pose that

\[ H3: \text{Higher human asset specificity of an IS leads to greater business unit application governance, and} \]

\[ H3': \text{Human asset specificity mediates the effect of higher functional specificity on greater business unit application governance.} \]

**Technical Specificity**

Technical specificity has been considered explicitly by few IS researchers (Loebbecke and Huyskens 2006) and has thus so far only been weakly conceptualized. However, we may relate this dimension to physical specificity, referred to by (Williamson 1979) as the use of special equipment or tools in an organization to produce certain goods or services. Increased functional specificity of a business application may also lead to higher technical specificity (H1) since some functionality may require a special technical implementation. For example, in the case of a SaaS CRM implementation, a company built a special buffer-acknowledge database to exchange data, such as offers, orders, bills, delivery receipts, products and prices, in both ways between the CRM and the ERP systems (Winkler and Günther 2012). In the view of TCT, such technical integration approaches reduce the possibility of the IS to be redeployed to alternative user organizations.

For conventionally deployed information systems, technical specificity may refer to a number of different categories, e.g. the need of specific custom programming, databases, and server infrastructure. In a SaaS context, where the gross of technology is outsourced, the technical dimension largely refers to the integration with other backend systems (Benlian et al. 2009; Bezemer and Zaidman 2010; Sun et al. 2007). Thus, for SaaS, we might also term this construct simply as integration specificity. Arguably, higher technical (integration) specificity might be related with stronger application governance through IT units to minimize internal coordination costs. For example, in the given case the technical interfaces created “some discernible efforts also for the ERP team” (Winkler and Günther 2012). Altogether this adds an argument to our conceptual model that greater functional specificity also increases technical specificity, which in turn leads to more IT involvement in application-related decision authority and task responsibility. Accordingly we pose

\[ H4: \text{Higher technical specificity of an IS leads to greater IT unit application governance, and} \]

\[ H4': \text{Technical specificity mediates the effect of higher functional specificity on greater IT unit application governance.} \]

**Scope of Use**

Scope of use relates to the frequency construct in TCT and can be understood as the breadth to which a business application is used within an organization. This scope may be expressed by the share of users of an application or simply by the amount of organizational units in which it is utilized (e.g., on a scale from single-department to company-wide use). This construct does not relate to any of the six categories of system usage provided by Burton-Jones and Straub (2006), since it captures ‘lean’ system usage characteristics on the organization, rather than on the individual level.
As (Karimi-Alaghehband et al. 2011, p. 135) note, some IS studies that use TCT as a theoretical lens do not account for the frequency construct and/or its interaction effects. In line with TCT, we assume that scope of use acts as a moderator (or more precisely multiplicator) of the effects of functional specificity on coordination cost imposed by both human asset and technical specificity. This is, while functional specificity leads to according human asset and technical specificity, the strengths of these relationships are influenced by scope of use. Consider the examples from above: a highly customized system will require more user trainings, whereas the need for such trainings logically still increases with the number of users that actually need to be trained. Same applies to technical specificity: the highly customized CRM system causes certain initial as well as continuous integration efforts, i.e. technical specificity. These efforts would be arguably even higher, the more employees use the system, e.g. since more data needs to be handled and more change requests turn up. Accordingly we derive that

H5: Scope of use moderates the relationship of functional specificity and human asset specificity.

H6: Scope of use moderates the relationship of functional specificity and technical specificity.

Methodology

To test the proposed research model, we used data from a survey with 76 organizations that provided information about a SaaS application in use. In the following we describe our approach for the development of the measurement instrument and acquisition of the sample.

Instrument Development

Our approach for developing the measures has been oriented in the procedure propounded by Churchill (1979). The domains and subdomains of application governance and IS specificity had been initially described and specified in course of a number of explorative case studies presented elsewhere (Winkler et al. 2011). We compared these case findings with the extant literature and derived a number of items for each of the six presented constructs (plus further constructs not included in this research). In this manner, items for functional specificity (Benlian et al. 2009), human asset specificity (Dibbern et al. 2005) and decision authority (Weill and Ross 2004b) have been derived from literature, while items for technical specificity, task responsibility and scope of use can be regarded as entirely new measures. In order to validate these measures, we first conducted a category sorting with 8 fellow researchers, who had to assign the items according to given construct definitions, and eliminated ambiguous measures.

Second, we conducted and online pretest with 29 IT professionals from industry, consulting and research, which resulted in acceptable construct reliabilities (alpha>0.8). Due to size constraints for the final questionnaire, we needed to further reduce the number of items to 3 for five-point-scaled and 2 for seven-point-scaled items. Five-point scales were used for the business/IT application governance items (question: who decides/ who is responsible; scale anchors: business, business with IT involvement, business and IT equally, IT with business involvement, central IT) due to their adequate semantic granularity discussed in related works (Brown and Magill 1994; Winkler et al. 2011). Seven-point scales (very low, low, low-medium, medium, medium-high, high, very high) were used for the other measures. The final questionnaire has been pretested in think-aloud meetings with 3 different CIOs and passed only minor revisions. The measurement items are provided in the Appendix.

Sample Description

The survey was sent out to 2,886 large-sized companies in Germany (i.e., companies with >50 m Euros of revenues, >500 employees, from non-public sector) accompanied by a formal invitation, a return envelope, as well as references to the online version of the questionnaire and the project website. Relevant addresses and CIO contacts had been retrieved from a commercial publisher of company information. Besides providing the survey results, we were able to offer a small gadget for the first 100 respondents and the drawing of a tablet computer as further incentives. During a six-week response period in April/May 2011, we received feedback from total 534 companies, out of whom 220 provided usable responses (60% online, 40% paper-based, total response rate 7.6%).
In the main part of the questionnaire, the respondents were asked to provide details about one business application they are knowledgeable about. 131 of these respondents provided information about a conventional application, and 76 about a SaaS in use (13 were discarded from further analyses due to unclear application types or inconsistent answers). Regarding the SaaS subsample, respondents stated to be CIOs (27.6%), senior IT managers (39.5%), IT managers (23.7%), IT staff (2.0%), and senior business managers (6.6%) with an average 12 years of work experience in their current firms (median 10 years). T-tests revealed no response bias in the model variables neither regarding business versus IT roles, nor regarding early versus late responses. The respondents provided details about a wide range of common SaaS application types, in line with the categories introduced earlier, these can be characterized as ERP (13), CCC (12), CRM (10), SCM (6), HRM (5), MES (4), business intelligence and analytics (6), office and productivity (5), service management (3), and other (12) application types.

Analysis

For the purpose of model tests we employ partial least squares (PLS), a widely used structural equations modeling technique. The choice for PLS was primarily motivated by the explorative character of this research including various new constructs as well as the good ability of PLS to handle moderating effects (Chin 1998; Hair et al. 2011). In order to isolate the effects constituted by adding scope of use as an additional TCT construct, we test three variants of the proposed research model separately: M1 contains only the core constructs, i.e. functional specificity (FuncSp), human asset specificity (HumSp), technical specificity (TechSp), as well as application governance (AppGov), decision authority (DecAuth) and task responsibility (TaskResp) and suffices to analyze the core model including the hypothesized mediation effects. M2 integrates the scope of use (Scope) and its direct effect on HumSp and TechSp. M3 uses a product indicator approach (Henseler and Chin 2010) and includes the interaction terms (i.e., the Cartesian product indicators of Scope×FuncSp and Scope×TechSp) to test the hypothesized moderating effects. Significances were assessed based on the pseudo t-values from a bootstrapping procedure with 1,000 re-samples. We follow the recommended approach to assess the measurement model and structural model separately (Chin 1998).

Measurement Model

We use M2 to assess the psychometric adequacy of our measurement instrument since this model variant contains all relevant indicators (i.e., includes Scope). Convergent validity of the first order constructs is supported by acceptable values for Cronbach’s alpha (alpha>0.8), composite reliability (CR>0.7) and average variance extracted (AVE>0.5)(Hair et al. 2011), see Table 1. Furthermore, all indicator loadings are significant and clearly over the threshold of 0.7, see Appendix. This suggests that the indicators are sufficiently correlated and reflect the properties of their substantive constructs. For the reflective second-order construct (AppGov), convergent validity can be assessed by calculating the AVE of its subdimensions (MacKenzie et al. 2011, p. 315). With two highly significant path coefficients to DecAuth (.920, R²=.847) and TaskResp (.936, R²=.877), see Table 2, this second-order AVE amounts to highly an acceptable AVE of .862.

We analyze discriminant validity by the Fornell-Larcker criterion, indicator cross-loadings, and an exploratory factor analysis (EFA). Fornell and Larcker (1981) suggest that the squared latent variable correlations should be less than AVE for each construct to ensure support validity, which is the case for our model, see Table 1. Indicator cross-loadings are as well lower than loadings by their substantive constructs, see Appendix. However, we note that some loadings within the application governance domain, as well as within the IS specificity domain are above the threshold of 0.7, indicating that—statistically—these cannot not necessarily be viewed as separate dimensions. After varimax rotation, the EFA of our model variables (eigenvalue>1; KMO=.816) yields three clearly distinguishable factors relating to the indicators of application governance (30.8% of explained variance), IS specificity (24.15%), and scope (17.49%). This supports our nomological framework inasmuch as we stated that both DecAuth and TaskResp are subdimensions of application governance. Likewise, FuncSp, HumSp and TechSp can be viewed as components of overall IS specificity (which, based on our particular research interest, has not been explicitly modeled here).
Finally, we assess common method bias (CMB) by using a latent method factor approach for PLS. Especially in self-reported data where independent and dependent variables stem from the same source, CMB may arise due to the subjects’ motif to give consistent and/or socially desirable answers (Podsakoff and Organ 2003). Following the procedure described by Liang et al. (2007, pp. 85-87), we add a method factor to the model to account for this bias and compare the average variance presented by this factor with the variance presented by the substantive constructs. To account for each indicator only once, we omit the reflective first-order constructs and test the method influence directly on the six indicators of AppGov. The method factor accounts for .009 of the average variance while the substantive constructs explain .799 (ratio 1:88), suggesting that CMB is unlikely to affect our results.

### Table 1. Measurement model validity

<table>
<thead>
<tr>
<th>Operationalization</th>
<th>Convergent validity</th>
<th>Discriminant validity (Root AVE on diagonal)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Alpha</td>
<td>CR</td>
</tr>
<tr>
<td>DecAuth</td>
<td>3 Bus./IT</td>
<td>1-5</td>
</tr>
<tr>
<td>TaskResp</td>
<td>3 Bus./IT</td>
<td>1-5</td>
</tr>
<tr>
<td>FuncSp</td>
<td>2 low/high</td>
<td>1-7</td>
</tr>
<tr>
<td>HumSp</td>
<td>2 low/high</td>
<td>1-7</td>
</tr>
<tr>
<td>TechSp</td>
<td>2 low/high</td>
<td>1-7</td>
</tr>
<tr>
<td>Scope</td>
<td>2 low/high</td>
<td>1-7</td>
</tr>
</tbody>
</table>


### Table 2. Model tests

<table>
<thead>
<tr>
<th>Hyp. Path</th>
<th>M 1 (mediation model)</th>
<th>M2 (including Scope)</th>
<th>M3 (moderated med.)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Coefficient</td>
<td>$f^2$</td>
<td>Coefficient</td>
</tr>
<tr>
<td>- FuncSp→AppGov $^1$</td>
<td>-.010**</td>
<td>-.017</td>
<td>-.0145**</td>
</tr>
<tr>
<td>H1 FuncSp→HumSp</td>
<td>.736**</td>
<td>-</td>
<td>.691**</td>
</tr>
<tr>
<td>H2 FuncSp→TechSp</td>
<td>.541**</td>
<td>-</td>
<td>.404**</td>
</tr>
<tr>
<td>H3 HumSp→AppGov</td>
<td>-.105**</td>
<td>.008</td>
<td>-.121**</td>
</tr>
<tr>
<td>H3' FuncSp→HumSp→AppGov $^2$</td>
<td>-.077**</td>
<td>.878</td>
<td>-.084*</td>
</tr>
<tr>
<td>H4 TechSp→AppGov</td>
<td>.235**</td>
<td>.042</td>
<td>.229**</td>
</tr>
<tr>
<td>H4' FuncSp→TechSp→AppGov $^2$</td>
<td>.128**</td>
<td>1.086</td>
<td>.094**</td>
</tr>
<tr>
<td>- Scope→HumSp</td>
<td>-</td>
<td>-</td>
<td>.128**</td>
</tr>
<tr>
<td>- Scope→TechSp</td>
<td>-</td>
<td>-</td>
<td>.419**</td>
</tr>
<tr>
<td>H5 Scope×FuncSp→HumSp</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>H6 Scope×FuncSp→TechSp</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>- AppGov→DecAuth</td>
<td>.9217**</td>
<td>-</td>
<td>.920**</td>
</tr>
<tr>
<td>- AppGov→TaskResp</td>
<td>.9371**</td>
<td>-</td>
<td>.9364**</td>
</tr>
</tbody>
</table>

$^p<.05; \quad ^* p<.01; \quad ^{**} \text{ not significant}$

$^1$ direct path for comparison and VAF calculation, not included for further model tests

$^2$ total mediated effect, significance according to Sobel test, variance accounted for (VAF) reported instead of effect size

### Structural Model

The results of the structural model tests are provided in Table 2. Path coefficients and significances can be interpreted similar to coefficients in a simple regression; effect sizes $f^2$ express the relative change in the explained variance of an endogenous construct when eliminating one of its antecedent variables from the

Thirty Third International Conference on Information Systems, Orlando 2012 11
model (Chin 1998, p. 317). Analogously to regression analyses, effect sizes of 0.02, 0.15, and 0.35 can be regarded as small, medium, and large, respectively (Cohen 1988, p. 412). For the mediation effects, we report the specific indirect effect (i.e., the product of the path coefficients of the mediation) (Hayes 2009, p. 3) and its significance assessed by a Sobel (1982) test statistic; in the effect size column, we alternatively report the variance accounted for (VAF) of the specific indirect effect relative to the specific total effect (i.e., the total effect excluding the opposite mediation effect). In simple mediation models, values for VAF range from 0.0 (no mediation) to 1.0 (perfect mediation), while values greater 1.0 demonstrate a suppressor effect (Shrout and Bolger 2002, p. 430). In the following, we will evaluate our hypotheses successively.

Worth to begin with, we find that the direct effect from FuncSp to AppGov is close to zero and not significant (for none of the model variants). The effect size of FuncSp is even negative, which shows the rare case that excluding this construct from the model even increases explained variance in AppGov. (This will be explained more in detail in the following.)

The paths FuncSp→HumSp and FuncSp→TechSp are both strongly significant, which confirms H1 and H2, and underlines the nomological argument made earlier that these constructs are ultimately facets of an overall IS specificity dimension. When including Scope and interaction variables in M2 and M3, we note that the strength of both paths decreases only moderately so that support for these hypotheses is sustained. Accordingly, the effect sizes of this construct compared to the influence exerted by Scope are considerably strong.

We find partial support for H3 and the path HumSp→AppGov. This is, in the simple mediation model (M1) the path is negative as hypothesized, yet, not significant at the 0.05 level (t=1.573). However, when adding Scope and the interaction variable to the models M2 and M3, the path coefficient increases slightly and becomes statistically significant (t=1.777). We take this as evidence that including the Scope variable increases the overall explanatory power of the model, although the effect sizes presented by HumSp remain extremely weak. Similarly, we find that HumSp significantly mediates the effect from FuncSp to TechSp (H3') when Scope is included into the model (M2 and M3). The high VAF ratio (>0.8) expresses that this mediation effect is comparably strong to the non-existent direct effect (FuncSp→AppGov).

Analyzing the role of TechSp, we find also support for our hypotheses H4 and H4'. TechSp has a clear and significant impact on AppGov across all three model variants with small, but considerable effect sizes. Furthermore, as hypothesized, TechSp significantly mediates the effect from FuncSp to AppGov in opposite direction to the mediation exerted by HumSp. Thus, we find an explanation for why there is no direct effect from FuncSp to AppGov: While HumSp negatively exerts the effect from FuncSp on AppGov, TechSp positively counteracts it. For this reason the total effect (i.e., the sum of the direct and indirect effects) are close to zero across all three model variants. Or, in the words of Hayes (2009, p. 10), we provide a case where “two or more indirect effects with opposite signs can cancel each other out, producing a total effect and perhaps even a total indirect effect that is not detectably different from zero”.

Regarding the hypothesized moderating effects and the product indicator approach, we first add Scope as a second antecedent of HumSp and TechSp to the model (M2). This yields in significant paths and a considerable increase in explained variance for TechSp (f²=0.15). However, when introducing the two interaction terms (Scope×FuncSp), the coefficient Scope→TechSp is reduced and Scope→HumSp is practically absorbed. Coupled with the finding that the interaction terms are significant for both relationships (with considerable effect sizes), we can say the moderating hypotheses H5 and H6 are both supported, which underlines the explanatory power provided by the Scope of use construct.

---

4 Hayes (2010, p. 14) adds for consideration that “the quantification of effect size in intervening variable models remains a cutting-edge area of thinking and research” where “none of the methods proposed thus far are particularly satisfying”. Since total effects can approach zero and even have opposite signs than the denominator, any ‘VAF ratio’ should be interpreted with caution.
Discussion

This work was motivated by the principal argument that the fundamental characteristics of ‘ensemble’ information systems (IS) are not yet sufficiently understood, so that we still fall short in explaining individual application governance outcomes. We employed a transaction cost theoretic lens and hypothesized that different components of IS specificity (i.e., functional specificity, human asset specificity, and technical specificity) lead to different and partially contrary application governance patterns. Our hypotheses are grounded in the rationale of organizational and technical embedding. That is, enterprise IS (including packaged applications as well as SaaS) are typically first functionally specified before they are technically implemented and taken in use in an organizational context.

We find empirical evidence for our argument that greater technical embeddedness (represented by technical specificity) leads to greater IT involvement in application-related decisions and tasks (H2, H4, H4'), while greater organizational embeddedness (represented by human asset specificity) is likely to induce more involvement of business units (H1, H3, H3'). This unveils a theoretically interesting, and practically relevant new dualism regarding the contradictory role of IS specificity for explaining governance phenomena. Furthermore, we find strong support for the relevance of considering the scope of use of an IS as a moderator in these relationships. Consequently, the more employees use an IS, the greater is the impact of the systems functional specification on both the extent of organizational and technical embedding of the IS—thus the need to define appropriate governance arrangements. We see three major directions of theoretical impetus provided by this research.

Contribution to IT Governance Theory

Our findings contribute to past research on IT governance since they reconfirm the need to balance decision rights between different stakeholders involved in IT related decision making (e.g., Brown 1997; Brown and Magill 1994; Sambamurthy and Zmud 1999; Weill and Ross 2004a). In addition to stressing the need for balancing decision rights and proposing appropriate new governance dimensions—i.e., decision authority and task responsibility between business and IT units—this research provides reasons for why application governance arrangements vary. We find these reasons (at least partly) in the specificity characteristics of different information systems. This represents a new contribution inasmuch as (a) this is (to the knowledge of the authors) the first work to employ a purely transaction cost theoretic (TCT) lens for investigating internal IT governance phenomena (i.e., excluding outsourcing). TCT thus emerges as a particularly valuable lens to capture the different facets of IS specificity that play a role in IT governance contexts.

(b) Past IT governance research has focused on business related contingencies (such as firm strategy, size and diversification) and thus largely neglected the technological antecedents (Brown and Grant 2005, p. 704). Just more recently, authors have emphasized the complementarities between the technical and organizational architecture (Tiwana and Konsynski 2010). Thus, our findings add a new dualism to this complementary view. (c) As a result, we particularly advance in the understanding of IT governance phenomena on the application level. Such modular view, i.e. disaggregating ‘the bulk IT function’ along its technological architecture, appears long overdue given the increasing dispersion of information systems within, and outside of, organizational boundaries.

Contribution to Transaction Cost Theory in IS

We put forward, that our appropriation of TCT to an IS context, and particularly the distinctive view of IS specificity as a construct composed of different facets, may also provide a valuable lens to address other IT governance phenomena, particularly related to IS outsourcing. In line with Williamson (1979; 1981; 1985; 1991; 1996), we argued that application-related transaction cost (or better: coordination cost) can arise from different types of assets (here: functionality, human assets, and technological integration). However, in contrast to Williamson we do not argue that one single form of governance (e.g., hierarchical coordination) is optimal to economize on all types of coordination cost. Instead, we argued that, for example, costs arising from human asset specificity are economized through business coordination while costs arising from technical specificity are economized through stronger IT coordination.
This proposition is in line with prior organization researchers, who have emphasized that different kinds of transaction cost can arise from the possible forms of governance (Hennart 1993). While this seems to be a very simple logic, such multi-faceted understanding of IS specificity might provide a new key to resolve some of the inconsistencies regarding past TCT research in studying IS outsourcing phenomena (Karimi-Alaghehband et al. 2011; Lacity et al. 2011). Transferring our logic (back) to the domain of IS outsourcing, this could imply that human asset specificity is handled more efficiently under hierarchical coordination (i.e., in-house) while (at least some aspects of) technical specificity can be handled better through market coordination (i.e., by providers who are potentially more specialized to economize on production and technical coordination costs). The presented examples from the TCT/IS outsourcing literature (Benlian et al. 2009; Dibbern et al. 2005) provide at least a first indication for this argument. In contrast, aggregating a multitude of facets into a single dimension, as in (Aubert et al. 2004), potentially aggravates the prevailing ambiguities around asset specificity in an IS context.

As a way forward, future TCT/IS studies might consider the issue of multidimensionality of the IS specificity construct. Researchers might also revise past data in this regard, and thus exploit the body of “empirical IS studies carried out on IT outsourcing over the last nearly 20 years” (Lacity et al. 2011, p. 148). This perspective, however, contradicts the immediate claim to “discard constructs from reference disciplines that have not been empirically robust in the ITO context, like asset specificity from [TCT].” (Lacity et al. 2011, p. 149). The aim of this study was rather to provide a way for a better appropriation of this reference theory to the IS field (Karimi-Alaghehband et al. 2011). In this sense, we also demonstrated how we can interpret the frequency construct in an IS context as a scope of use. Ultimately, we expect a better understanding of ‘IT artifact’ specificity, its interaction with other TCT variables, and its impact on (internal and external) coordination mechanisms. Such understanding may be particularly helpful in the beginning era of SaaS and Cloud computing (Bento and Bento 2011; Khajeh-Hosseini et al. 2010).

**Contribution to Theories of the IT Artifact**

We believe that our distinct approach also allows us to position our findings in a wider context regarding the conceptualization of the IS/IT artifact. Since the call from Orlikowski and Iacono (2001), IS researchers have widely debated about conceptualizations of the IT artifact (e.g., in King and Lyytinen 2006). Strong and Volkoff (2010) theorize on a conceptual model originally propounded by Wand and Weber (1990), which characterizes the IT artifact to consist of deep, surface, and physical structures. Surface structures represent the facilities that allow users to interact with the system, i.e. the user interface and its usability criteria. Deep structures represent things, properties, states and transformations of real-world systems, i.e. functionality and data; and physical structures refer to the underlying technology. Some readers may also recognize these elements as presentation, logic, data and hardware layers in software engineering (e.g., Fowler 2003, pp. 17-24).

Strong and Volkoff (2010) extend this conceptualization by a fourth element that they refer to as “latent structures” containing roles and control— which we may consider as core elements of IT governance—as well as organizational culture. As the authors argue, “the latent structures emerge from and depend on the other three structures”, or in other words, the arrangements for governing and managing IT artifacts are inherently related with the usability, functionality, data, physical configuration, and potentially other properties of an IS. However, even studies taking an ensemble view “focus on what the technology is connected to, while black-boxing the structures that constitute the technology itself”, as the authors continue (Strong and Volkoff 2010, p. 752).

The study by Strong and Volkoff (2010) (which can be regarded as a type II theory, according to the taxonomy by Gregor (2006)), makes a valuable point by postulating an integrated ensemble view of latent structures and the IT artifact. However, it falls short in developing testable propositions to “learn to design [such] latent structures” as the authors acknowledge (p. 752). We suggest that the dimensions of IS specificity set forth in our study (foremost functional and technical specificity) can be well regarded as measurable properties of the deep structures of an IS, while the dimensions of application governance clearly relate to the latent structures. In this sense, our study provides first causal explanations and tested propositions on the relationship of deep and latent structures. This, we argue, also advances our understanding of ‘IT artifacts’ from a type II towards a type IV theory (Gregor 2006).
**Practical Implications**

This research also provides relevant insights for practitioners, both in business and IT positions. First, our model facilitates a contingent understanding on how management and governance patterns for single applications depend on the type of application itself. Taken the examples from the outset, this would mean that the management of conventional ERP systems is best organized in central IT competence centers due to their high functional and technical specificity (Kræmmergaard and Rose 2002). In contrast, emerging social software tools can, and potentially should, be given to the hands of business stakeholders, since/if they are operated as largely isolated applications (i.e., technically unspecific) and deal with highly company-specific knowledge (i.e., human asset specificity) (Deans 2011).

As our data was based on the assessment of various SaaS applications, the results specifically provide a rationale how to design such governance arrangements in the ‘Cloud’ era. We argued that, under some conditions, it can be useful to loosen the imperative of ‘strictly centralized’ IT control known from many traditional enterprise IS. This implies that stakeholders and IT decision rights will be increasingly dispersed across the organization. On the other hand, business practitioners may find our insights helpful to realize that even for SaaS, not only in case of high technical integration, some coordination with IT units remains a key necessity. Thus, altogether our ‘dualist’ view strengthens the importance of aligning business with IT (and vice versa), all the more in the era of Cloud-based computing.

**Limitations and Future Work**

Three limitations of this study merit consideration, foremost related to issues of operationalization, generalizability, causality and endogeneity. First, due to practical restrictions, we operationalized constructs with a comparably low number of items. Arguably, a semantically broader operationalization may have improved predictive validity of our research model (Diamantopoulos et al 2012). Second, our sample is based on data from large-sized enterprises in Germany using SaaS applications. Thus, our generalization to any other macro and micro context is purely argumentative. Third, the cross-sectional analysis can only ascertain association, not the causal ordering that derives from our theoretical argument. Conversely, some governance arrangements may also influence specificity characteristics of the ensemble information system that was subject of this research.

We see multiple directions for further research. First, as the conceptualization of the IS specificity construct is still in an exploratory phase, future works may unveil further subdimensions and develop semantically broader measures. According to the presented conceptualization of the IT artifact, deep structures (e.g., data) and surface structures (e.g., usability) represent promising characteristics that could be dimensionalized in a higher-order theoretical model. In addition, as suggested earlier, researchers in the IT outsourcing field might find it helpful to reconsider the demonstrated multidimensionality of IS specificity to isolate further potentially opposing effects. Finally, future works should also address the question in how far the proposed dualism, besides for SaaS, also holds for the domain of traditional delivery models. As we motivated at the outset, a comprehensive understanding of IS specificity might provide a way to unite the important, yet separate research streams of internal IT governance and external outsourcing governance towards a broader understanding of IT governance phenomena in expanding IT landscapes.

**References**


### Appendix – Items and Cross-loadings

<table>
<thead>
<tr>
<th></th>
<th>FuncSp</th>
<th>HumSp</th>
<th>TechSp</th>
<th>Scope</th>
<th>DecAuth</th>
<th>TaskResp</th>
</tr>
</thead>
<tbody>
<tr>
<td>The degree to which this application has been customized to the company's processes</td>
<td>.949</td>
<td>.722</td>
<td>.493</td>
<td>.272</td>
<td>.055</td>
<td>.015</td>
</tr>
<tr>
<td>The effort for adapting this application to the company's needs</td>
<td>.945</td>
<td>.660</td>
<td>.517</td>
<td>.316</td>
<td>.025</td>
<td>.004</td>
</tr>
<tr>
<td>The amount of trainings that were provided to the users</td>
<td>.705</td>
<td>.931</td>
<td>.546</td>
<td>.312</td>
<td>.006</td>
<td>-.024</td>
</tr>
<tr>
<td>The need for personnel with specialized skills to manage this application</td>
<td>.643</td>
<td>.918</td>
<td>.344</td>
<td>.322</td>
<td>-.022</td>
<td>.007</td>
</tr>
<tr>
<td>The degree to which the application is technically integrated with other applications</td>
<td>.473</td>
<td>.386</td>
<td>.945</td>
<td>.522</td>
<td>.302</td>
<td>.151</td>
</tr>
<tr>
<td>The number and types of interfaces (none, one-way, two-way, several, highly-integrated)</td>
<td>.534</td>
<td>.533</td>
<td>.943</td>
<td>.504</td>
<td>.128</td>
<td>.027</td>
</tr>
<tr>
<td>The scope of use of this application within your company (single org. unit / company-wide)</td>
<td>.097</td>
<td>.111</td>
<td>.328</td>
<td>.823</td>
<td>.505</td>
<td>.396</td>
</tr>
<tr>
<td>The share of people in the company who use this application</td>
<td>.373</td>
<td>.410</td>
<td>.580</td>
<td>.963</td>
<td>.445</td>
<td>.388</td>
</tr>
<tr>
<td>Who decides on... Changes to this application (e.g.. approvals of a change request or customizations)</td>
<td>-.024</td>
<td>-.037</td>
<td>.154</td>
<td>.476</td>
<td>.844</td>
<td>.563</td>
</tr>
<tr>
<td>...IT spending for this application (e.g., for licenses. enhancements. Maintenance)</td>
<td>.079</td>
<td>.039</td>
<td>.271</td>
<td>.452</td>
<td>.924</td>
<td>.655</td>
</tr>
<tr>
<td>...Architecture issues regarding this application (e.g., integration with other systems, infrastructure)</td>
<td>.050</td>
<td>-.028</td>
<td>.165</td>
<td>.391</td>
<td>.834</td>
<td>.664</td>
</tr>
<tr>
<td>Who is responsible for... Changes to this application (e.g., doing customizations. implementing a change)</td>
<td>-.030</td>
<td>-.065</td>
<td>.037</td>
<td>.372</td>
<td>.714</td>
<td>.925</td>
</tr>
<tr>
<td>...1st level support for this application (e.g., answering user requests, incident management)</td>
<td>-.005</td>
<td>.020</td>
<td>.096</td>
<td>.445</td>
<td>.649</td>
<td>.927</td>
</tr>
<tr>
<td>...2nd level support for this application (e.g., finding technical errors, problem management)</td>
<td>.065</td>
<td>.019</td>
<td>.133</td>
<td>.364</td>
<td>.643</td>
<td>.921</td>
</tr>
</tbody>
</table>