Journal of Information Technology (2018) 33, 151–170
© 2017 Association for Information Technology Trust All rights reserved 0268-3962/18
palgrave-journals.com/jit/

Open

Research Article

How do requirements evolve over time? A case study investigating the role of context and experiences in the evolution of enterprise software requirements

Stephan Schneider[1], Jan Wollersheim[2], Helmut Krcmar[3], Ali Sunyaev[4]

[1]University of Cologne, Cologne, Germany;
[2]fortiss, Technische Universität München, Munich, Germany;
[3]Technische Universität München, Munich, Germany;
[4]Kassel University, Kurt-Schumacher-Str. 25, 34117 Kassel, Germany

Correspondence:
Ali Sunyaev, Kassel University, Kurt-Schumacher-Str. 25, 34117 Kassel, Germany.
Tel: +49 561 804-3450;
E-mail: sunyaev@uni-kassel.de

The original version of this article was revised: April 2017. The original was originally published electronically on the publisher's internet portal (currently SpringerLink) on June 28, 2016, without open access, but this has since then been amended.

Abstract
In recent years, organizations have increasingly sourced cloud-based enterprise software (ES). Although comprehensively capturing organizations' requirements considerably affects the success of an ES sourcing project, little is known about how requirements evolve beyond the implementation. We conduct a longitudinal, exploratory single-case study of the life cycle of cloud-based ES in a medium-sized organization. Over 5 years, we trace the evolution of requirements throughout the ES life cycle, beginning with the initial adoption decision and ending with considerations to retire the ES. We develop a process theory that explains how requirements evolve beyond ES implementation and throughout its life cycle. We isolate nine mechanisms that explain how contextual factors and experiences are intertwined and shape the evolution of requirements. We then develop seven propositions that explain how sourcing cloud-based ES alters the mechanisms that shape the evolution of requirements. Our findings emphasize that the evolution of requirements for cloud-based ES follows similar mechanisms to that of the requirements for on-premises ES but changes how particular mechanisms manifest. Sourcing cloud-based ES changes the influence of business divisions in acquisition and configuration activities, the role of upgrade and customization procedures, and the influence of the ES' ecosystem.

Journal of Information Technology (2018) **33**, 151–170. https://doi.org/10.1057/s41265-016-0001-y; published online 28 June 2016

Keywords: enterprise software; requirements evolution; context; cloud computing; case study

The online version of this article is available Open Access

Introduction

Investigating requirements determination in organizations has a long tradition in information systems (IS) research (Byrd *et al.*, 1992). Requirements determination is a central activity when implementing or selecting enterprise software (ES), and failing to thoroughly capture requirements is a key reason for project failure (Hofmann and

Lehner, 2001). Driven by the shift in organizational practices to increasingly source packaged ES instead of relying on custom-developed solutions (Melgarejo, 2012), the focus and scope of requirements determination activities for customer organizations have changed considerably (Mathiassen *et al.*, 2007). The relevant issues for IS research and practice have shifted from defining requirements for engineering individual systems and components toward the selection and adaptation of configurable ES that is developed, provisioned, and maintained by external service providers (Jarke *et al.*, 2011). Further, with cloud computing having outgrown its infancy (Sunyaev and Schneider, 2013), organizations increasingly access packaged ES from cloud service providers (i.e., cloud-based ES) instead of hosting the packaged ES on-premises (i.e., on-premises ES). However, sourcing cloud-based ES bears certain peculiarities, such as self-service acquisition and shifting task responsibilities for requirements determination, which require organizations to adjust their sourcing processes (Schneider and Sunyaev, 2016). Hence, it has particular relevance to understand the processes surrounding requirements determination for packaged ES and the implications arising from cloud computing as a sourcing option.

As organizations need to be flexible and constantly adapt to changing situations, requirements regarding their ES are equally dynamic (Holmström and Sawyer, 2011). Once implemented, users and organizations learn from experiences with the ES and adjust their requirements (Wagner and Newell, 2007), resulting in customizations and ongoing maintenance of the ES (Light, 2001). Furthermore, requirements are shaped by contextual factors, such as organizational and societal structures (Howcroft *et al.*, 2004). Changes in the context can trigger changes in an organization's ES requirements (McGee and Greer, 2012) and individuals' understandings of those requirements (Davidson, 2002). Hence, it is important to focus on the dynamics of ES requirements beyond the implementation stage by analyzing how different contextual factors and experiences with the ES intertwine and shape the evolution requirements (Jarke *et al.*, 2011). Although previous research acknowledges the volatility and evolution of requirements over time (Markus and Tanis, 2000), the evolution of requirements throughout the ES life cycle scarcely represents the unit of analysis (Esteves and Bohorquez, 2007; Eden *et al.*, 2014). Particularly, research provides little explanation of the mechanisms that shape the requirements evolution process beyond implementation of the ES (Pollock and Williams, 2007). Hence, understandings of the phenomenon can be greatly improved if we "follow the process of selection through to implementation and use" (Howcroft and Light, 2010: 143).

Therefore, our aim is to shed light on the process whereby requirements evolve after an ES is initially selected and answer the following research question: *What are the contextual factors that shape the evolution of requirements for ES throughout its life cycle and how do requirements evolve over time?* When answering our research question, we particularly emphasize the contextual factors specific to sourcing cloud-based ES (as opposed to on-premises ES) and discuss how sourcing a cloud-based ES affects requirements evolution.

This paper is organized as follows. First, we provide an overview of the relevant background literature. We then outline our research method and the case investigated. Subsequently, we present our findings and theory development and conclude with a discussion and implications.

Related work and background

ES is defined as application software that supports core business processes across departments and between organizations (Seddon *et al.*, 2010), for instance, software for enterprise resource planning or customer relationship management (CRM). If ES is purchased as a package (often referred to as commercial-off-the-shelf or packaged ES), existing research highlights specifics concerning adoption and ongoing maintenance activities (Light, 2001). Throughout adoption and acquisition, requirements serve as a guideline to identify the best-fitting package. However, as ES packages are primarily built to serve an anonymous market, only some of the initially specified requirements are fulfilled by any package selected (Light, 2005). Consequently, besides adapting traditional workflows of the procuring organization to better suit the packaged ES, the ES is also tailored. At least two organizations perform maintenance activities on ES packages: the ES provider to advance its software and each company using the ES to manage its individual tailoring. The trigger for companies using packaged ES to engage in maintenance activities is the inherent dependence on the package vendor for product evolution, which in turn induces maintenance activities for each upgrade, for example, to test and reconfigure upgraded software packages and their inherent workflows and eventually to upgrade any tailoring (Lucas *et al.*, 1988). Consequently, the software-using company needs to cope with common software maintenance problems such as a lack of knowledge and participation on the part of users, a lack of programmer time, or poor-quality code modifications (Flynn, 1998).

By ES sourcing, we refer to all organizations' activities throughout the life cycle of obtaining an ES, beginning with the initial adoption decision, to acquisition, implementation, use, maintenance, and evolution and ending with considerations to retire the ES, which might initiate a new life cycle for another ES (Esteves and Bohorquez, 2007). To secure decent business support, decisions are required along the life cycle to balance considerations of evolving business needs, advancing features of ES packages offered, and resource needs associated with the alternatives (Khoo and Robey, 2007). Hence, requirements evolve accordingly throughout the life cycle.

Requirements determination throughout the life cycle of enterprise software

Requirements determination is an iterative process and organizations' requirements evolve based on their current set of requirements, contextual factors, and the methods used to elicit requirements (Hickey and Davis, 2004). Existing research provides myriad analytical frameworks, models, and techniques to gather and manage requirements (Byrd *et al.*, 1992; Mathiassen *et al.*, 2007). Furthermore, scholars address the complexity of requirements determination and investigate the social and political interactions of stakeholders that shape the process (Davidson, 2002; Holmström and Sawyer, 2011). The involvement of multiple stakeholders such as developers, consultants, and users to understand and specify requirements is crucial in the domain of acquiring software packages (Howcroft and Light, 2002). Users involved in requirements-determination activities for ES include managers or procurement specialists, as specific skills are required to successfully communicate requirements to vendors (Cavaye, 1995), particularly because packaged ES is product-focused rather than concerned with the actual needs of users (Howcroft and Light, 2002). However, research often identifies a sort of pseudo-participation, whereby users were involved in ES-sourcing projects but did not actually influence the decision (Howcroft and Light, 2006) or inform the requirements (Kawalek and Wood-Harper, 2002). Nonetheless, research has also shown that users have a strong influence after the implementation of an ES, that is, on the evolution of requirements based on actual usage behavior and adapted work routines (Wagner and Newell, 2007).

The evaluation of ES according to an organization's requirements is, along with requirements determination, a generally continuous job that spans all phases of the ES life cycle (Kaasbøll, 1997). Reassessments are required periodically, for example because of upgrades to the ES package in use, new ES on the market, or shifting requirements (Holmström and Sawyer, 2011). The latter are an integral part of any organization, as users learning to use the ES results in evolving requirements (Wagner and Newell, 2007). Furthermore, organizational actors adjust their work routines, which leads to organizational transformation over time and accordingly evolving ES requirements (Kawalek and Leonard, 1996; Orlikowski, 1996). Additionally, resistance to the use of implemented software may arise (Griffiths and Light, 2009). Hence, requirements evolve for various reasons, such as corrective (fix defects), adaptive (enhancements), and perfective (improving software quality) maintenance needs (Arthur, 1988).

The environment and context of the ES are crucial for understanding the process of requirements evolution. For instance, changes in regulations, organizational structures, personal needs, and the products and services on offer lead to evolving requirements (Cavaye, 1995). Substantive evidence from the IS and management literature suggests that organizational processes need to be investigated with respect to the context in which the phenomenon under consideration is embedded to fully grasp its complexity and dynamics (Howcroft *et al.*, 2004; Shepherd and Rudd, 2014). Contextual factors at various levels influence processes throughout the ES life cycle, including the acquisition process (e.g., manager's personality traits (Benlian and Hess, 2010)), the requirements determination process (e.g., collaboration between groups (Chakraborty *et al.*, 2010)), or the retirement process (e.g., technological integration of the ES (Walther *et al.*, 2013)). Common frameworks applied by IS researchers to structure contextual factors include the model of organizational buying behavior (Webster and Wind, 1972) and the technology–organization–environment (TOE) framework (Tornatzky and Fleischer, 1990). Webster and Wind (1972) describe four categories of contextual factors that influence the outcome of organizational buying decisions: environmental factors such as the business ecosystem surrounding the procuring organization; organizational factors such as company-wide policies; interpersonal social factors such as the interaction among employees; and individual factors such as the knowledge and skills of stakeholders involved in the decision. By dividing the environmental factors into the categories of technology and environment, the TOE framework explicitly considers the technological context. Similarly, Shepherd and Rudd (2014) specifically address aspects of the decision-specific domain (e.g., matter, motive, importance) as contextual factors influencing decision-making processes. This explicit accounting for the technological context is in line with the suggestion of Orlikowski and Iacono (2001) and supports our objective of a particular focus on the characteristics cloud-based ES in our theorizing. Therefore, the following section outlines the specifics of cloud-based ES sourcing, which we use as analytical devices in our theorizing on the evolution of requirements.

The context of sourcing cloud-based enterprise software

Cloud services are accessed over the Internet from a shared pool of managed and scalable hard- and software resources on a pay-per-use basis. Customers do not own, manage, or operate the underlying infrastructure, platform, or application capabilities but instead access resources that are remotely controlled by the provider through the Internet (Mell and Grance, 2011).

We distinguish two types of packaged ES: cloud-based ES, which is rented from external cloud service providers and provisioned over the Internet, and on-premise ES, which is acquired as a usage license from an external vendor but hosted under the client's control, for instance, in its own or an outsourced data center (Sawyer, 2000). While there are many similarities in the processes surrounding the life cycle of and requirements for on-premises ES and cloud-based ES, below, we summarize the peculiarities of sourcing cloud-based ES (see Table 1).

Theoretical pre-understanding of the evolution of requirements throughout the life cycle of enterprise software

Our theoretical pre-understanding is aligned with the three main components of a process theory (Van de Ven and Huber, 1990): the process itself (ES life cycle), antecedents (contextual factors), and outcomes (requirements). Regarding the process, we conceptualize the ES life cycle, which depicts the current state of the ES and the related events and activities from adoption until retirement of the ES.[1] Regarding the antecedents, we conceptualize contextual factors as antecedent conditions that initiate the process and shape its evolution (McGee and Greer, 2012) As outlined above, we adopt the common classifications of environmental, organizational, group, and individual factors (Webster and Wind, 1972). To explicitly account for contextual factors closely associated with sourcing cloud-based ES (see Table 1), we complement the classification with the technology and decision-specific domain. Regarding the outcomes, we conceptualize the ES requirements as outcomes that are influenced by the process (Hickey and Davis, 2004). However, we emphasize that ES requirements are constantly evolving throughout the ES life cycle and should not be regarded as a static end product of this process (Wagner and Newell, 2007; Holmström and Sawyer, 2011), particularly because the requirements per se shape the ES life cycle (Light, 2001). However, in terms of process theory in this research context, the dependent variables shaped by the process under consideration are the ES requirements.

Table 1 Peculiarities of sourcing cloud-based ES

Characteristic	*Description*	*References*
Self-service acquisition	Cloud-based ES is evaluated with limited provider interaction in a self-service manner. This enables adopters to use and test an ES online before actually buying it and shifts task responsibilities to verify the fit of requirements from providers to customers. Hence, instead of writing requests for proposals and tasking providers with evaluating the fulfillment of requirements, customers evaluate the fulfillment of requirements themselves.	(Pollock and Williams, 2007; Susarla *et al.*, 2010)
Limited customization	Cloud-based ES is provided in a multi-tenant manner on shared resources with a common code stack. Thus, customers are unable to access or modify the source code. Therefore, customizations are limited to developing plugins or integrating external applications via provided interfaces.	(Brehm *et al.*, 2001; Xin and Levina, 2008)
Limited upgrade control	Cloud service providers control the scope and timing of service upgrades. Given the multi-tenant architecture and the common code stack, customers cannot control if and when to implement upgrades. Thus, customers are limited in their ability to postpone or skip upgrades.	(Khoo and Robey, 2007; Marston *et al.*, 2011)
UI-based configuration	Cloud-based ES is a highly standardized service that provides versatile and profound configuration possibilities via the user interface (UI). Particularly, as a browser-based UI is often the only way to access the service, an increasing number of parameters that have to be customized for on-premise ES are shifted to the UI for cloud-based ES, including workflow changes and plugin installations. Thus, customers are able to apply modifications themselves to a greater extent than in on-premises scenarios.	(Light and Wagner, 2006; Jutras, 2015)
Extended service ecosystem	Cloud services are based on open web standards and built with high integration capabilities, enabling customers to choose from a broad range of interfaces to integrate the service into their existing IT landscape. Additionally, cloud-based ES providers offer platform and marketplace services to encourage third-party vendors to develop custom-built plugins closely integrated with their core service (e.g., Force.com for Salesforce). Hence, customers have access to an open ecosystem of plugins and extensions to customize their ES.	(Light and Wagner, 2006; Beimborn *et al.*, 2011)
Outsourced maintenance	Cloud-based ES is rented from external providers that are responsible for the operation and maintenance of their services. Thus, customers do not need to account for the infrastructure and human resources required to install and maintain the ES, thereby enabling faster service setups with low upfront capital investments while also increasing the dependency on the provider to ensure reliable service operation.	(Cusumano, 2007; Lins *et al.*, 2016)
Pay-per-use pricing	Cloud services are designed for scalability and priced on a pay-per-use-basis, which enables customers to scale resources in line with demand and pay based on actual use. Hence, customers can cope with new requirements by activating additional modules, integrating new plugins, or scaling up the number of users on-demand without provider interaction or contractual amendments.	(Choudhary, 2007; Armbrust *et al.*, 2010)

Figure 1 represents our theoretical pre-understanding of the evolution of requirements throughout the ES life cycle and acknowledges that the ES life cycle consists of multiple iterating and not necessarily sequential activities from adoption until retirement (Verville and Halingten, 2003). The requirements evolve throughout the ES life cycle (e.g., use of the ES reveals new needs for additional functionalities). In turn, the ES life cycle is shaped by evolving requirements (e.g., changed requirements may facilitate modifications to the ES). Both the life cycle and the evolution of requirements are shaped by contextual factors at the decision-specific, technological, environmental, organizational, group, and individual levels.

Following Sarker *et al.* (2012), the purpose of this research is discovery and not to deductively test the theoretical framework depicted in Figure 1. Hence, Figure 1 represents a snapshot of our theoretical sensitivity and provides a legitimate frame and vocabulary to guide our theory-building purpose.

Research method

We apply a longitudinal, exploratory single-case study approach in a medium-sized private-sector organization (pseudonym: Alpha) and follow established recommendations for positivist case-study research (Dubé and Paré, 2003; Yin, 2009). Data collection and data analysis overlapped iteratively.

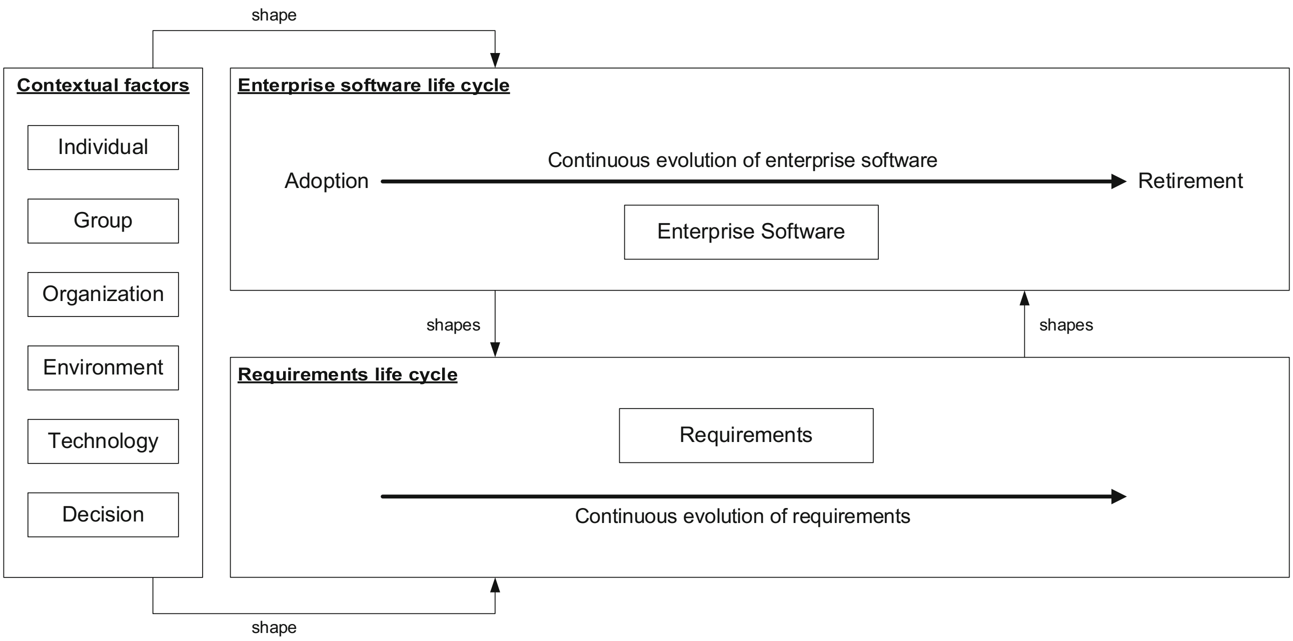


Figure 1 Theoretical pre-understanding of requirements evolution throughout the ES life cycle.

Table 2 Overview of interviewees

Reference	*Position*	*Company*	*Formal interviews*	*Informal inquiries*
[COO]	Chief Operations Officer	OM	2	3
[CM]	CRM Manager	Alpha	4	5
[CS-1]	Head of CS	OM	1	2
[SA-1]	Sales Account Manager	OM	2	1
[SA-2]	Sales Account Manager	OM	2	1
[CTO]	Director Technology	Alpha	1	0
[CON]	External CRM Consultant	Alpha	1	2
[GC]	General Legal Counsel	Alpha	1	1
[CS-2]	Head of CS	IM	1	0
[IT-1]	Head of IT Division	OM	1	0
[IT-2]	Software Developer	OM	1	0
Total number of interviews			**17**	**15**

Note: The references in square brackets in the reference column are used in the text to refer to the interviewees. OM and IM are subsidiaries of Alpha; see Case description section for details.

Empirical data were collected through two rounds of semi-structured interviews (in 2013 and 2014), documents (internal slide decks, spreadsheets, public documents), and informal ad hoc inquiries (phone calls, emails, informal on-site meetings, and instant messages).

The purpose of the first round of interviews was to obtain a holistic understanding of Alpha's sourcing activities. We asked open questions regarding the course of the sourcing process and contextual factors influencing the process. We completed semi-structured interviews with 11 employees (see Table 2) that lasted between 30 and 130 min each. These 11 employees represent the core sourcing team who contributed the majority of expertise and time to the project. To eliminate misunderstandings in our findings, we crafted a brief summary of each interview with reflective remarks from the researchers. We then returned the transcripts and summaries to the interviewees and asked for their comments.

To obtain a holistic picture, we gathered multiple documents from Alpha. Some of the documents are supportive materials that provided additional insights to confirm dates and facts (e.g., emails, business social network profiles of employees, and annual reports). Other documents provided in-depth insights into Alpha's sourcing activities. Internal slide decks and spreadsheets contain detailed information on Alpha's situation in 2007, requirements, and decision criteria. Furthermore, documents provided insights regarding the stepwise evolution of the ES at Alpha (subsequent rollouts and integrations). One document is the decision draft used in 2012 to decide

Table 3 List of requirements OM used during the re-evaluation project in 2012

Requirement	*Description*
Upgrade	Amount of effort and risk associated with each service upgrade (e.g., to check configurations and customizations, quality assurance, update workstations, perform compatibility checks with on-premise systems such as the operating systems or office suites)
Configuration	Degree and ease of configurability via the user interface (without writing code)
Customization	Degree and ease of customizability (extensions by writing code or integrating third-party plugins); Ecosystem, i.e., availability of integrators to customize or ready-to-use third-party plugins (e.g., via marketplace)
Portal/FAQ	Features of FAQ integration and customer portal (e.g., analysis, tracking, reporting, ranking and sorting of FAQs, ease of FAQ administration)
UX	Usability, response times and availability, user interface, and end-user experience
Mobile	Availability of mobile app for tablets or smartphones
Outlook integration	Simplicity/possibility of workflow integration with Microsoft Outlook
IT effort	Effort required of the IT division to maintain all devices with which the service is used (e.g., compatibility checks before each update of the operating system, office suite, or web browser)
Customer support	Quality and price of enterprise-level customer support
API	Scope of functionality, limits of API usage, and respective costs
License cost	Total cost to rent the required capabilities for all divisions
Effort to switch	Effort to retire RightNow and set up Salesforce (does not include migration costs): manpower and time needed to extract and transform data, reconfigure Salesforce, train users, etc
Effort for admin	Effort for ongoing administration and evolution of the service; this does not include effort for the IT division but for the business divisions, i.e., the CRM manager
Two CRM systems	Additional effort (not license cost) to maintain and configure two CRM systems
Training/jobs	Availability of training material and qualified/certified administrators, scope and quality of the support community

Note: The 15 requirements stem from the provided documents and were inductively derived from the data. Definitions are based on provided documents and explanatory comments of [COO] and [CM].

whether to switch providers or to continue employing the cloud-based ES in use (RightNow). This document provided key facts concerning the re-evaluation project comparing two alternative CRM systems (Salesforce and RightNow) according to 15 requirements and elaborating the key strengths and weaknesses of each service on multiple slides. The gathered documents provided information beyond the scope of our interviews and complemented our empirical data with a rich level of detail.

Based on the first round of data collection, we identified the lessons learned about the sourcing process and crafted a consolidated case report for Alpha. Although our research was guided by established guidelines, we maintained a flexible approach rather than using the guidelines as a fixed blueprint (Keutel *et al.*, 2014). Hence, after the first round of data collection and interpretation, we regarded the re-evaluation project in 2012 as particularly puzzling.

Given the rich documentation of all sourcing milestones, we were able to trace requirements from the initial sourcing decision in 2007 until the re-evaluation decision in 2012. Some requirements already existed in 2007 (e.g., Customer Portal/FAQ), while some developed during usage (e.g., Customer Support). Hence, the purpose of the second round of interviews was to gain an in-depth understanding of the re-evaluation project in 2012, particularly to discuss each of the 15 requirements in the decision draft (see Table 3) and to gain insights into the evaluation of the CRM systems of Salesforce and RightNow.

Therefore, we conducted formal follow-up interviews in 2014 with the re-evaluation team [COO, CM, SA-1, SA-2] and informal follow-up interviews with [CS-1, CON], as the latter were only marginally involved as advisors. Interviews lasted between 20 and 64 min. We discussed each of the 15 requirements in detail to clarify their definition and scope and to distinguish why each requirement appeared on the final list. We also asked questions to understand how each of the alternatives was evaluated and assessed according to the requirements.

Data coding and analysis was conducted independently by two researchers and jointly in a team (using NVivo 10 and spreadsheet software) and followed an iterative approach aligned with Van de Ven and Poole (1990). Appendix A provides a visualization and examples of our coding and analysis process. We first analyzed the transcripts and documents from both data-collection periods and inductively applied descriptive codes (open coding) to identify incidents. Incidents are the focal objects of coding, in our case any contextual factors, events, facts, or actions taken that might have affected the course of the ES life cycle or impacted the ES requirements. We then identified linkages between incidents (axial coding) by analyzing each incident and determining its antecedent incidents (i.e., incidents that caused or facilitated the incident) and its descendant incidents (i.e., incidents that were caused or facilitated by the incident). We next discussed and grouped sets of incidents, for instance, multiple incidents facilitating the same incident or being the effect of the same incident. The result was a collection of jointly established write-ups, data display tables, and visual displays that summarize linkages between incidents and explain how the contextual factors of Alpha's sourcing context and the incidents that occurred along the observed life cycle of RightNow ultimately shaped the requirements for the re-evaluation in 2012. To derive the theory for the evolution of requirements in ES sourcing contexts, we systematically analyzed the patterns underlying our data and synthesized the mechanisms that shaped the evolution of requirements (selective coding). In this step, we iteratively identified patterns in the data by merging groups of incidents that had similar effects (e.g., affected similar requirements). This step included multiple meetings for crafting data and visual displays (manually drawn on whiteboards and flip charts), presentations at academic workshops, and iterative refinement of our results.

Case description

Alpha is a publicly traded, multinational, medium-sized organization headquartered in Germany with approximately 140 million euros in annual revenue and 330 employees. Alpha owns two subsidiary companies, Online Marketplace (OM) and Internet Marketing (IM). Both subsidiaries are headquartered in Germany, have international offices, and serve a global market of business and private customers. OM and IM operate separately and have their own organizational structures and management boards. However, the subsidiaries share resources, for instance, employees situated within the organizational structure of Alpha and serving both subsidiaries, such as the legal counsel. OM provides an online marketplace for over two million customers to sell, buy, and exchange virtual goods. IM provides Internet marketing and advertising solutions to over 500,000 customers.

The sourcing process we analyze spans from 2007, when RightNow was introduced in OM's customer service (CS) division, until 2012, when OM considered retiring the implemented service and switching to another service provider. The following three milestones summarize Alpha's sourcing activities.

Initial sourcing

In 2007, OM realized a significant increase in revenue. Requests to be handled by the CS division also increased considerably. However, the CS division at OM handled customer requests with regular on-premises email software. Requests were manually filtered and distributed to agents. The CS division could not handle the increased workload with its manual processes and therefore urgently needed a solution to support its activities. After evaluating multiple alternatives, OM ultimately decided to implement the CRM service RightNow for its CS division. The service went live in 2008.

Operation and subsequent rollouts

Over the years of operation, RightNow was rolled out in more divisions, and the functional scope was extended by acquiring additional modules, integrating external applications, and utilizing custom-developed plugins. For instance, in 2009, IM demanded CRM functionality for the sales and CS divisions. IM began a comprehensive evaluation process from scratch. The final decision resulted in the acquisition of RightNow for both divisions, independent of the fact that OM also used RightNow. The service went live at IM in 2011. Furthermore, in 2010, OM's management strove for greater transparency in the sales division and decided to also roll out RightNow. RightNow provides sales modules, and OM did not want to utilize two separate services. Thus, alternatives were not considered. In 2011, RightNow went live for OM's sales division.

Re-evaluation

By 2012, RightNow was used extensively across multiple divisions of OM and IM and thoroughly integrated within Alpha's IT landscape. However, RightNow was never fully accepted by OM's sales division, where resistance increased over time. An upgrade in 2012 led both divisions of OM to suffer significant downtime, incompatible configurations, changed interfaces, and broken plugins, which ultimately resulted in OM initiating a re-evaluation process. OM invited providers to present their CRM cloud services in on-site workshops and shortlisted RightNow and Salesforce. The CRM manager [CM] crafted an initial draft of 15 requirements and evaluated both services on each requirement. Next, [CM] presented his evaluation to the Chief Operations Officer [COO], and they discussed each requirement and the according evaluations of both services in fulfilling the requirements. [COO] added his input from a management perspective, which resulted in adjusting the requirements and the evaluation scores of each system on particular requirements. [COO] presented the decision draft in a meeting of the executive board of Alpha and suggested retaining RightNow. Ultimately, OM renewed its contract with RightNow for another year but stated its intention to discontinue the contract if the next upgrade caused complications. However, the next upgrade was smoothly implemented.

Case analysis

Our unit of analysis is the *process whereby OM's requirements evolved from its initial sourcing decision in 2007 until the re-evaluation in 2012*. Hence, the focal part of this analysis is to understand how the context and the events encountered along the observed lifecycle of RightNow shaped the requirements for re-evaluating services in 2012.

Initial sourcing

The inefficient processes of OM's CS division resulted in long response times, no consistent customer history, unsatisfied customers, and non-measurable employee workloads. Hence, the CS division at OM urgently needed a solution to support its activities. CS employees evaluated and tested three preselected ticketing systems in detail, but none met its expectations. While evaluating the ticketing systems, requirements evolved beyond pure ticketing procedures; OM refocused its efforts on finding a CRM service, which ultimately resulted in selecting RightNow for six reasons: (1) reduced email traffic, (2) increased customer satisfaction (professionalism, reduced response time), (3) hosted service to avoid administrative IT effort,

(4) continuous advancement of the service, (5) modular design for CRM (integration into existing processes and back-office systems), and (6) cost control and efficacy measurement.

The CS division decided to roll out RightNow autonomously. A major driver of the single-handed rollout was the urgent need to resolve process inefficiencies. Consequently, the CS division focused on its distinct business goals such as increasing the workflow efficiency of CS employees. This urgency, paired with a constant lack of resources in OM's IT division, resulted in excluding the IT division. There were no internal policies that required consulting the IT division before purchasing such software. Employees within the CS division and the COO have a strong IT background and therefore executed the project autonomously. Additionally, the high degree of configurability of the UI drove the CS division's perception that it could configure the service itself.

After initial attempts to implement this highly complex service, the CS division realized that it lacked skills and knowledge to configure the service to meet its needs. Hence, OM hired a consultancy for the initial setup in 2008 and to provide training. Nonetheless, a considerable amount of work remained for the CS division, and it required extensive support from RightNow to complete the setup. The entire configuration process required over 6 months.

Furthermore, OM signed the contract without including a premium support level. The CS division underestimated its support needs and did not thoroughly grasp the pricing of support services. The basic support of RightNow was limited and allowed only twelve requests per year; each additional request would incur further charges. Due to its unwillingness to pay the additional cost for the premium support plan, OM experienced very limited support with long response times of up to several months. First, the employees responsible for configuring the service relied on the support community and self-service support structures, such as user forums and FAQ pages. However, some issues had to be resolved by RightNow, and given the lack of the premium support plan, response times were unacceptable for operations; hence, OM was required to increase the support level, resulting in an unexpected additional cost of 18 % of the contract volume.

As a mid-sized organization, OM acquired RightNow with limited provider involvement and evaluated the fit of the service with its requirements. Experiencing this self-service principle of cloud computing for the first time for a rather complex ES, OM faced the challenge of unfulfilled requirements. OM did not recognize all limitations of the acquired service; for instance, it failed to verify that the service was compatible with the desired email workflow. When it attempted to configure the workflow, OM recognized that the workflow was not configurable to the company's needs, and OM had to maintain the predefined process.

Table 4 summarizes the contextual factors, their influences on the ES life cycle, and the experiences gained by OM as outlined in the preceding narrative.

Based on the events during the initial sourcing milestone, we identified two cloud-specific contextual factors (UI-based configuration and self-service acquisition) that facilitated two relevant experiences at OM. First, OM underestimated the effort needed to configure the service in terms of time, complexity, and required support. Second, OM's requirements evaluation lacked rigor in terms of ensuring the fulfillment of specific requirements. These two experiences in turn can be associated with the evolution of five requirements for re-evaluating services in 2012, as summarized in Table 5.

Operation and subsequent rollouts

Due to the modularity of RightNow, the service has been integrated in several other divisions at OM and IM. However, the business processes in OM's sales division were rather unstructured and intuitive. OM attempted to configure the service to best support the existing sales workflows, which were based on the use of email, spreadsheets, and back-office tools. Instead of following the predefined workflows in RightNow, OM made extensive use of the UI-based configuration options and adapted the system instead of convincing sales agents to adapt their workflows. An absence of process definitions and a lack of clear responsibilities for analyzing and defining requirements meant that certain requirements of individual employees became constituent parts of the configuration.

Table 4 Contextual factors and their influence on the ES life cycle during initial sourcing

Contextual factors	*Influence on the ES life cycle*
IT affinity Distinct business goals IT division capacities IT-procurement policies UI-based configuration Complexity	**Event**: Single-handed rollout of RightNow by the CS division (no cross-functional alignment or consulting IT expertise) **Result**: Configuration was problematic and required considerable time **Experience**: Underestimated configuration effort (time, complexity)
IT affinity UI-based configuration Cost pressure	**Event**: Signing a contract without premium support level **Result**: Poor customer support during configuration and rollout, unexpected additional cost for support **Experience**: Underestimated configuration effort (required support)
Self-service acquisition Complexity	**Event**: Requirements evaluation conducted by CS division **Result**: Unfulfilled requirements **Experience**: Lack of rigor in self-service requirement evaluation

Table 5 Requirements evolution based on experiences during initial sourcing

Requirement	*Evolution*
Configuration	Due to the negative experiences regarding the complexity and time required to configure the service, the ease of configurability became a very important consideration. Configurability was also a requirement during initial adoption; however, it was instead checked if specific workflows and requirements could be configured at all. Hence, the scope of this requirement was adjusted (to also cover the ease of configuration).
License cost	Due to the underestimated support needs and the resulting additional cost for premium support, this requirement was adjusted in scope to also cover the available support plans and associated cost.
Customer support	Due to the poor support by RightNow, the importance of reliable customer support increased and the emphasis was on rigorously and comprehensively assessing customer support procedures (how and by which terms are support requests handled and what types of support plans are available).
Training/jobs	Due to the poor support by RightNow and the complexity involved in configuring the service, the configurators identified ways to help themselves and used self-support (e.g., support forums, tutorials). However, they experienced that such channels offering help were scarce at that point in time. Hence, the importance of a viable support community was a new requirement during re-evaluation (subsumed in training/jobs).
Outlook integration	OM assumed that RightNow supported its established Outlook integration workflow but was disabused when it wanted to configure the workflow to its needs. Hence, rigorously and comprehensively assessing the possibilities and workflows of Outlook integration was emphasized.

Due to a lack of IT expertise in the sourcing team, tremendous effort was made to configure the service, which strained RightNow to the limits of its configurability. The modifications of RightNow became very complex and over-specific, resulting in a slow service, complex workflows, and poor usability. Low user acceptance and discontent in the sales division plagued the outcome.

To address more sophisticated demands that evolved over time (e.g., voice over IP integration), OM implemented customizations with tailor-made plugins because source code modifications were not possible and RightNow lacked a marketplace to acquire ready-made plugins. Furthermore, because RightNow offered a broad scope of modular features that was useful for a broad variety of application contexts (e.g., sophisticated reporting), the level of integration with other specialized back-office systems increased. RightNow became a central system for OM and crucial for daily operations. Workflows in many divisions were handled in RightNow, and other systems in OM's IT landscape made extensive use of RightNow's open interfaces to synchronize and exchange data.

This high degree of customization and integration was not considered problematic until RightNow rolled out upgrades and depreciated features (particularly when one API was depreciated that was utilized by many other systems). The continuous evolution of the service was one of the favorable aspects during the initial selection in 2007 but subsequently proved problematic. RightNow controls the scope of released or depreciated features and the schedule of upgrades. Alpha is responsible for maintaining customizations and the interface using plugins. Hence, the high level of customization and integration within Alpha's IT landscape resulted in multiple upgrades failing and requiring tremendous effort to restore configurations, customizations, and plugins. This resulted in high ongoing maintenance effort for [CM] to assess the compatibilities of configurations and customizations for each provider-induced upgrade. Significant additional costs accumulated to keep configurations and interfaces consistent with the evolving service core and its interfaces.

Furthermore, problems with the local workstations and network infrastructure occurred. The IT division's effort to perform compatibility checks before each update of on-premise systems (e.g., the operating system, office suite, or web browser) increased significantly with the customized service. For instance, a .NET update on workstations resulted in RightNow crashing on startup. However, the plugins, and not RightNow, were responsible for such crashing. Hence, considerable additional effort was necessary because specific requirements for the workstations had to be considered before installing workstation updates.

As RightNow became critical for daily operations, the ongoing effort to configure and maintain the service increased. Alpha had to establish a dedicated CRM manager, who administered the service for both subsidiaries and was accountable for new feature rollouts, configuration, upgrade planning, and training key users. However, when attempting to hire adequate personnel, OM experienced the consequences of a job market that lacked candidates with the required skillsets. Moreover, to account for the increased effort to configure and maintain the service, OM established a network of 'power users.' IT-affine staff members from business divisions were trained to configure dedicated business workflows, reducing the CRM managers' workload. However, training material was barely available for these power users.

Table 6 summarizes the contextual factors, their influences on the ES life cycle, and the experiences gained by OM as outlined in the preceding narrative.

Based on the events during service operation and subsequent rollouts, we identified four cloud-specific contextual factors that facilitated relevant experiences at OM: UI-based configuration, limited customization, and extended

Table 6 Contextual factors and their influence on the ES life cycle during operation and subsequent rollouts

Contextual factors	*Influence on the ES life cycle*
Modularity of existing ES	**Event**: Forced rollout in the sales division **Result**: Discontent in the sales division **Experience**: Lack of end-user involvement
Heterogeneous requirements IT expertise of sourcing team Unstructured business processes UI-based configuration	**Event**: Excessive configuration (UI-based) **Result**: Complex workflows, performance issues, poor usability **Experience**: Over-configuration remote from service core
Availability of plugins Limited customization Sophisticated demands	**Event**: Excessive customization (plugin development) **Result**: High maintenance effort (need to check modifications for each update) and high IT effort (need to check workstations for each update) **Experience**: Over-customization remote from service core
Limited upgrade control Open interfaces	**Event**: Upgrades rolled out on highly customized and integrated service **Result**: Failed upgrades **Experience**: Importance of the service provider's upgrade and maintenance policies
Available community and consulting Job market and trainings Criticality of ES for daily operations	**Event**: Establish a dedicated CRM manager **Result**: Scarcity of skilled personal in the job market and lack of training material **Experience**: Significance of skilled configurators

Table 7 Requirements evolution based on experiences during operation and subsequent rollouts

Requirement	*Evolution*
Mobile	Given the discontent in the sales division and the growing resistance to RightNow, providing mobile access to the service emerged as new requirement, which was fulfilled by Salesforce but not by RightNow. However, this requirement was assessed as result of resistance to RightNow rather than the actual need for mobile access
Configuration	Due to the high degree of configuration conducted by business divisions, OM realized the degree of configuration required to depict its workflows. In combination with the failed upgrades due to customizations, a high degree of configurability gained importance as a requirement. Furthermore, with the introduction of power users from business divisions executing configurations, the importance of the ease of configurability also increased
Effort for admin	Although RightNow is hosted and maintained by an external provider, with the extended use of RightNow in multiple divisions, [CM] had to cope with a vast amount of ongoing effort (e.g., ongoing configuration, managing customizations). Particularly, due the failed upgrades that required tremendous effort to restore configurations, customizations, and plugins, the effort for the CRM manager evolved as an important new requirement
UX	Due to the increased prevalence of RightNow throughout the company and the problems with poor usability, the need for a satisfying user experience gained importance
Customization	Due to the lack of ready-made plugins (e.g., via a marketplace) and the problems with custom-developed plugins during upgrades, the ease and degree of customization gained importance, and the scope of this requirement shifted to include the ecosystem of the service (e.g., marketplace, integration partners)
Upgrade	Failed upgrades were one of the triggers of the re-evaluation project and resulted in the provider's upgrade and change management procedures becoming the most important requirement for service re-evaluation
IT effort	Due to the increased effort required by the IT division to maintain the workstation infrastructure in OM's offices to constantly assess compatibility with customizations and new service versions, internal IT effort associated with the service evolved as new requirement
Training/jobs	Due to the difficulty in finding a skilled configurator in the job market, training and the job market evolved as new requirement

ecosystem are factors that shape how modifications are conducted in cloud-based ES. These three factors in combination with the fourth factor of limited upgrade control led to OM's experiences related to problems with the upgrade process of highly customized and integrated services. OM's relevant experiences during service operation and subsequent rollouts were the lack of end-user involvement during the sales rollout, the over-configuration and over-customization

Table 8 Contextual factors and their influence on the ES life cycle during re-evaluation

Contextual factors	*Influence on the ES life cycle*
Accumulated knowledge Distinct business goals Level of integration Established relationships Criticality of ES for daily operations	**Event**: Considerations to switch providers **Result**: Considerable efforts to switch **Experience**: Bond with the service (established integrations, skills, and relationships)

Table 9 Requirements evolution based on experiences during re-evaluation

Requirement	*Evolution*
Effort to switch	Considerable configuration expertise has been accumulated within the organization and RightNow has been geared toward Alpha's needs. Considerable effort would be required to achieve the same level of integration and configuration with a new service. Hence, the effort to switch evolved as a new requirement.
API	The high degree of integration with other back-office systems and special-purpose systems that were developed to be integrated into RightNow workflows (e.g., to handle special-purpose data processing) increased the importance of a flexible and wide-ranging API.
Two CRM systems	Two CRM systems evolved as a new requirement because IM was very pleased with RightNow and was not willing to switch. Thus, switching OM would still require the maintenance of two CRM systems by [CM].

of RightNow remote from the service core, the importance of the cloud service provider's upgrade and maintenance policies, and the significance of skilled configurators. These experiences can, in turn, be associated with the evolution of eight requirements for re-evaluating services in 2012, as summarized in Table 7.

Re-evaluation

Alpha had a flexible contract that it could terminate on short notice. Furthermore, RightNow offered well-described interfaces that permitted complete data retrieval in the event of service termination. However, when re-evaluating RightNow and considering switching to an alternative solution, Alpha realized that it was highly committed to the service.

RightNow had become an integral component of Alpha's IT landscape, was used in multiple divisions for daily operations, and supported core business processes throughout the organization. The service had become highly integrated with other systems, meticulously configured and suited to the company's needs, and the necessary knowledge to configure and use the service accumulated within the organization. In addition to individual knowledge, Alpha became increasingly familiar with the service ecosystem, and relations were established with the support community, other customer organizations, and capable consultancy and integration partners. Additionally, IM was pleased with RightNow and did not consider switching, which would have required [CM] to manage two systems.

Table 8 summarizes the contextual factors, their influences on the ES life cycle, and the experiences gained by OM as outlined in the preceding narrative.

Based on the events during service re-evaluation, none of the contextual factors that facilitated relevant experiences at OM was cloud-specific; rather, they related to the ES being widely diffused in the organization. OM decided to stay with RightNow because of its bond with the service (considerable configuration expertise and knowledge had been accumulated; the high level of integration would require costly re-integrations with another service), and the advantages of Salesforce in some areas were offset by disadvantages in other areas (i.e., efforts of administration, portal/FAQ, and configuration). The bond with the service can be perceived in the evolution of three requirements for re-evaluating services in 2012, as summarized in Table 9.

Theory development

When analyzing the ES life cycle at OM with respect to our research question, we focus on two components of our process theory: the identification of contextual factors that shape the evolution of requirements and the identification of the mechanisms that explain how these contextual factors shape requirements.

First, the preceding section describes the events that occurred at Alpha and identifies the contextual factors that shaped the evolution of requirements throughout the ES life cycle of RightNow (the first part of our research question, summarized in Tables 4, 6, and 8). Factors in all contextual domains of our theoretical pre-understanding influenced the evolution of requirements at OM. These factors include the IT affinity members of the sourcing team (individual context), the distinct business requirements of the CS division (group context), the lack of adequate support communities and consulting services (environmental context), the lack of capacities in the internal IT division (organizational context), the high degree of UI-based configurability (technological context), and the asset to be sourced for complex business needs (decision context). Figure 2 summarizes the identified factors according to the contextual domains of our theoretical pre-understanding.

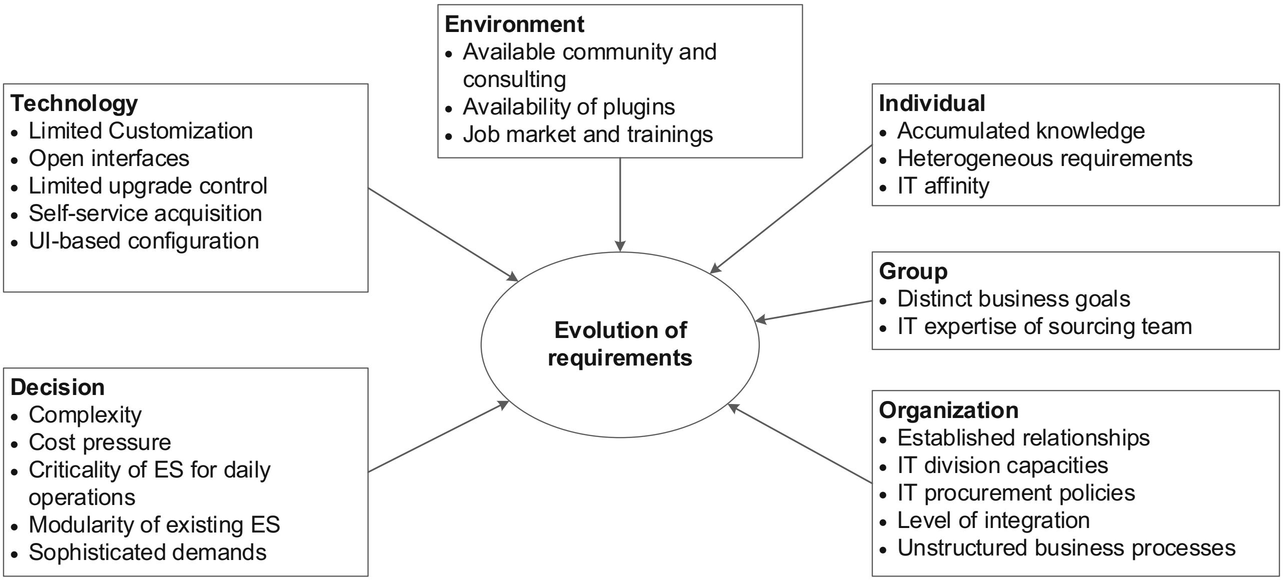

Environment
• Available community and consulting
• Availability of plugins
• Job market and trainings
Technology
• Limited Customization
• Open interfaces
• Limited upgrade control
• Self-service acquisition
• UI-based configuration
Individual
• Accumulated knowledge
• Heterogeneous requirements
• IT affinity
Group
• Distinct business goals
• IT expertise of sourcing team
Evolution of requirements
Decision
• Complexity
• Cost pressure
• Criticality of ES for daily operations
• Modularity of existing ES
• Sophisticated demands
Organization
• Established relationships
• IT division capacities
• IT procurement policies
• Level of integration
• Unstructured business processes

Figure 2 Contextual factors that shaped the evolution of requirements at OM.

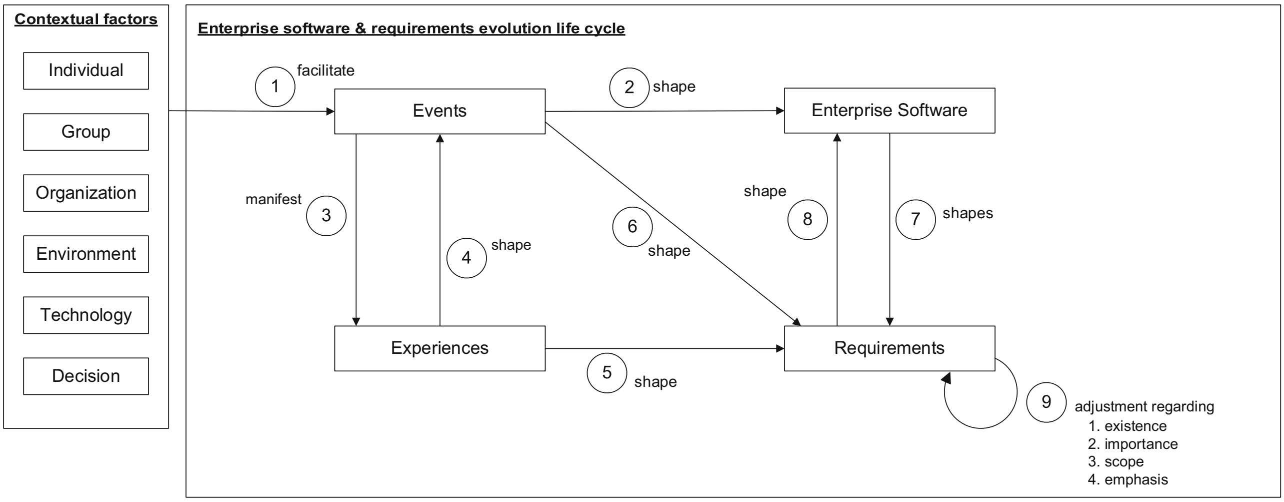


Figure 3 A model of the mechanisms shaping the evolution of requirements throughout the ES life cycle.

Second, the preceding section explains how OM's requirements evolved over time (the second part of our research question, summarized in Tables 5, 7, and 9). The contextual factors did not directly influence OM's requirements. Contextual factors facilitated particular events throughout the ES life cycle (e.g., the high IT affinity of members of the sourcing team facilitated the CS division's single-handed rollout of RightNow). Events refer to occurrences during the ES life cycle, for instance, activities conducted by stakeholders (the high degree of configuration of the service to meet OM's needs) or external triggers (the rollout of an upgrade by the provider). The events during the life cycle of the ES shaped the ES itself (e.g., the excessive configurations resulted in poor usability of the ES). Furthermore, managing the events throughout the ES life cycle enabled OM to learn particular lessons (experiences) that in turn affected the events during the re-evaluation project in 2012 (e.g., the experienced lack of rigor in self-service requirement evaluation changed how OM conducted requirement evaluation). The requirements employed in the re-evaluation project can be traced not only to events and experiences OM witnessed during the last years of operating RightNow but also to characteristics of the ES itself (e.g., the flexible capabilities of RightNow's interfaces facilitated the demand for further integrations). The evolution of requirements, in turn, shaped the ES (e.g., the sales agents' demands for custom processes resulted in customizations of RightNow). Requirements evolved based on different types of adjustments (e.g., adjusting the scope). Based on the preceding synthesis, Figure 3 and Table 10 depict the mechanisms that shape the evolution of requirements throughout the ES life cycle.

Throughout the preceding analysis, we emphasize the cloud-specific contextual factors that shaped the evolution of requirements at OM. Sourcing cloud-based ES and sourcing on-premises ES share common characteristics; however, sourcing cloud-based ES entails peculiarities (see Table 1). Hence, the subsequent discussion focuses

Table 10 Mechanisms shaping the evolution of requirements throughout the ES life cycle

ID	*Mechanism*	*Example at Alpha*
1	Contextual factors facilitate events	IT affinity of sourcing team members facilitates the single-handed rollout of RightNow by the CS division
2	Events shape ES	Excessive configurations result in complex workflows, performance issues, and poor usability of RightNow
3	Events manifest in experiences	The forced rollout of RightNow in the sales division and the resulting discontent among sales employees demonstrated the significance of end-user involvement for Alpha
4	Experiences shape events	The experienced lack of rigor in self-service requirement evaluation resulted in the emphasis on evaluating and testing specific requirements in online trial services (e.g., Outlook integration)
5	Experiences shape requirements	The experience of failed upgrades resulted in reliable upgrade and change management procedures of the provider emerging as the most important requirement
6	Events shape requirements	Involving sales agents in the re-evaluation process brought the requirement 'mobile' onto the requirements list
7	ES shapes requirements	The flexible capabilities of RightNow's interfaces facilitated the demand for further integrations (e.g., other back-office systems)
8	Requirements shape ES	The sales agents' demands for custom processes resulted in customizations to RightNow
9	Requirements are adjusted regarding 1. existence 2. importance 3. scope 4. emphasis	1. Added 'mobile' to requirements list 2. Increased importance and weighting of 'upgrade' 3. Adjusted 'customization' to additionally cover a marketplace for plugins 4. Emphasis on the rigorous and comprehensive assessment of 'customer support' procedures

Note: IDs in this table refer to the circled numbers in Figure 3.

on how the peculiarities of the cloud-sourcing context affect requirements and the process whereby requirements evolve.

Discussion

Flexible licensing options paired with UI-based configurability and self-service acquisition possibilities enable business divisions to acquire ES without involving IT expertise. OM's requirements in 2007 were predominantly business-driven, targeting the distinct business goals of the CS division. However, OM's requirements in 2012 were far more concerned with IT-specific requirements, as OM recognized the importance of IT-specific requirements over time (e.g., upgrades, customization, configuration, in-house IT and maintenance effort). Hence, we propose that the self-service acquisition of cloud-based ES affects organizations' ES requirements as follows:

> **Proposition 1:** When business divisions conduct the acquisition of cloud-based ES autonomously, IT-specific requirements are less important in the first place, but with extended use and tighter integration in an organization's IT landscape, IT-specific requirements gain importance.

Cloud-based ES is ready to use but needs to be adjusted to fit an organization's individual workflows and needs. UI-based configurability allows business divisions to conduct the configuration and parameterization of the ES themselves. However, while accessible via the UI, this configuration necessitates proper training and support. OM's CS division underestimated its support needs for service configuration in 2007 and experienced misconfigurations and severe delays in rollout. Learning from its experiences, requirements such as the ease of configuration, customer support, and the size of the support community were among the most important requirements in 2012. Therefore, we propose that the UI-based configurability of cloud-based ES affects ES requirements as follows:

> **Proposition 2:** When business divisions conduct the configuration of cloud-based ES themselves, the requirements of the ease of configuration and reliable customer support (training material, support community, and provider support) gain importance.

Cloud services promise up-to-date IT resources operated by the service provider, implying reduced maintenance effort for the customer, but on the downside reduced control over the service and data. However, due to limited control over service upgrades, maintenance effort increases with growing customization and integration, requiring the customer to ensure interface and plugin compatibility for each upgrade. Depending on the upgrade policy of the provider, this 'supposedly reduced maintenance effort' can considerably exceed expectations, as in the case of OM. In 2007, OM favored RightNow as cloud-based ES with regular updates and enhancements. However, based on the company's experiences with failed upgrades, in 2012, the continuous and ongoing upgrades became an inhibiting factor and solutions with less-conflicting upgrade procedures were favored. Therefore, we propose that the limited upgrade control of cloud-based ES affects ES requirements as follows:

Proposition 3: With extended customization and tighter integration of a cloud-based ES, the requirement of reliable and side-effect-free upgrade procedures that minimize the effort required of the customer gains importance over the requirement of ongoing service evolution.

Given the multi-tenant provisioning model and its limited customizability, cloud-based ES limits the means by which customers can modify their ES and the range of available modifications. First, to account for new requirements in on-premises scenarios, customers can evaluate and decide whether a new release meets their new requirements, resulting in an upgrade decision process (Khoo and Robey, 2007). However, this way of accounting for new requirements in cloud-based ES is eliminated as the provider determines scope and time of upgrades made to the service. Second, cloud-based ES limits the range of modifications that can be made to the ES (Brehm *et al.*, 2001; Light, 2001). Modification options for on-premises ES include configuration changes, adapted screen masks, and integration of plugins, which are also applicable in cloud-based scenarios. However, changes regarding the core software that involve programing, interface development, and package code modification are not possible for cloud-based ES. OM implemented custom plugins to depict special-purpose workflows not available in RightNow. However, these plugins were limited to the scope predetermined by the provider. Core functions could not be modified. For instance, when OM wanted to customize the Outlook integration workflow, no interface to modify this workflow was provided. Consequently, the workflow could not be customized, leaving OM with only the option of retaining the predetermined workflow. Therefore, we propose that the limited customizability of cloud-based ES affects the process of how requirements shape the ES as follows:

Proposition 4: Sourcing cloud-based ES limits the range of modification options to depict shifted requirements in the ES.

Cloud-based ES is hosted by an external provider, and therefore the responsibility for maintaining the service also lies with the provider. While ES requirements are affected by external events (e.g., regulatory or technological changes), in cloud-based scenarios, some external events do not affect customers' requirements or their need to act (e.g., adjust the ES) but affect the ES provider. For instance, in the case of the Heartbleed bug (OpenSSL, 2014), the need to update core services and infrastructures was the responsibility of RightNow, not OM. However, this does not generally apply to external events but reduces the number of events that affect customer requirements. Therefore, we propose that the outsourced maintenance of cloud-based ES affects the process of how events shape requirements as follows:

Proposition 5: Sourcing cloud-based ES reduces the number of external events that affect customer requirements by shifting responsibilities to act to the service provider.

When sourcing packaged ES, organizations face the challenge of selecting alternatives that offer heterogeneous functionalities and differ in their intrinsic characteristics and ability to seamlessly integrate within the organizational IT landscape (Strong and Volkoff, 2010). As packaged ES are configurable, standardized solutions designed to fit generic requirements, it is unlikely that one solution fits all of an organization's requirements (Light, 2005). Hence, the ES needs to be adjusted to meet the organization's needs. When sourcing cloud-based ES, customers are not acquainted with the services; they have to cope with a vast amount of unstructured, incomplete, and widespread information to compare service features with their requirements (Ayala and Franch, 2014). As the case of OM demonstrates, this uncertainty may lead to unfulfilled requirements that the service was thought to fulfill. Furthermore, given the UI-based configurability, customers conduct configurations of the service themselves. Hence, in cloud-based ES settings, users are involved earlier and more intensely in the sourcing process and learn about the service and its potential usage scenarios (Sawyer, 2001), for instance, when OM changed its requirements from sourcing a simple ticketing system to a CRM system. This makes sourcing cloud-based ES a particularly complex decision context that results in the need to iteratively adjust requirements, expectations, and business processes and increases the complexity for the sourcing team. Therefore, we propose that the self-service acquisition of cloud-based ES affects the processes of how experiences shape requirements and requirements shape the ES as follows:

Proposition 6: Sourcing cloud-based ES fosters end-user involvement early in the sourcing process and manifests shorter requirements evolution cycles.

Cloud-based ES is designed for high integration capabilities. Customization efforts are limited to the degree that the provider intends, and customers have to rely on the interfaces provided by either integrating ready-made plugins from third-party providers or developing own plugins. However, as OM experienced, maintaining its own custom plugins can prove troublesome and costly. RightNow did not offer a marketplace; hence, OM had to rely on custom-developed plugins and utilize the APIs to integrate external applications into workflows. For instance, OM seamlessly integrated its custom-developed fraud management solution with RightNow and established processes to handle fraud management with both services in a single workflow. Learning from the integration capabilities of RightNow, similar workflows for other divisions were established and integrated with external special-purpose applications. For OM, the requirements evolved in accordance with the possibilities the API provided, inhibiting requirements that needed customizations to the service core. Therefore, we propose that the extended service ecosystem of cloud-based ES affects the process of how the ES shapes requirements as follows:

Proposition 7: When sourcing cloud-based ES, requirements evolve in accordance with the integration capabilities of the service and the extent of the service ecosystem.

The preceding discussion highlights that the evolution of requirements in cloud-based ES is similar to that in on-premises ES but differs in the details of the mechanisms. Hence, the developed theory applies to packaged ES in general, with the propositions regarding the specifics of cloud-based ES being slight adjustments to the mechanisms

Table 11 Cloud-specifics of the mechanisms shaping the evolution of requirements

Proposition	*Contextual factors*	*Affected mechanism*	*Effect of cloud-specifics on requirements evolution*
1	Self-service acquisition	Requirements importance (9)	Technological requirements increase in importance later in the ES life cycle
2	UI-based configuration	Requirements importance (9)	Requirements of ease of configuration and reliable customer support gain importance
3	Limited upgrade control	Requirements importance (9)	Requirements of reliable and side-effect-free upgrade procedures gain importance
4	Limited customization	Requirements shape ES (8)	Limited range of modification options to depict shifted requirements
5	Outsourced maintenance	Events shape requirements (6)	Reduced amount of external events that affect customer requirements
6	Self-service acquisition	Experiences shape requirements (5), requirements shape ES (8)	End-user involvement fostered early in the sourcing process, short cycles to reflect requirements in the ES
7	Extended service ecosystem	ES shape requirements (7)	Requirements evolution aligned with the integration capabilities of the ES and the extent of the service ecosystem

Note: Numbers in brackets in the column 'Affected mechanisms' refer to the circled numbers in Figure 3, respectively the ID column in Table 10.

within the process. Table 11 summarizes how the mechanisms of requirements evolution are affected by cloud-specific contextual factors.

Implications for research and practice

This research provides a theoretical grounding for further research. First, we conceptually identified the peculiarities of sourcing cloud-based ES and discussed their influence on the evolution of requirements. The peculiarities can be employed as analytical devices and transferred to other research settings to identify cloud-specific implications, for instance, implications for decision-making processes or critical success factors. Second, as we conducted a single-case study and gathered empirical data from one particular research setting, we would welcome additional qualitative or quantitative research to further challenge and enhance our theory and propositions. Third, the focus of our theory development effort is to explain how context and experiences shape requirements. We would welcome research that extends our work and explores other closely related mechanisms that our analysis only touched upon, for instance, how the evolution of requirements shapes the context. Finally, we encourage researchers to expand the analysis beyond the context and experiences shaping requirements, for instance, investigating how social interactions, conflicting group interests, and political behavior shape the evolution of requirements. Recent research offers promising results regarding the 'social shaping' of decision-making processes (Wilson and Howcroft, 2005; Howcroft and Light, 2010; Bidwell, 2012), and our case indicated political behavior by the sales division (e.g., resistance to RightNow resulted in mobile access evolving as a new requirement).

Our discussion of how the context shaped the process and requirements solely focused on the contextual factors that are specific to the cloud-sourcing context. However, other contextual factors not specific to cloud sourcing also influenced the evolution of requirements and require further consideration. For instance, ES sourcing projects serve the specific purpose of addressing technological (e.g., the end of the life cycle of an existing solution), operational (e.g., process improvement), or strategic (e.g., strategic change) needs of an organization (Verville *et al.*, 2005). The characteristics of these needs (e.g., associated risks, urgency) determine the existence and prioritization of requirements, as in the case of OM (urgency of implementation). Specifically, our findings indicate that contextual factors on the decision-specific, environmental, organizational, group, and individual levels require reconsideration in light of the technological characteristics of cloud sourcing. For instance, as the case of Alpha shows, the ease and degree of UI-based configuration of cloud services expands the role of individual business executives' IT affinity. With the underlying self-service principle of cloud computing, organization-wide governance of IT-procurement on the application level becomes increasingly important (Winkler and Brown, 2013) to avoid single business divisions pursuing their distinct business goals to procure dedicated cloud services autonomously. Additionally, the sourcing of highly complex and specific services that need to be adapted to organizations' needs (e.g., ES) are perceived to require only limited IT resources in cloud-sourcing contexts when divisions configure the services via the UI. We therefore emphasize the need for further research to investigate the interplay of cloud-specific and other non-cloud-specific contextual factors and the implications for activities throughout the ES life cycle (such as decision-making and requirements determination).

The fact that OM initiated a re-evaluation of RightNow because OM's sales staff rejected the service, while RightNow created substantial business value for all other divisions, emphasizes the influence of non-management employees on organizational decision-making. Recent research supports this argument (Wilson and Howcroft, 2005; Howcroft and Light,

2010) and calls for the application of multi-stakeholder perspectives to understand decision-making in cloud-sourcing contexts (Benlian *et al.*, 2009). While research on group interests predominantly focuses on managers' interests (Bidwell, 2012), we collected data from multiple relevant stakeholders within the organization, including top management and operational staff. We regard this approach as particularly fruitful and recommend that future studies consider data from multiple stakeholders within an organization and further leverage the insights that such data may provide. For instance, future research could illuminate how requirements evolve in different stakeholder groups, how alternatives are evaluated or perceived differently, or how the influence of specific contextual factors varies across groups.

This research provides several implications for practice. By identifying mechanisms that shape the evolution of requirements, customers of cloud-based ES can learn from our findings. Knowing the mechanisms that shape the evolution of requirements over time supports managers in controlling their processes, efficiently mitigating challenges, and actively monitoring the evolution of requirements to proactively assess alternatives on the market. The evolution of requirements for cloud-based ES follows similar patterns to that of on-premises ES but requires further attention to specific processes. For instance, when acquiring cloud-based ES, it is even more important to evaluate not only the service itself but also the interfaces and ecosystem, as requirements evolve along with the integration capabilities of the service.

Customers of cloud-based ES can learn from Alpha's experiences and adjust their processes throughout the ES life cycle. Regarding the acquisition of cloud-based ES, customers need to enforce a rigorous and iterative evaluation of trial services to assure business support and the fulfillment of each requirement. When integrating cloud-based ES into the IT landscape, integration and maintenance capabilities need to be established and the amount of ongoing internal effort required for configurations and integrations must be considered. When operating cloud-based ES, customers need to establish a cloud-ready release management. The scope and time of scheduled service upgrades needs to be monitored, and integrated systems and services need to be reviewed for dependencies to avoid service outages. When considering switching to alternative solutions for an established cloud-based ES, switching costs need to be evaluated considering the effort necessary to achieve the same level of integration with and alignment of the service to the organization, which is considerably high for ES.

Providers of cloud-based ES can learn insights from the case by considering OM's requirements and how they evolved over time. For instance, providers should establish a high-quality support community with reference sites, active and helpful support forums, and multiple communication and self-service support channels such as demo services, tutorials, and training materials. Furthermore, providers need to maintain reliable upgrade practices with clear customer involvement. To avoid downtime and errors when rolling out service upgrades in highly customized environments, providers need to closely collaborate with customers and provide support to identify potential impacts of upgrades on customers' configurations and customizations upfront.

Conclusion

Following Alpha's sourcing activities over a timeframe of 5 years enables us to trace how contextual factors of the sourcing context and experiences with the ES are intertwined and shape the evolution of requirements throughout the ES life cycle. Hence, we are able to develop theoretical grounding that focuses on the dynamics of ES requirements beyond the implementation stage. We isolate nine mechanisms that explain how requirements evolve and discuss how sourcing cloud-based ES (as opposed to sourcing on-premises ES) alters the mechanisms that shape the evolution of requirements. Our findings show that the evolution of requirements for cloud-based ES follows similar mechanisms to the evolution of requirements for on-premises ES but changes how particular mechanisms manifest, including the influence of business divisions on requirements, the role of upgrade and customization procedures on how requirements shape the ES, and how the ES' ecosystem shapes requirements.

We provide two unique contributions to research and practice. First, the developed process theory sheds light on mechanisms that shape the evolution of ES requirements. Hence, we advance existing research, as the theory poses a nuanced and distinct understanding of how ES requirements evolve over time that goes beyond the level of detail investigated in existing research. We thereby answer calls to study the dynamics of the IT artifact under consideration (Orlikowski and Iacono, 2001), to investigate how ES evolves beyond adoption and acquisition (Howcroft and Light, 2010; Williams and Pollock, 2012), and to explain how different contextual factors and experiences with the ES shape the evolution of organizations' ES requirements over time (Jarke *et al.*, 2011). Second, by explicitly theorizing on the peculiarities of cloud-based ES in our research (Orlikowski and Iacono, 2001), we emphasize the specifics of cloud computing as a sourcing option for ES and its implications for organizational processes (Schneider and Sunyaev, 2016). Hence, the discussion aids researchers and practitioners in identifying activities and requirements that demand specific attention when sourcing cloud-based ES.

Acknowledgments

The research was conducted in context of the NGCert research project, funded by the German Federal Ministry for Education and Research (grant no. 16KIS0079). The authors would like to thank the Senior Editor and anonymous reviewers for their valuable comments and suggestions to improve the quality of this paper.

Notes

1 Research provides ES life cycle models with various conceptualizations of phases and activities (e.g., Markus and Tanis, 2000; Verville and Halingten, 2003; Esteves and Bohorquez, 2007; Schneider and Sunyaev, 2015). We deliberately abstract from adopting a specific life cycle model in our theoretical pre-understanding. We instead conceptualize the ES life cycle as the series of events and activities related to an organization's ES sourcing efforts, beginning with the decision to adopt an ES, continuing with acquisition, implementation, integration, use, and maintenance activities, and ending with retiring the ES. The activities during the life cycle are not confined to linear and

sequential execution but are instead iteratively intertwined and overlapping, caused by events that occur during the process.

References

Armbrust, M., Fox, A., Griffith, R., Joseph, A.D., Katz, R., Konwinski, A., et al. (2010). A View of Cloud Computing, *Communications of the ACM* **53**(4): 50–58.

Arthur, L.J. (1988), *Software Evolution: The Software Maintenance Challenge*. New York: Wiley.

Ayala, C. and Franch, X. (2014). Dealing with Information Quality in Software Package Selection: An empirically based approach, *IET Software* **8**(5): 204–218.

Beimborn, D., Miletzki, T. and Wenzel, S. (2011). Platform as a Service (PaaS), *Business & Information Systems Engineering* **3**(6): 381–384.

Benlian, A. and Hess, T. (2010). Does Personality Matter in the Evaluation of ERP Systems? Findings from a Conjoint Study, in Proceedings of the 18th European Conference on Information Systems (Pretoria, South Africa).

Benlian, A., Hess, T. and Buxmann, P. (2009). Drivers of SaaS-Adoption: An empirical study of different application types, *Business & Information Systems Engineering* **1**(5): 357–369.

Bidwell, M.J. (2012). Politics and Firm Boundaries: How organizational structure, *Group Interests, and Resources Affect Outsourcing, Organization Science* **23**(6): 1622–1642.

Brehm, L., Heinzl, A. and Markus, M.L. (2001). Tailoring ERP Systems: A Spectrum of Choices and their Implications, in Proceedings of the 34th Hawaii International Conference on System Sciences (Maui, Hawaii, USA).

Byrd, T.A., Cossick, K.L. and Zmud, R.W. (1992). A Synthesis of Research on Requirements Analysis and Knowledge Acquisition Techniques, *MIS Quarterly* **16**(1): 117–138.

Cavaye, A.L. (1995). User Participation in System Development Revisited, *Information & Management* **28**(5): 311–323.

Chakraborty, S., Sarker, S. and Sarker, S. (2010). An Exploration into the Process of Requirements Elicitation: A grounded approach, *Journal of the AIS* **11**(4): 212–249.

Choudhary, V. (2007). Comparison of Software Quality Under Perpetual Licensing and Software as a Service, *Journal of Management Information Systems* **24**(2): 141–165.

Cusumano, M.A. (2007). The Changing Labyrinth of Software Pricing, *Communications of the ACM* **50**(7): 19–22.

Davidson, E.J. (2002). Technology Frames and Framing: A socio-cognitive investigation of requirements determination, *MIS Quarterly* **26**(4): 329–358.

Dubé, L. and Paré, G. (2003). Rigor in Information Systems Positivist Case Research: current practices, *Trends, and Recommendations, MIS Quarterly* **27**(4): 597–635.

Eden, R., Sedera, D. and Tan, F. (2014). Sustaining the Momentum: Archival analysis of enterprise resource planning systems (2006–2012), *Communications of the AIS* **35**(1): 40–82.

Esteves, J. and Bohorquez, V. (2007). An Updated ERP Systems Annotated Bibliography: 2001–2005, *Communications of the AIS,* **19**(1): 386–446.

Flynn, D. (1998), *Information Systems Requirements: Determination and Analysis*. London: McGraw-Hill.

Griffiths, M. and Light, B. (2009). An Investigation into Resistance Practices at an SME Consultancy, *Journal of Enterprise Information Management* **22**(1/2): 119–136.

Hickey, A.M. and Davis, A.M. (2004). A Unified Model of Requirements Elicitation, *Journal of Management Information Systems* **20**(4): 65–84.

Hofmann, H.F. and Lehner, F. (2001). Requirements Engineering as a Success Factor in Software Projects, *IEEE Software* **18**(4): 58–66.

Holmström, J. and Sawyer, S. (2011). Requirements Engineering Blinders: Exploring information systems developers' black-boxing of the emergent character of requirements, *European Journal of Information Systems* **20**(1): 34–47.

Howcroft, D. and Light, B. (2002). A Study of User Involvement in Packaged Software Selection, in Proceedings of the 23rd International Conference on Information Systems (Barcelona, Spain).

Howcroft, D. and Light, B. (2006). Reflections on Issues of Power in Packaged Software Selection, *Information Systems Journal* **16**(3): 215–235.

Howcroft, D. and Light, B. (2010). The Social Shaping of Packaged Software Selection, *Journal of the AIS* **11**(3): 122–148.

Howcroft, D., Newell, S. and Wagner, E. (2004). Understanding the Contextual Influences on Enterprise System Design, Implementation, Use and Evaluation, *The Journal of Strategic Information Systems* **13**(4): 271–277.

Jarke, M., Loucopoulos, P., Lyytinen, K., Mylopoulos, J. and Robinson, W. (2011). The Brave New World of Design Requirements, *Information Systems* **36**(7): 992–1008.

Jutras, C. (2015). The Three Dimensions of Sap Business Bydesign Set the Stage for Growth [WWW document] http://www.sap.com/bin/sapcom/en_us/downloadasset.2015-02-feb-17-00.the-three-dimensions-of-sap-business-bydesign-set-the-stage-for-growth-pdf.bypassReg.html.

Kaasbøll, J.J. (1997). How Evolution of Information Systems May Fail: Many improvements adding up to negative effects, *European Journal of Information Systems* **6**(3): 172–180.

Kawalek, P. and Leonard, J. (1996). Evolutionary Software Development to Support Organizational and Business Process Change: A case study account, *Journal of Information Technology* **11**(3): 185–198.

Kawalek, P. and Wood-Harper, T. (2002). The Finding of thorns: user participation in enterprise system implementation, *ACM SIGMIS Database* **33**(1): 13–22.

Keutel, M., Michalik, B. and Richter, J. (2014). Towards Mindful Case Study Research in IS: A critical analysis of the past 10 years, *European Journal of Information Systems* **23**(3): 256–272.

Khoo, H.M. and Robey, D. (2007). Deciding to Upgrade Packaged Software: A comparative case study of motives, contingencies and dependencies, *European Journal of Information Systems* **16**(5): 555–567.

Light, B. (2001). The Maintenance Implications of the Customization of ERP Software, *Journal of Software Maintenance and Evolution: Research and Practice* **13**(6): 415–429.

Light, B. (2005). Going Beyond 'Misfit' as a Reason for ERP Package Customisation, *Computers in Industry* **56**(6): 606–619.

Light, B. and Wagner, E. (2006). Integration in ERP Environments: Rhetoric, realities and organisational possibilities, *New Technology, Work and Employment* **21**(3): 215–228.

Lins, S., Schneider, S. and Sunyaev, A. (2016). Trust is Good, Control is Better: creating secure clouds by continuous auditing, *IEEE Transactions on Cloud Computing* forthcoming: 1–14.

Lucas, H.C.J., Walton, E.J. and Ginzberg, M.J. (1988). Implementing Packaged Software, *MIS Quarterly* **12**(4): 536–549.

Markus, M.L. and Tanis, C. (2000). The Enterprise System Experience: From Adoption to Success. In R. W. Zmud & M. F. Price (Eds.), *Framing the Domains of IT-Management: Projecting the future through the Past* (pp. 173–207). Cincinnati, OH: Pinnaflex Educational Resources.

Marston, S., Li, Z., Bandyopadhyay, S., Zhang, J. and Ghalsasi, A. (2011). Cloud Computing: The business perspective, *Decision Support Systems* **51**(1): 176–189.

Mathiassen, L., Saarinen, T., Tuunanen, T. and Rossi, M. (2007). A Contingency Model for Requirements Development, *Journal of the AIS* **8**(11): 569–597.

McGee, S. and Greer, D. (2012). Towards an Understanding of the causes and Effects of Software Requirements Change: Two case studies, *Requirements Engineering* **17**(2): 133–155.

Melgarejo, P. (2012). Worldwide Software 2012 – 2016 Forecast Summary [WWW document] http://www.idc.com/research/viewtoc.jsp?containerId=235326.

Mell, P. and Grance, T. (2011). A Nist Definition of Cloud Computing [WWW document] http://csrc.nist.gov/publications/nistpubs/800-145/SP800-145.pdf (accessed 28 April 2015).

OpenSSL (2014). Openssl Vulnerabilities [WWW document] https://www.openssl.org/news/vulnerabilities.html (accessed 21 November 2015).

Orlikowski, W.J. (1996). Improvising Organizational Transformation Over time: A situated change perspective, *Information Systems Research* **7**(1): 63–92.

Orlikowski, W.J. and Iacono, C. S. (2001). Research Commentary: Desperately seeking the "IT" in IT research—A call to theorizing the IT artifact, *Information Systems Research* **12**(2): 121–134.

Pollock, N. and Williams, R. (2007). Technology Choice and its Performance: Towards a sociology of software package procurement, *Information and Organization* **17**(3): 131–161.

Sarker, S., Sarker, S., Sahaym, A. and Bjørn-Andersen, N. (2012). Exploring Value Cocreation in Relationships Between an ERP Vendor and its Partners: A revelatory case study, *MIS Quarterly* **36**(1): 317–338.

Sawyer, S. (2000). Packaged Software: Implications of the differences from custom approaches to software development, *European Journal of Information Systems* **9**(1): 47–58.

Sawyer, S. (2001). A Market-Based Perspective on Information Systems Development, *Communications of the ACM* **44**(11): 97–102.

Schneider, S. and Sunyaev, A. (2015). Cloud live: a life cycle framework for cloud services, *Electronic Markets* **25**(4): 299–311.

Schneider, S. and Sunyaev, A. (2016). Determinant Factors of Cloud-Sourcing Decisions: Reflecting on the IT outsourcing literature in the era of cloud computing, *Journal of Information Technology* **31**(1): 1–31.

Seddon, P.B., Calvert, C. and Yang, S. (2010). A Multi-Project Model of Key Factors Affecting Organizational Benefits From Enterprise Systems, *MIS Quarterly* **34**(2): 305–328.

Shepherd, N.G. and Rudd, J.M. (2014). The Influence of Context on the Strategic Decision-Making Process: A review of the literature, *International Journal of Management Reviews* **16**(3): 340–364.

Strong, D.M. and Volkoff, O. (2010). Understanding Organization-Enterprise System Fit: A path to theorizing the information technology artifact, *MIS Quarterly* **34**(4): 731–756.

Sunyaev, A. and Schneider, S. (2013). Cloud Services Certification, *Communications of the ACM* **56**(2): 33–36.

Susarla, A., Barua, A. and Whinston, A.B. (2010). Multitask Agency, Modular Architecture, and Task Disaggregation in SaaS, *Journal of Management Information Systems* **26**(4): 87–118.

Tornatzky, L.G. and Fleischer, M. (1990), *The Processes of Technological Innovation*, Lexington, MA: Lexington Books.

Van de Ven, A.H., and Huber, G.P. (1990). Longitudinal Field Research Methods for Studying Processes of Organizational Change, *Organization Science* **1**(3): 213–219.

Van de Ven, A.H. and Poole, M.S. (1990). Methods for Studying Innovation Development in the Minnesota Innovation Research Program, *Organization Science* **1**(3): 313–335.

Verville, J., Bernadas, C. and Halingten, A. (2005). So You're Thinking of Buying an ERP? Ten critical factors for successful acquisitions, *Journal of Enterprise Information Management* **18**(6): 665–677.

Verville, J. and Halingten, A. (2003). A Six-Stage Model of the Buying Process for ERP Software, *Industrial Marketing Management* **32**(7): 585–594.

Wagner, E.L. and Newell, S. (2007). Exploring the Importance Of Participation in the Post-Implementation Period of an ES Project: A neglected area, *Journal of the AIS* **8**(10): 508–524.

Walther, S., Sarker, S., Sedera, D. and Eymann, T. (2013). Exploring Subscription Renewal Intention of Operational Cloud Enterprise Systems: A Socio-Technical Approach, in Proceedings of the 21st European Conference on Information Systems (Utrecht, Netherlands).

Webster, F.E, Jr. and Wind, Y. (1972). A General Model for Understanding Organizational Buying Behavior, *Journal of Marketing* **36**(2): 12–19.

Williams, R. and Pollock, N. (2012). Moving Beyond the Single Site Implementation Study: How (and why) we should study the biography of packaged enterprise solutions, *Information Systems Research 23*(1): 1–22.

Wilson, M. and Howcroft, D. (2005). Power, Politics and Persuasion in IS Evaluation: A Focus on 'Relevant Social Groups', *The Journal of Strategic Information Systems* **14**(1): 17–43.

Winkler, T. J. and Brown, C. V. (2013). Horizontal Allocation of Decision Rights for On-Premise Applications and Software-as-a-Service, *Journal of Management Information Systems* **30**(3): 13–48.

Xin, M. and Levina, N. (2008). Software-as-a-Service Model: Elaborating Client-Side Adoption Factors, in Proceedings of the 29th International Conference on Information Systems (Paris, France).

Yin, R.K. (2009), *Case Study Research: Design and Methods* (4th ed.), Thousand Oaks, CA: Sage.

About the Authors

Stephan Schneider is a post-doctoral researcher at the Department of Information Systems, University of Cologne, Germany. His research focuses on strategic decision-making in cloud-sourcing projects as well as on cloud security and certification of cloud computing infrastructures. He has (co)-authored several books and international journal articles (e.g., Journal of Information Technology, Communications of the ACM, IEEE Transactions on Cloud Computing) and presented his work at major IS conferences (ICIS, ECIS, HICSS).

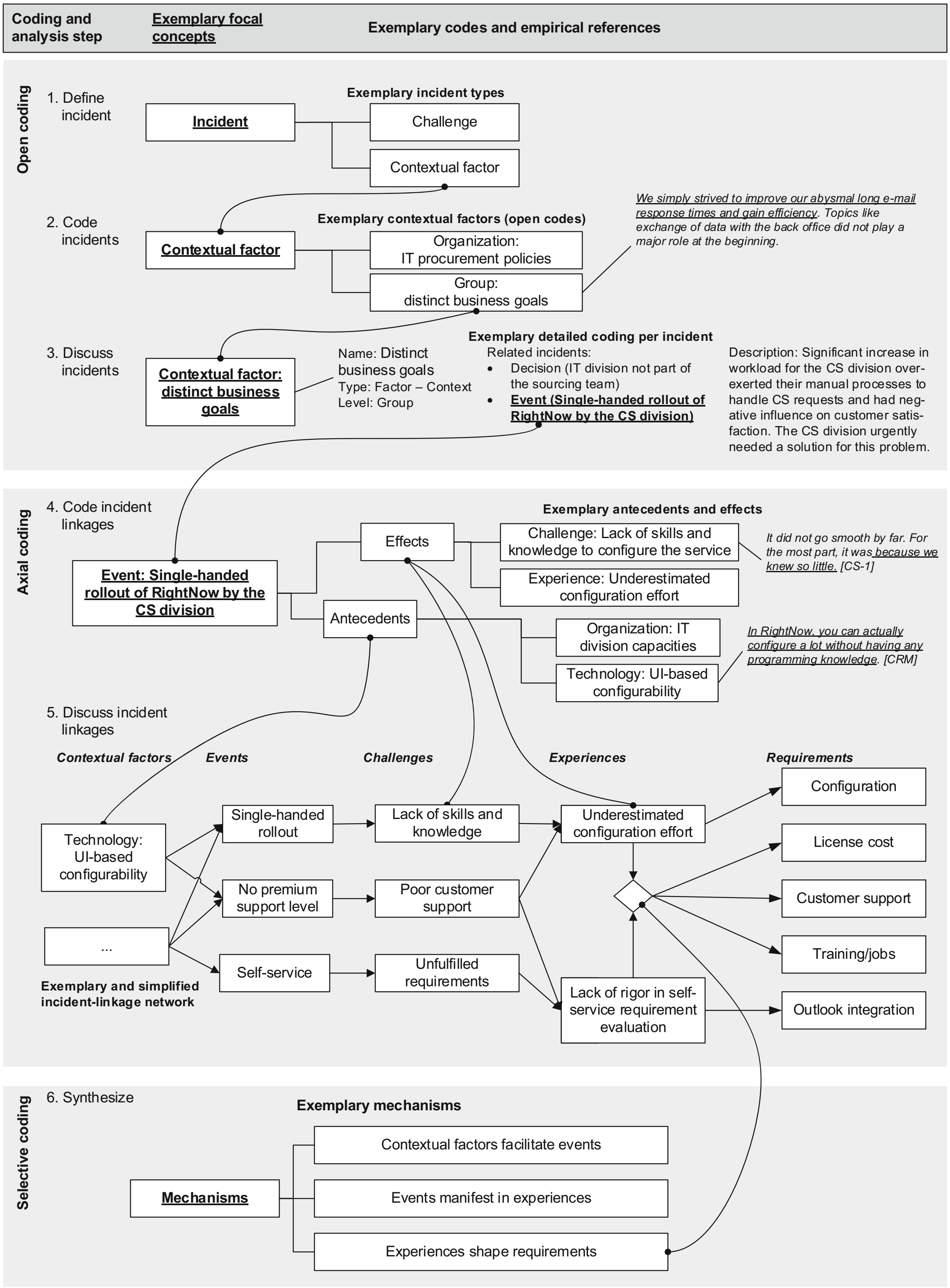
Jan Wollersheim is a full-time researcher at fortiss GmbH – An-Institut Technische Universität München. Prior to that, he worked for Saba Software as Business Consultant EMEA and conducted internships with multiple international companies. Jan Wollersheim received a M.Sc. in Information Systems from Technische Universität München and a B.Sc. in Information Systems from Hohenheim University and the University of Stuttgart. His research interests focus service management, IT-procurement, service quality analysis, and the application of cloud services by organizations.

Helmut Krcmar has been a Full Professor of Information Systems and holds the Chair for Information Systems at the Department of Informatics, Technische Universität München, Germany, since 2002. He worked as Post Doctoral Fellow at the IBM Los Angeles Scientific Center, as Assistant Professor of Information Systems at the Leonard Stern School of Business, NYU and at Baruch College, CUNY. From 1987 to 2002, he held the Chair for Information Systems, Hohenheim University, Stuttgart. His research interests include information and knowledge management, service management, information systems in health care and electronic government.

Ali Sunyaev holds the Chair for Information Systems and Systems Engineering and is Director of the Research Center for IS Design (ITeG) at the University of Kassel, Germany. His research interests include design and management of information systems, development of innovative healthcare applications, and information privacy. His work has been published in international journals such as Journal of Information Technology, Communications of the ACM, IEEE Software, and IEEE Transactions on Cloud Computing.

Appendix A

See Figure 4.


Coding and analysis step
Exemplary focal concepts
Exemplary codes and empirical references
Open coding
1. Define incident
Incident
Exemplary incident types
Challenge
Contextual factor
2. Code incidents
Contextual factor
Exemplary contextual factors (open codes)
Organization: IT procurement policies
Group: distinct business goals
3. Discuss incidents
Contextual factor: distinct business goals
Name: Distinct business goals
Type: Factor – Context
Level: Group
Exemplary detailed coding per incident
Related incidents:
• Decision (IT division not part of the sourcing team)
• Event (Single-handed rollout of RightNow by the CS division)
Axial coding
4. Code incident linkages
Event: Single-handed rollout of RightNow by the CS division
Effects
Antecedents
Exemplary antecedents and effects
Challenge: Lack of skills and knowledge to configure the service
Experience: Underestimated configuration effort
Organization: IT division capacities
Technology: UI-based configurability
5. Discuss incident linkages
Contextual factors
Events
Challenges
Experiences
Requirements
Technology: UI-based configurability
...
Single-handed rollout
No premium support level
Self-service
Lack of skills and knowledge
Poor customer support
Unfulfilled requirements
Underestimated configuration effort
Lack of rigor in self-service requirement evaluation
Configuration
License cost
Customer support
Training/jobs
Outlook integration
Exemplary and simplified incident-linkage network
Selective coding
6. Synthesize
Exemplary mechanisms
Mechanisms
Contextual factors facilitate events
Events manifest in experiences
Experiences shape requirements

We simply strived to improve our abysmal long e-mail response times and gain efficiency. Topics like exchange of data with the back office did not play a major role at the beginning.

Description: Significant increase in workload for the CS division overexerted their manual processes to handle CS requests and had negative influence on customer satisfaction. The CS division urgently needed a solution for this problem.

It did not go smooth by far. For the most part, it was because we knew so little. [CS-1]

In RightNow, you can actually configure a lot without having any programming knowledge. [CRM]

Figure 4 Examples and visualization of data coding and analysis procedures.