

A Service of

ZBW

Leibniz-Informationszentrum Wirtschaft Leibniz Information Centre for Economics

Kopper, Andreas; Westner, Markus; Strahringer, Susanne

Article — Published Version

From Shadow IT to Business-managed IT: a qualitative comparative analysis to determine configurations for successful management of IT by business entities

Information Systems and e-Business Management

Provided in Cooperation with:

Springer Nature

Suggested Citation: Kopper, Andreas; Westner, Markus; Strahringer, Susanne (2020) : From Shadow IT to Business-managed IT: a qualitative comparative analysis to determine configurations for successful management of IT by business entities, Information Systems and e-Business Management, ISSN 1617-9854, Springer, Berlin, Heidelberg, Vol. 18, Iss. 2, pp. 209-257, https://doi.org/10.1007/s10257-020-00472-6

This Version is available at: https://hdl.handle.net/10419/288329

Standard-Nutzungsbedingungen:

Die Dokumente auf EconStor dürfen zu eigenen wissenschaftlichen Zwecken und zum Privatgebrauch gespeichert und kopiert werden.

Sie dürfen die Dokumente nicht für öffentliche oder kommerzielle Zwecke vervielfältigen, öffentlich ausstellen, öffentlich zugänglich machen, vertreiben oder anderweitig nutzen.

Sofern die Verfasser die Dokumente unter Open-Content-Lizenzen (insbesondere CC-Lizenzen) zur Verfügung gestellt haben sollten, gelten abweichend von diesen Nutzungsbedingungen die in der dort genannten Lizenz gewährten Nutzungsrechte.



https://creativecommons.org/licenses/by/4.0/

Terms of use:

Documents in EconStor may be saved and copied for your personal and scholarly purposes.

You are not to copy documents for public or commercial purposes, to exhibit the documents publicly, to make them publicly available on the internet, or to distribute or otherwise use the documents in public.

If the documents have been made available under an Open Content Licence (especially Creative Commons Licences), you may exercise further usage rights as specified in the indicated licence.





ORIGINAL ARTICLE



From Shadow IT to Business-managed IT: a qualitative comparative analysis to determine configurations for successful management of IT by business entities

Andreas Kopper¹ · Markus Westner² · Susanne Strahringer¹

Received: 23 June 2019 / Revised: 14 May 2020 / Accepted: 28 June 2020 / Published online: 15 July 2020 © The Author(s) 2020

Abstract

Shadow IT describes covert/hidden IT systems that are managed by business entities themselves. Additionally, there are also overt forms in practice, so-called Businessmanaged IT, which share most of the characteristics of Shadow IT. To better understand this phenomenon, we interviewed 29 executive IT managers about positive and negative cases of Shadow IT and Business-managed IT. By applying qualitative comparative analysis (QCA), we derived four conditions that characterize these cases: Aligned, local, simple, and volatile. The results show that there are three sufficient configurations of conditions that lead to a positive outcome; one of them even encompasses Shadow IT. The most important solution indicates that IT systems managed by business entities are viewed as being positive if they are aligned with the IT department and limited to local requirements. This allows to balance local responsiveness to changing requirements and global standardization. In contrast, IT systems that are not aligned and permanent (and either organization-wide or simple) are consistently considered as negative. Our study is the first empirical quantitativequalitative study to shed light on the success and failure of Shadow IT and Businessmanaged IT.

Keywords Business-managed IT \cdot Shadow IT \cdot IT governance \cdot Qualitative comparative analysis (QCA)

Susanne Strahringer susanne.strahringer@tu-dresden.de

Electronic supplementary material The online version of this article (https://doi.org/10.1007/s1025 7-020-00472-6) contains supplementary material, which is available to authorized users.

¹ TU Dresden, Helmholtzstr. 10, 01069 Dresden, Germany

² OTH Regensburg, Pruefeninger Str. 58, 93049 Regensburg, Germany

1 Introduction

Many businesses today use information technology (IT) systems that are "in the shadows" for the IT department which is expected to be aware of them. Kopper et al. (2018b), for example, report on an event management system sourced by marketing staff because of an upcoming exhibition and the need to manage participants. Because the marketing department knew the IT department would not be able to provide a suitable system as quickly as needed, marketing simply sourced a software as service (SaaS) solution, adapted it, and used spreadsheets for data transfer. Only later, IT managers became aware of the system. They only had moderate levels of concerns about the system's covert usage but still planned to address the situation to mitigate potential risks (Kopper et al. 2018b; Zimmermann et al. 2017). The mentioned event management system which was hidden from IT is an example for the phenomenon called Shadow IT.

Shadow IT is defined as all software (incl. SaaS, PaaS, IaaS), hardware, or IT service processes that are used or created autonomously by business units (BUs) without alignment with the IT department (Kopper and Westner 2016b; Zimmermann et al. 2014). We define the term BU to include all types of business entities (individual users and business workgroups/units/departments/divisions) and subsequently use it accordingly. Shadow IT is highly common and relevant in practice. Gartner (2017) has estimated that 38% of technology purchases are managed, defined, and controlled by business leaders. Capgemini (2016) has reported that in more than 60% of companies, BUs were given direct control for certain IT investments (such as consulting services for pilot projects). A survey also found that less than 50% of IT and non-IT workers believe their Chief Information Officers (CIOs) are aware of digital technology problems that affect them (Gartner 2018). This could be a reason that 50–70% of all employees use Shadow IT (Chua et al. 2014). Shadow IT imposes both risks and opportunities. Researchers commonly highlight security risks, inefficiencies, or integration issues as potential negative effects (Györy et al. 2012; Hetzenecker et al. 2012; Kretzer and Maedche 2014). However, recent academic and practitioner literature increasingly addressed the associated opportunities such as increased agility, productivity, or innovation (Behrens 2009; Singh 2015).

Although there are IT systems that are truly hidden from the organizational IT management, it changes as soon as they become visible, for example, due to monitoring mechanisms (Buchwald et al. 2014a). As in our introductory marketing case vignette above, a decision has to follow after detection: IT departments can categorize a solution and then either decommission, fully take control, co-govern with the BU, or leave the responsibility with its creators in the BU, therefore legitimizing it (Chua et al. 2014; Zimmermann et al. 2016a). Furthermore, cases of instances can be observed in practice that share most of the characteristics of Shadow IT but that are deliberately and openly encouraged by the IT department (Kopper 2017). To enable a more nuanced understanding of this phenomenon, Kopper et al. (2018a, b) introduced a terminology to distinguish between covert (hidden) Shadow IT (SIT) and overt (visible) Business-managed IT (BMIT).

Despite its relevant implications for governance practices, the professional and academic discourse around SIT does not yet clearly discern between the two terms because instances generally share mostly the same characteristics. For both, IT task responsibility usually lies with the BU (the only exception would be SIT within the IT department itself). However, BMIT is (overtly) managed in agreement with the organizational IT in contrast to SIT. BMIT is therefore similarly defined as all software (incl. SaaS, PaaS, IaaS), hardware, or IT service processes that are overtly created/procured or managed by business entities (individual users, business workgroups, or BUs) either in alignment with the IT department or in a split responsibility model (Kopper et al. 2018b).

The introduction of the term "Business-managed IT" allows scholars to better understand the phenomenon, for example, to study the lifecycle of instances that may migrate from SIT to BMIT due to their detection. The phenomenon of interest in this study is, therefore, IT managed in and by BUs, which includes both covert SIT and overt BMIT. As initially mentioned, scholars and practitioners recognize both their associated risks and opportunities. However, empirical research on the characteristics of its manifestations is sparse. Particularly, the current body of research does not explain which characteristics influence or determine whether SIT/ BMIT is perceived positively, i.e., offering potential value, in contrast to a negative perception where adverse risks may materialize. The paper at hand closes this gap from an IT management perspective, as IT management is usually responsible for balancing IT value and risk. Our research questions are, therefore, as follows:

- RQ1: What can be considered a positive (or respectively a negative) form of Shadow IT (SIT) and Business-managed IT (BMIT) from an IT management perspective?
- RQ2: What are the characteristics of SIT/BMIT instances that influence whether SIT/BMIT is perceived positively or negatively from an IT management perspective?
- RQ3: What combinations of these characteristics clearly yield either positive or negative forms of SIT/BMIT?

To answer those questions, we employ the qualitative comparative analysis (QCA) methodology (Rihoux and Ragin 2008). The research field of SIT/BMIT is still developing and primarily relying on qualitative-explorative approaches (Kopper and Westner 2016a; Magunduni and Chigona 2018). One of the reasons for choosing QCA was, therefore, to help the field to mature by using a qualitative-quantitative approach.

Our results are relevant for both researchers and practitioners. Scholars can use them to get a better conceptual understanding of SIT/BMIT and to develop appropriate governance mechanisms. Using the identified characteristics and their influential combinations, practitioners can also make systematic decisions about the distribution of IT task responsibilities.

The paper at hand is structured as follows: In Sect. 2, we review the state-ofthe-art through a literature review and define the phenomenon and terminology of interest. In Sect. 3, we describe the research methodology, data collection, and the analysis using QCA methodology. We consciously covered the QCA analysis in detail to allow the interested reader to follow our approach with regard to replicability. In Sect. 4, we present the results in the form of the (inductively and deductively derived) dependent (RQ1) and independent variables (RQ2) and combinations of the latter that yield either positive or negative forms of SIT/BMIT (RQ3). In Sect. 5, we summarize the findings and discuss balancing autonomy and control in IT governance setups. In Sect. 6, we conclude with remarks on the paper's contributions, its limitations, and possibilities for future research.

2 State-of-the-art

2.1 Literature review

To capture the state-of-the-art of academic research around SIT/BMIT we conducted a literature review (Levy and Ellis 2006). Because we only recently published a paper in this field that contains an up-to-date literature review, we largely recapture our findings in this section (Klotz et al. 2019).

In our review, we found three related themes in the identified publications: Positive and negative consequences of SIT, aspects about IT governance and organizational structure, and systems/platforms for end-user IT/development.

Potential negative consequences or risks of SIT are well covered in academic research (Kopper and Westner 2016a). The phenomenon can lead to inefficiencies due to loss of synergies or scale effects (Györy et al. 2012; Kretzer and Maedche 2014), can pose data security and data inconsistency risks (Györy et al. 2012; Kretzer and Maedche 2014; Silic and Back 2014), and may be prone to integration issues (Hetzenecker et al. 2012). However, there is also an emerging theme of recognizing the *positive potential* or value of SIT among researchers. Behrens (2009) has highlighted that SIT can be a source of creativity and innovation. It helps employees adapting to environmental changes (Singh 2015) and is generally intended to benefit the organization (Buchwald et al. 2014b), although conducted covertly (Ferneley 2007). Köffer et al. (2015) have suggested that the diffusion of consumer IT within organizations is beneficial for innovation. Fürstenau and Rothe (2014) propose a method to discern "good" and "bad" SIT systems based on their architectural embeddedness. Silic et al. (2016) have noted that leveraging users' innovation potential in the form of SIT is often overlooked and needs to be addressed from an IT governance perspective.

There is also a theme that focuses on *aspects of IT governance and organizational structure*. Chua and Storey (2016) have concluded that bottom-up initiatives are inevitable and that the IT department must work together with BUs to guide them. Governing and monitoring user-driven innovation initiatives also allows mitigating their risks (Györy et al. 2012) because covert initiatives might pose a higher security risk due to the inability to control them (Haag and Eckhardt 2017). Zimmermann et al. (2016a), therefore, deal with the allocation of IT task responsibilities of detected SIT. On a more general level, Andriole (2015) suggests a participatory



Fig. 1 Conceptual framework for defining Business-managed IT (Kopper et al. 2018a, p. 6)

governance matrix for the distribution of decision rights, similar to the principle of horizontal allocation of decision rights (Winkler and Brown 2014a), or a "hybrid" locus of responsibility (Brown and Magill 1994). Practitioners are also dealing with how to benefit from SIT (Kopper et al. 2017), and IT managers actively enable BMIT under certain conditions (Kopper 2017). Urbach and Ahlemann (2016) have suggested that the development of systems will increasingly happen in interdisciplinary teams in BUs. This is one reason Peppard (2016) has highlighted as to why new conceptualizations of the IT organization are required that also acknowledge activities and decisions occurring outside of traditional functional IT boundaries.

A smaller related theme revolves around *systems/platforms to embrace end-user IT/development* or, more general, IT consumerization. This includes "bring your own device" (BYOD), which allows controlled usage of end-user devices through "mobile device management" (MDM) (Ortbach et al. 2014). Another form in this context is represented by platforms or development technologies that are provided to end-users (Melo et al. 2017). Bygstad (2016) describes innovation with lightweight IT in the form of small innovative apps that are created by users on platform systems, for example, Business Intelligence (BI) platforms (Kretzer and Maedche 2014). Sedera et al. (2016) also found that enterprise system platforms have a significant impact on innovation in organizations.

Although the first theme deals with both risks and opportunities about the phenomenon, existing research has not yet systematically examined the characteristics of its different manifestations. Specifically, no research in the governance and platform themes has yet dealt with the question of which characteristics potentially lead to positive/desirable forms or negative/undesirable forms of SIT/BMIT. The paper at hand closes this gap.

2.2 Phenomenon and terminology

While the three identified themes describe aspects of both ("hidden") SIT and IT activities managed by BUs ("outside the shadow"), they do not sufficiently describe the differences and the transition between them. Kopper et al. (2018b), therefore, introduced the term "Business-managed IT" in addition to "Shadow IT" by outlining and demonstrating a corresponding conceptual framework (Fig. 1).

Previous research dealt with instances of hidden ("covert") SIT that were detected in an organizational IT management context and became subsequently legitimized (Zimmermann et al. 2017). For these cases, the term SIT would not apply even if the instance is still managed by the respective BU as before because the instance is not "in the shadows" anymore. Other IT systems shared all characteristics of SIT but were openly ("overtly") managed by a BU in agreement with the organizational IT management (Kopper 2017).

Several concepts have been discussed in the past in a related context: "Enduser Computing" (Panko and Port 2012) historically describes independent usage of IT systems by end-users, not including the creation of new sophisticated IT artifacts, but of small and simple solutions developed with end-user tools such as spreadsheet applications. Furthermore, its focus is on individual users rather than whole BUs. "Workarounds" (Alter 2014) primarily describe goal-driven adaptations or modification of existing socio-technical systems without changing their overarching architecture (Lund-Jensen et al. 2016). "Decentralized IT" (Winkler and Brown 2014b) represents multiple divisional IT units separate from BUs. BYOD (Köffer et al. 2015) is related to BMIT in the way that usage of individually owned devices is aligned with the IT department, but it does not include all kinds of IT systems.

Furthermore, some alternative terms that are related to the concept of BMIT are used by practitioners. "Citizen IT/development" (Techtarget 2016) is used to describe users of low-code platforms and is limited to this area. "Embedded IT" (Techtarget 2014) deals with the attachment of IT staff members to BUs and does not include non-IT staff. Gartner's (2016) "Business Unit IT" has some resemblance to BMIT but is used inconsistently. "Business-driven IT/innovation" (Györy et al. 2012) might be confusing because all IT should inherently drive business value. This similarly applies to "User-driven IT/innovation" (Fürstenau and Rothe 2014) but with a focus on users rather than BUs. Kopper et al. (2018b), therefore, settle on the term BMIT to accentuate the task responsibility aspect from a governance and managerial perspective.

To summarize, we differentiate between the following terms (Fig. 1):

- *Business-managed IT* describes overt IT systems from an organizational IT management perspective with a high degree of responsibility for IT components and tasks in the BUs.
- Shadow IT describes covert IT systems. It typically exists in BUs with the respective functional responsibility. However, SIT in the IT department is also possible, e.g., when IT employees develop their own covert IT or when they support SIT in BUs by providing interfaces or services not involved in organizational IT management (Chua et al. 2014).
- *IT-managed systems* describe the traditional IT landscape: Overt systems, controlled within the organizational IT management, with a high degree of responsibility for IT components and IT tasks such as planning, engineering or sourcing, testing, documenting, operation, etc. in the IT department.

3 Methodology

3.1 QCA and overall approach

As motivated in the introduction, we chose to use QCA for our study. QCA includes comparative methods and techniques that "enable systematic crosscase comparisons, while at the same time attending to within-case complexity" (Rihoux and Ragin 2008, p. xxiv). QCA is a method to compare conditions across an intermediate number of cases systematically and rigorously and thereby increases external validity in comparison to single case studies (Basurto and Speer 2012). In contrast to standard quantitative analyses, it also considers within-case complexity and allows to derive understandable and applicable configurations (Wendler et al. 2013). QCA, therefore, offers advantages by combining "the intensiveness of case-oriented research strategies and the extensiveness of variable-oriented approaches in a single framework. QCA is specifically designed for a moderate number of cases, too few for variable-oriented research designs and too many for in-depth, case-oriented analysis" (Ragin et al. 2003, p. 323). Assuming that quantitative research usually deals with hundreds or thousands of cases, but sometimes fifty or fewer, and qualitative case-study research usually only with a handful, often only one or two, QCA fills the gap between those thresholds (Ragin et al. 2003). It not only provides some advantages for small- and intermediate-N research designs, but is also a particularly transparent technique because it requires to justify choices made during the whole procedure from a theoretical and/or empirical perspective (Rihoux 2006).

QCA allows examining the effects of conditions (comparable to independent variables) on an outcome (comparable to dependent variables). Most commonly, there is a differentiation between "crisp sets" (csQCA) where variables can only have a value of 0 or 1 (present or absent) and "fuzzy sets" (fsQCA) where variables can have any value between 0 and 1. A difference to classic regression analysis is that configuration theory assumes the existence of more than one combination of conditions that give rise to an outcome (Liu et al. 2015), the so-called principle of equifinality. QCA, therefore, generates single-level models with multiple paths to the outcome variable (Goertz 2006). When using linear, correlationbased models, equifinality is usually overlooked (Schneider and Eggert 2014). This limitation comes from the assumption of symmetric relationships between a set of independent variables and a dependent variable, which is often violated. In symmetric relationships, high/low values of variable X are associated with high/ low values of variable Y and vice versa. This assumes that X is both a necessary and sufficient condition for Y. QCA allows to consider asymmetric relationships where X can be either a sufficient or a necessary condition for Y (Liu et al. 2015).

In addition to assuming symmetric relationships between variables, classic regression models also treat variables as competing in explaining variance in outcomes, showing the net effects of individual independent variables on an outcome. In contrast to that, QCA shows how multiple conditions cooperate or combine to create an outcome (Fiss 2007; Liu et al. 2015). This principle of causal

complexity is called *conjunctural causation*. Sufficient conditions identified with QCA can, therefore, be different combinations of conditions because they individually may not be sufficient on their own (Schneider and Eggert 2014). QCA offers a rigorous and nuanced way to examine the complex ways in which causes combine to create outcomes.

Still, QCA is not meant to be universally "better" than other approaches. It has advantages over regressions when the links are complex (when involving conjunctural causation, asymmetric links, and equifinality) or when the main causes for a certain outcome should be identified. However, it is inferior when trying to determine how much a factor influences the outcome (Schneider and Eggert 2014). QCA also requires that the researcher is familiar with the cases before, during, and after the analytical moment (Schneider and Wagemann 2010). Especially the interpretation of the results gained by QCA is labor-intensive (Liu et al. 2015).

Overall, the number of publications applying QCA has increased substantially in recent years (Schneider and Eggert 2014). QCA is most actively used in business and economics as well as management and organization studies (Wagemann et al. 2016). However, it is also suitable for application in the discipline of Information Systems (IS). Due to the applied character of IS, researchers regularly deal with complex case studies for which systematic comparisons and generalization of findings are often difficult. QCA can help by abstracting and comparing the findings (Wendler et al. 2013). Some IS researchers applied configurational analysis with a focus on organizational performance (Liu et al. 2015). El Sawy et al. (2010) used QCA to better understand the complexity of digital ecodynamics. Liu et al. (2015) applied configurational analysis to IS behavioral research. Park et al. (2017) used a configurational theory approach to explain how IT, organizational, and environmental elements simultaneously combine to produce agility. They also found that for achieving a high level of agility, organizations can choose one of multiple paths that best fits their own contextual condition, instead of following a uniform "herd" industry practice. Despite the recent uptake of this method in the field of IS, it is still not yet broadly applied (Wendler et al. 2013). Liu et al. (2015) have hypothesized that this might be due to IS scholars' unfamiliarity with the approach and due to its rapid evolution.

In the following part, we describe the methodological steps we took for our study. Figure 2 summarizes and depicts all the activities and the specific artifacts resulting from our study along the four phases of a QCA, which are (1) data collection, (2) condition development, (3) calibration, and (4) analysis.

Especially steps (2) and (3) rely on QCA terminology. We, therefore, provide a mapping of our more general terminology used in the introduction and the research questions to QCA terminology: What we have generally termed "characteristics" before is mapped to "measures" and "conditions" where conditions are derived from measures via aggregation. Conditions can be compared to independent variables, whereas outcome corresponds to the dependent variable. What we have termed "combinations of characteristics" are "configurations" in QCA terminology.



Fig. 2 Methodological approach for using QCA with qualitative interview data

3.2 Data collection

3.2.1 Interview guide creation

Schneider and Wagemann (2010) have noted that the conditions and outcome should be selected based on prior theoretical knowledge as well as empirical insights. We, therefore, used a combined deductive and inductive approach in an iterative fashion, where conditions are first derived from existing knowledge and later refined on the basis of case knowledge (Rihoux and Ragin 2008).

The guiding questions for our measures (see Fig. A1 in Online Resource 1) were based on our previous findings from literature (Kopper and Westner 2016a) and hypotheses we derived. Berg-Schlosser and de Meur (2008) have explained that both cases with a "positive" outcome and cases with a "negative" outcome should be included. We, therefore, asked each of our participants—if possible/applicable—to describe one or more positive and one or more negative cases of SIT/BMIT. For each case, we discussed the context, the positive/negative effects for the IT department and the BU, the causing factors, and the preliminary measures on organizational, technical, and general levels.

3.2.2 Semi-structured interviews and transcription

To collect cases, we chose to interview senior IT managers, mostly CIOs. Crilly et al. (2012), for example, took an inductive approach in combination with fuzzy-set

analysis and conducted interviews with senior managers to determine the conditions that shape subsidiaries' stakeholder orientations. We similarly assume that a contemporary CIO not only takes the role of a technical manager but that of an executive-level manager focused on the firm's strategy and processes (Chun and Mooney 2009). CIOs are, therefore, able to provide a holistic view and meet the characteristics of typical "key informants" (Kappelman et al. 2018; Kearns and Lederer 2003). To talk to (mostly) only one key informant in each company is also the most practical way to collect a medium-N number of cases (in comparison to interviewing multiple contacts in each company). As we explicitly wanted to discuss cases with positive as well as negative outcomes, it was important to get access to informants who are aware of several cases to choose from. This not only enhances the identification of cases with a clearly distinct outcome but also allows for better reasoning on why some cases are perceived as positive or negative. Overall, interviewing CIOs enabled us to get a holistic perspective on a company level, and it most likely results in a more balanced selection of cases. However, CIOs/IT managers might show the tendency to be more critical about SIT (Koch et al. 2014). This is a potential bias we needed to keep in mind during the following steps.

To look for study participants, we primarily targeted medium and large companies because they tend to have distinct organizational structures (for example, a separate IT department). Focusing predominantly on German-speaking countries allowed us to reduce linguistic misunderstandings and cultural differences. Potential candidates were approached by mail, professional networking platforms, and through personal recommendations. Overall, 29 interviews with senior or executive IT managers or business managers with a close link to IT were conducted (Table 1). A pilot case interview (Yin 2003, p. 78–80) took place in July 2016 and the remaining ones from October 2016 to June 2017. We sent non-disclosure agreements in advance to build trust from a social and legal perspective. All data and results in this paper are, therefore, anonymized.

By using the previously discussed interview guide and by including open questions, the discussions were semi-structured and allowed participants to speak freely about their experiences and perceptions (Myers and Newman 2007; Yin 2003, p. 89–90). Improvisation and listening strategies were used for detailed follow-up questions and to get a better understanding of the respective case context (Myers and Newman 2007).

In the end, we had 29 participants from a wide variety of industries and company sizes (Table 1). They described 42 different cases of SIT/BMIT. All but one company are located in the EU, and most of them in Germany (D), Austria (A), and Switzerland (CH).

3.3 Condition development (conceptualization)

3.3.1 Open coding of measures

We used a preliminary list of measures (a non-aggregated pre-stage list of conditions) in the form of guiding questions during our interviews (Basurto and Speer Financial

Electrical

Utilities

Health care

Health care

Wholesale

Insurance

Health care

Electronic equipment

equipment

services

equipment

Public sector

P21 CIO

P23 CTO

P24 CIO

P25 CEO

P26 CIO

P27 CIO

P28

P29 CIO

P22

Senior IT mgr.

Senior IT mgr.

> 10.000

> 5000

> 5000

< 500

> 5000

> 20,000

>2000

> 5000

> 5000

> 20.000

Participant	IT users	Industry	ID	Participant	IT users	Industry
CIO	> 20,000	Engineering services	P16	CIO	>100,000	Transportation
CIO	>2000	Electronic equipment	P17	CIO	>20,000	Public sector
Senior IT mgr.	>10,000	Machinery	P18	CIO	< 500	Commercial services
Senior IT mgr.	>40,000	Commercial services	P19	Senior bus. mgr.	>100,000	Telecommunica- tions
CIO and CFO	< 500	Energy	P20	CIO	< 500	Financial ser- vices

< 500

>40,000

> 2000

>40,000

> 500

> 2000

> 5000

> 5000

>100.000

Financial ser-

vices

Insurance

Commercial services

Information technology

Information technology

Health care

Consumer goods

Utilities

Retail

Table 1	Study	participants
---------	-------	--------------

ID

P03

P04

P05

P07 CIO

P08 CIO

P06 CIO

P09 CIO

P10 CIO

P11 DTO

P12 CIO

P13 CIO

P15 CIO

P14 Senior IT mgr.

P01 CIO

P02 CIO

2012). This increased the degree of comparability of the cases (Rihoux 2006). However, we also intended to use a more inductive approach where conditions are primarily selected on the basis of case knowledge (Rihoux and Ragin 2008). We combined this with deductive reasoning to ultimately derive the conditions in a dialog between evidence and theory (Schneider and Wagemann 2010).

We coded the interview transcripts (Corbin and Strauss 2008) using MAX-QDA 12 but considered the preliminary measures from the interview guide as an initial coding scheme. Open coding was used for the identification of additional (sub) categories. A thematic analysis transformed the raw qualitative data into categories of meaning (Forsythe 2012; Haworth-Hoeppner 2000). Two separate researchers reviewed the codes in multiple iterations using random sampling and specifically validated ambiguous cases. In total, 11 measures were identified that could describe commonalities across most cases in the interviews, and that allowed for a systematic comparison (see Sect. 4.1).

Table 2 Mechanism foraggregating values for measures	Measure 1	Measure 2	Condition
to conditions (and the outcome)	0	0	0
	0	n/a	0
	n/a	0	0
	1	1	1
	1	n/a	1
	n/a	1	1
	0	1	Decision 0 or 1
	1	0	Decision 0 or 1

3.3.2 Aggregation of measures to conditions and outcome

The actual number of conditions used for QCA should be kept at a moderate level (Schneider and Wagemann 2010). One of the reasons is that a high number of conditions "individualizes" cases too much and makes it difficult to find regularities across the cases (Berg-Schlosser and de Meur 2008). Another reason is that the issue of "limited diversity" increases with each additional condition (Fiss 2007). In the end, a good balance between the number of conditions and the number of cases must be found. Berg-Schlosser and de Meur (2008) have described that the common practice in an intermediate-N analysis (10-40 cases) is to select from 4 to 6-7 conditions. Wagemann et al. (2016) similarly found in their review that most studies rely on a number of 4 and even up to 8 conditions.

We used the Boolean OR operator to aggregate clusters of measures to conditions (and the outcome variable), as illustrated in Table 2. To make the results more robust and to better support claims that these measures are individually sufficient (Goertz 2006), we did not allow any "conflicting" combinations of values for measures to be aggregated to condition-values. In the case of "conflicts", a context-based decision was made whether the aggregated condition value results in 0 or 1. In our study, there were just a few ambiguous cases that needed to be resolved (see "Calibration" phase).

3.4 Calibration (value assignment)

3.4.1 Definition of anchor points

Calibration in QCA deals with the issue of assigning concrete degrees of membership (values) to the previously defined conditions and outcome (or respectively to the non-aggregated measures first) for each assessed set (case).

To define the metric/scale for a condition/outcome/measure, so-called anchor points must be defined. They represent the main membership thresholds for a set, for example, 1 for full membership and 0 for full non-membership (Basurto and Speer 2012). We defined the anchor points for each of our measures in an active dialog

between data and theory (the same principle previously used for the definition of the measures, conditions, and outcome themselves). We chose csQCA for our study to do so because our interviews were semi-structured with preliminary measures and had an exploratory character. Therefore, the collected data is not detailed enough to justify highly granular fuzzy-set scales for the final measures.

The final measures and aggregated conditions/outcome we used in our study are presented in Sect. 4.1. The outcome variable is called *Positive* and the conditions *Aligned, Local, Simple, Volatile*. We describe them in detail to allow replicable calibration results and support the anchor points with existing theoretical knowledge.

3.4.2 Assignment of measure values with template coding

To calibrate qualitative data for QCA, Basurto and Speer (2012) have suggested to first perform a content analysis and then to summarize the data to qualitative measures for each case. We did so as previously described during phase 2 "Condition development". It is then necessary to choose the degree of precision for the measures and to define their values, including anchor points, as previously described. Those values are subsequently assigned to the measures from the previous step.

We followed a similar procedure but assigned values to the identified measures directly in our qualitative data analysis software using codes instead of summarizing and exporting the data first. This allowed us to retain the visibility of the context during the iteration process. Similar to Haworth-Hoeppner (2000), we coded each case in the qualitative data for the presence (value [1]) or absence (value [0]) of the measures.

3.4.3 Aggregation of measure values to condition and outcome values

Simultaneous to the coding process, we built and maintained the data matrix (Fig. B1 in Online Resource 1) based on our assigned values/codes in a spreadsheet. Each line represents a case and contains its values for the measures in multiple columns. In our study, a case is either a distinct IT system or a general organizational setup that describes characteristics for the same type of IT systems. Due to different factors such as personal communication style of the interviewed participants (fast/slow, concise/expansive) or different circumstances in the respective companies, the number of cases that could be sufficiently provided with values varied from participant to participant. For seven of them it was not possible to extract cases including values for conditions at all (P08, P09, P10, P20, P22, P25, P28). On the other hand, some provided enough depth and breadth to be able to sufficiently derive four cases (P16, P23).

3.5 Analysis

Schneider and Wagemann (2010) have described that two separate analyses should be performed—one for the outcome and one for the negation of the outcome—as each of them can provide complementary insights. The resulting solutions of a

Truth tab	les for Po	ositive an	d ~Positi	ve				
	Cond	itions		Posi	tive	~Pos	itive	# 00000
Aligned	Local	Simple	Volatile	Outcome	Consist.	Outcome	Consist.	# cases
1	1	1	1	1	1	0	0	11
1	1	0	0	1	1	0	0	10
0	0	0	0	0	0	1	1	4
0	0	1	0	0	0	1	1	4
0	1	0	0	0	0,33	1	0,66	3
0	1	1	0	0	0	1	1	3
1	1	1	0	1	1	0	0	3
1	0	1	0	1	1	0	0	2
1	1	0	1	1	1	0	0	1
0	1	1	1	1	1	0	0	1
1	0	0	0					0
0	0	0	1					0
1	0	0	1					0
0	1	0	1					0
0	0	1	1					0
1	0	1	1					0

Fig. 3 Truth table for occurrence (Positive) and non-occurrence (~Positive) of the outcome

QCA provide sufficient conditions for the occurrence of the outcome. However, the sufficient conditions for the non-occurrence of the outcome cannot be directly inferred from those solutions (also see discussion around asymmetric relationships in Sect. 3.1). We, therefore, performed each step described in this phase for the outcome 1 (positive cases) and the outcome 0 (negative cases).

3.5.1 Truth table creation

The first step in the analysis phase is the creation of the truth table from the data matrix. In the truth table, all logically possible configurations are evaluated, and each row corresponds to one configuration. If there are k conditions, there will be 2^k configurations to assess (Liu et al. 2015). To create the truth table, we made use of the software fsQCA 3.0 and followed Ragin (2017). The final data matrix (see Fig. B1 in Online Resource 1) contains six columns: Case (the case name), *Positive* (the outcome variable), and the four aggregated condition variables. For the non-occurrence of the outcome, the negation is assigned (*~Positive*), meaning that all values 0 and 1 are inverted.

The data matrix with 42 cases/rows (4 conditions each) resulted in a truth table with 16 (2^4) possible configurations/rows (Fig. 3). We then determined for each configuration if the outcome is 0 or 1. This is based on the consistency value. A consistency of 1 means that all cases having the same condition values in the current row also have the same outcome of 1 (analogous for 0). If this applies, 0 or 1 can be assigned for the outcome of the configuration unambiguously. If the consistency lies between 0 and 1, not all cases with the current configuration produce the same outcome, and the researcher needs to decide whether to assign 0 or 1. Schneider and Wagemann (2010) have suggested that no consistency values lower than 0.75 should be accepted, but there are no generally accepted valid and exact threshold values. In

our analysis of the occurrence of the outcome (*Positive*), only one configuration has a consistency between 0 and 1. Because its value of 0.33 lies safely below the recommended threshold of 0.75 and after determining that case P03-A (see Fig. B1 in Online Resource 1) is indeed an outlier compared to the other two (P05-A, P18-B), we assigned 0 to the outcome of this configuration. While the same configuration has a higher consistency of 0.66 for the non-occurrence of the outcome (*~Positive*), its value still lies below 0.75 and we therefore assigned 0 as well to the outcome.

The second step before proceeding with the truth table minimization is to decide on the minimum number of cases a configuration needs to have to include it in the further process. After checking the cases for the two configurations that only contain one each (P27-A, P01-A), we determined that they are solid cases for representing the respective configuration, and therefore included all configurations with at least one case (similar to Liu et al. 2015).

However, there are still six configurations left that do not have any case. Therefore, it is not possible to determine from evidence if those produce an outcome of 1 or 0. This issue is called "limited diversity" in QCA. To minimize this issue, a good balance between the number of conditions and the number of cases should be achieved as previously discussed (Berg-Schlosser and de Meur 2008). We opted to at least understand which configurations do not occur in the empirical data and describe/contrast them together with the solutions in the final analysis step (Fiss 2007; Schneider and Wagemann 2010).

3.5.2 Necessary condition analysis (NCA)

Due to the set-theoretic nature of QCA, it allows identifying not only sufficient but also necessary conditions (Ragin 2009). Schneider and Wagemann (2010, 2012) suggest to conduct a necessary condition analysis (NCA) before analyzing the sufficient conditions in order to avoid that necessary conditions remain uncovered ("problem of hidden necessary conditions") or are suggested without actually being existent ("problem of false necessary conditions"). The focus of NCA is on *single conditions* in contrast to the assumptions of "conjunctural causation" in QCA and can complement it meaningfully (Lasrado et al. 2016). Additionally, NCA allows to not only make *statements of necessity in kind but also in degree* (Vis and Dul 2016), meaning the level of the condition that is necessary for the outcome (Lasrado et al. 2016). The latter, however, is only possible when NCA uses linear algebra. When NCA is applied with dichotomous variables, as it is done in this study, boolean algebra is used, which only allows for statements of necessity in kind (Dul et al. 2018; Dul 2019, p. 36–37).

NCA researchers either adopt a deterministic or a probabilistic view on necessity. Although there is some debate on this issue (Goertz 2003), we claim—in-line with Dul (2019)—that both views are legitimate, but that "the probabilistic view may be more realistic and pragmatic" (p. 34). Only in the probabilistic view, one may claim necessity in the presence of exceptions or outliers. This, in turn, requires some agreement on what the number of outliers should be one is willing to accept (e.g., consistency cutoff) and how to verbally declare that necessity is assumed in spite of exceptions. Both aspects are addressed in the following paragraph.

-									
Necessary condit	ion analysis fo	r Positive: A	ligned and L	ocal are alm	iost always i	necessary			
	Positive	2	27	2	27	12	17	16	13
	~Positive	13	0	8	5	6	7	13	0
Conditions		~Aligned	Aligned	~Local	Local	~Simple	Simple	~Volatile	Volatile
	Concistency	0,	93	0,	93	0,	59	0,	45
	Coverage	1,	00	0,	84	0,	71	1,	00
	Positive	27	2	27	2	17	12	13	16
Negation of	~Positive	0	13	5	8	7	6	0	13
negation of		Aligned	~Aligned	Local	~Local	Simple	~Simple	Volatile	~Volatile
conditions	Concistency	0,	07	0,	07	0,	41	0,	55
	Coverage	0,	13	0,	20	0,	67	0,	55
Necessary condit	ion analysis fo	r ~Positive: '	~Aligned and	d ~Volatile a	re always n	ecessary			
	~Positive	13	0	8	5	6	7	13	0
	Positive	2	27	2	27	12	17	16	13
Conditions		~Aligned	Aligned	~Local	Local	~Simple	Simple	~Volatile	Volatile
	Concistency	0,	00	0,	38	0,	54	0,	00
	Coverage	0,	00	0,	16	0,	29	0,	00
	~Positive	0	13	5	8	7	6	0	13
Negation of	Positive	27	2	27	2	17	12	13	16
Negation of		Aligned	~Aligned	Local	~Local	Simple	~Simple	Volatile	~Volatile
conditions	Concistency	1,	00	0,	62	0,	46	1,	00
	Coverage	0	87	0	80	0	33	0	45

mber of cases (Almost) always necessary	Not necessary
---	---------------

Fig. 4 Necessary condition analysis (NCA) results

Figure 4 shows the necessary condition analyses for the occurrence of the outcome (Positive) and the non-occurrence of the outcome (~Positive). For each of them, both the individual conditions and the negation of the individual conditions were tested. For the outcome *Positive*, the conditions *Aligned* and *Local* are almost always necessary (but not necessarily sufficient) because there are almost no contradicting cases. Dul (2015) and Schneider and Eggert (2014) have argued that it might be useful to relax the rule for necessity for outliers and speak of "almost always necessary conditions". The notion of "almost always necessary conditions" has been described by other researchers and is also known as "virtually necessary", "probabilistically necessary", "practically necessary", "usually necessary", and "pragmatic determinism" (Dul 2015). The reason for doing so is to not dismiss valid results in the presence of outliers or exceptions (as it is the case here) or potentially minor measurement errors (Schneider and Eggert 2014). From a deterministic viewpoint, however, one would dismiss a condition as being necessary as soon as there is a single, perhaps even very unlikely exception. In the probabilistic view, however, one needs to decide "depending on the context" (Dul 2015, p. 31) how many exceptions are acceptable. Although there is no general rule (Dul 2015), the consistency cutoff for necessity is very often set at 0.9 (Vis and Dul 2016; Schneider and Wagemann 2012). Also, higher cutoffs are possible, but lower cutoffs are not recommended (Siewert 2017). In our study, where new and emerging phenomena are investigated, we follow the probabilistic view because it allows a more nuanced picture and differentiates between conditions found to be necessary without any exceptions and those with exceptions: for *Aligned*, the exceptions which we will discuss later are P01-A and P03-A and for Local P06-A and P11-A. For the outcome ~Positive, the conditions ~*Aligned* and ~*Volatile* are always necessary, that is without any exceptions. However, as the QCA solutions later show, they are necessary, but only in combination with other conditions sufficient for ~Positive.

Consistency and coverage as calculated by fsQCA 3.0 for each tested condition are included in Fig. 4. For the always necessary conditions the resulting *consistency* value is exactly 1 (no exceptions), and for the almost always necessary conditions the consistency value is 0.93, i.e., close to 1. The *coverage* value in addition indicates to which degree the respective condition is present when the outcome is present. It can, therefore, provide information about the relative importance of the condition.

3.5.3 Truth table minimization

All configurations/rows in the truth table with an outcome value of 1 are already individually sufficient to produce the outcome. However, this does not represent an understandable and applicable set of solutions. QCA allows creating generalized solutions that are logically equivalent by minimizing the truth table. Those solutions are configurations that are supported by a high number of cases and consistently lead to the outcome. fsQCA 3.0 can minimize the truth table in an algorithmic fashion by using the Quine-McClusky method and produces three different types of solutions. The three solution types mainly differ in how they treat the configurations without any cases (see the previous discussion about limited diversity).

The set of "complex solutions" is calculated by taking the logical union of the sufficient configurations (outcome 1) in the truth table and simplifying them using logical operations (Liu et al. 2015). All configurations without cases are simply coded with an outcome of 0 and are, therefore, not considered for the calculation. As the name suggests, the resulting set comprises the longest and most granular solutions. In contrast, the set of "parsimonious solutions" comprises the minimum number of conditions. This set is derived by using information from all configurations for the Boolean minimization without considering a (case) frequency threshold, therefore making strong assumptions and resulting in the smallest and least granular set of solutions. For this, the algorithm simulates all possible outcomes for the configurations without cases and picks the shortest possible overall solutions. The complex solutions all contain at least one of the parsimonious solutions. Finally, the "intermediate solutions" are derived from the complex and parsimonious solutions through counterfactual analysis by using the domain knowledge of the researcher, and the results of the NCA to avoid assumptions that contradict the always necessary conditions in case of perfect necessity (Schneider and Wagemann 2012). Whether configurations may be included that are still in line with the NCA in the sense that an additional exceptional configuration (reducing consistency) is created in the case of almost always necessary conditions that, however, would not lead to a violation of 0.9 consistency cutoff for necessity, is certainly a matter of debate.¹ For that, all configurations in the truth table without

¹ In our data, this is the last line in Fig. 3 (1011), where we assume that a positive outcome [1] is in line with theoretical considerations (see Sect. 4.2.2). Directional expectations, because the presence of volatility contributes to the outcome [1], not its absence, and the fact that the configuration 1010 with outcome [1] is supported with two cases in the data set make the decision to assume a positive outcome for 1011 plausible (so-called easy counterfactual) (Schneider and Wagemann 2012). This results in the omission of ~*Volatile* in the intermediate solution S1-B in comparison to the complex counterpart.

0,85

Difference	due	to	theoretical	assumptions

Fo	cus of	discussio	n
----	--------	-----------	---

X/X Core condition (both in pars. and interm. solutions)

Sufficient con	figurati	ons for outcome Positive [1]					
Solution set type		Configuration	Consis- tency	Raw coverage	Unique coverage	Solution set consistency	Solution set coverage
		Aligned*Local	1,00	0,86	0,38		
Complex		Aligned*Simple*~Volatile	1,00	0,17	0,07	1,00	0,97
		Local*Simple*Volatile	1,00	0,41	0,03		
Demoins en lesse		Aligned	1,00	0,93	0,52	1.00	0.07
Parsimonious		Volatile	1,00	0,45	0,03	1,00	0,97
	S1-A	Aligned*Local	1,00	0,86	0,38		
	S1-B	Aligned*Simple	1,00	0,55	0,07	1,00	0,97
Intermediate	S1-C	Local*Simple*Volatile	1,00	0,41	0,03		
		Assumptions: Aligned (present),	Local (pres	ent), Simple	(present), Vo	platile (present)	
Frequency cuto	off: 1, co	onsistency cutoff: 1.00					
Sufficient cont	figurati	ons for outcome ~Positive [0]					
Solution set		Configuration	Consis- tency	Raw coverage	Unique coverage	Solution set	Solution set
	1		1.00	0.62	0.31	,	J
Complex		~Aligned*Simple*~Volatile	1,00	0,02	0.23	1,00	0,85
		-Aligned Simple ~Volatile	1,00	0,04	0,23		
Parsimonious		Aligned & Circal	1,00	0,62	0,31	1,00	0,85
		~Aligned Simple"~Volatile	1,00	0,54	0,23		

Assumptions: ~Aligned (absent), Frequency cutoff: 1, consistency cutoff: 0.75

Aligned*~Local*~Volatile

Aligned*Simple*~Volatile

S0-A

S0-B

Intermediate

Fig. 5 Complex, parsimonious, and intermediate solutions for outcomes Positive and ~Positive

cases are complemented with an outcome by using theoretical considerations and directional expectations (Buche and Carstensen 2009; Ragin 2009; Schneider and Wagemann 2012).

1.00

1,00

0.62

0,54

~Local (absent), ~Simple (absent),

0.31

0,23

1,00

~Volatile (absent)

Figure 5 shows all *complex, parsimonious, and intermediate solutions* for both the occurrence of the outcome (*Positive*) and the non-occurrence of the outcome (*~Positive*). Buche and Carstensen (2009) are critical of the method of calculating the parsimonious solutions because it works purely methodically and without theoretical reflection. Liu et al. (2015) argue that the parsimonious solutions are usually not presented as the final solutions of the QCA, but they are necessary for calculating the intermediate solutions.

The parsimonious solutions also serve as an input for differentiating between *core and peripheral conditions* in the intermediate solutions, as introduced by Fiss (2011). Core conditions (marked as bold font in Fig. 5) are, therefore, those that

exist both in the parsimonious and in the intermediate solutions. Peripheral conditions are those that exist only in the intermediate solutions (Liu et al. 2015). Core conditions have a strong causal relationship with the outcome (Park et al. 2017) because both their presence and absence are supported by the data. For peripheral conditions, only their presence is supported, but it is unclear whether the countercondition is valid due to a lack of relevant data. However, despite having a relatively weaker relationship with the outcome, peripheral conditions complement core conditions and can serve as necessary conditions in the configuration (Liu et al. 2015).

Figure 5 also shows two metrics calculated for each solution by fsQCA 3.0: Consistency and coverage. The *consistency* values of each individual solution and of the overall solution sets are derived from the consistency values in the underlying truth table (Fig. 3). Because all configurations with an outcome of 1 in the truth table for *Positive* and *~Positive* have a consistency of 1.00 (all underlying cases for each of those configurations produce the same outcome of 1 and do not contradict each other), the consistency for all resulting solutions is 1.00 as well.

The *coverage* values indicate the relative importance of each path towards the outcome (Fiss 2007). In other words, coverage explains to what degree the outcome is covered or explained by a particular solution term or set of solutions (Schneider and Wagemann 2010) and can be compared to effect size in statistical hypothesis testing (Liu et al. 2015). Practically speaking, *raw coverage* indicates the percentage of observed cases (which lead to the outcome) that can be explained by the respective solution term (or set). Taking, for example, S1-A, for 25 of all 28 cases with an outcome of 1 (86%) *Aligned* and *Local* has a value of 1 and the raw coverage is therefore 0.86. *Unique coverage* indicates to which degree the outcome can only be explained by a specific configuration and by none of the others in the solution set. Because it is easier to calculate for a solution set with only two configurations, we use the parsimonious solution set for *Positive* for illustrative purposes. The unique coverage of the configuration *Aligned* (0.52) is determined by the coverage of the whole solution set (0.97) minus the raw coverage of the second configuration *Vola-tile* in the set (0.45).

During minimization of the truth table with fsQCA 3.0, two prime implicants (terms cannot be reduced any further) were tied and a choice was necessary (one or more prime implicants were logically redundant): ~*Local**~*Simple* and ~*Aligned**~*Local*. The latter was chosen to proceed with the calculation due to higher solution coverage. Though, ~*Local**~*Simple* would have been valid (in addition) as well due to theoretical considerations (as described in the theoretical assumptions about contributions of each condition to the solution). The choice of selecting the prime implicants only affects the parsimonious solution in this case, not the intermediate solution.

When comparing the results of the NCA with the results of the QCA, no contradictions are found for the outcome *~Positive*. Both produce the same necessary conditions, as *~Aligned* and *~Volatile* and only those conditions occur in all sufficient configurations. For the outcome *Positive* the QCA does not produce any additional necessary conditions (no false necessary conditions) but does not confirm the NCA, as *Aligned* and *Local* do not occur in every sufficient configuration. This problem of hidden necessary conditions may occur in cases of limited diversity while calculating the parsimonious solution as simulated configurations without cases may be used that contradict the NCA results. Both conditions were almost always necessary, as we accepted each of them in spite of two exceptions. Instead of adding the almost always necessary conditions to each of the sufficient configurations after conducting the QCA (Ragin 2009), we opt for keeping the NCA and QCA results separate, as the QCA consistency and coverage values correspond with the solutions found by the QCA without these additions (Schneider and Wagemann 2012). Separating these results also allows to specifically account for the fact that the conditions derived from the NCA for the outcome *Positive* were only almost always necessary.

3.5.4 Visualization

Schneider and Wagemann (2010) have suggested that different presentational forms should be used to depict not only the variable-oriented, but also the case-oriented aspects of QCA. For the former, we use the "quasi standard" notation system for QCA results from Ragin (2009) in Fig. 6 which, for example, was also used by Park et al. (2017). The goal for the latter is to link the cases back to the solution formulas by using graphical representation. In the end, the QCA results should enable a better understanding of the original cases (Schneider and Wagemann 2010). To achieve this, we use a Venn diagram in Fig. 7, as similarly applied by Rihoux and Ragin (2008) and Rantala and Hellström (2001). Both figures are depicted in the following section to be able to show them together with a detailed description of the results.

4 Results

4.1 Outcome (RQ1) and conditions (RQ2)

As previously described, we derived the outcome and condition variables in an active dialog between empirical data and theory. Each of them includes multiple measures to help concretize them. The values of the measures are aggregated to the outcome and the condition variables. By describing the outcome and conditions we also answer RQ1 and RQ2.

4.1.1 Positive (case)

The outcome variable *Positive* is determined by the two measures *Distinct* and *General*. A case can, therefore, represent either a distinct IT system managed by a BU (covertly in the form of SIT or overtly in the form of BMIT) or a general organizational setup which describes characteristics for the same type of IT system.

4.1.1.1 Distinct (positive case) A distinct case in the context of this study is understood as a concrete manifestation of SIT/BMIT (as defined in Sect. 2.2).

competitive environments (Kretzer and Maedche 2014).

The object of interest is, therefore, a distinct artifact or specifically an IT system with a high degree of responsibility for its components and tasks in a BU. This can include both overt (BMIT) and covert (SIT) instances. The calibration of this measure, that is, whether it is a positive case [1], is primarily based on the assessment of the interviewed participant (typically the CIO) who serves as a "key informant" (Kappelman et al. 2018; Kearns and Lederer 2003) and who is able to oversee the whole firm's strategy and processes (Chun and Mooney 2009). In practice, it means that a case is coded *Distinct* [1] if the observed positive consequences (Kopper and Westner 2016a) outweigh the negative ones. This is the case if the participant predominantly associates positive attributes with the instance such as innovativeness (Behrens 2009; Rentrop and Zimmermann 2012), a good fit with users' requirements (Behrens and Sedera 2004), efficiency gains (Röder et al. 2014), or the creation of a new revenue stream (Ferneley 2007). Other possibilities include increased flexibility and the ability to adapt in uncertain and

On the other hand, the measure is coded *Distinct [0]* if the participant primarily associates negative consequences with a case. This, for example, includes failed or problematic projects that led to financial losses or inefficiencies. Inefficiencies can also arise due to loss of synergies, loss of scale effects, or redundant systems (Györy et al. 2012; Kretzer and Maedche 2014). Other negative attributes can be associated with a lack of overall system quality, such as performance issues, operational issues, poor integration with other systems (Hetzenecker et al. 2012), or data security and data consistency risks (Györy et al. 2012; Kretzer and Maedche 2014; Silic and Back 2014).

4.1.1.2 General (positive case) In contrast to distinct cases, *General* cases describe organizational settings that represent multiple instances of SIT/BMIT sharing the same characteristics. This includes settings with IT governance models where BUs are involved to a significant degree, specifically federal models, as defined by Weill and Ross (2004). Such a model can, for example, define patterns in which infrastructure decisions are centralized, and business application decisions are made by the BUs (Andriole 2015; Winkler and Brown 2014b).

Similar to *Distinct* the value for *General* is primarily based on the assessment of the participant. A case is coded *General* [1] if it is predominantly associated with positive attributes. This includes settings that, for example, actively encourage joint initiatives between the IT department and the BUs that create new benefits for the entire organization (Silic et al. 2016). Specifically, this can mean a possibility for low-cost innovation and the ability to respond to changes in business conditions rapidly (Tambo and Bækgaard 2013). Altogether, the same positive and negative consequences already described for *Distinct* were used as indicators to code *General* [1] or [0] with the difference that they apply to multiple instances in a general setting.

4.1.2 Aligned

The condition *Aligned* is determined by the measures *Approved*, *Organizational split*, and *Technical split*. Generally, it means that there is alignment between the IT plan and the BU plan (Kearns and Lederer 2003) and, more specifically, that instances in the observed cases are approved and aligned with the organization's IT strategy and standards (Chua and Storey 2016; Ferneley 2007). This can be achieved by involving both the IT department and the BUs in an organizational task split or by employing standardized infrastructure in a technical split. *Organizational split* [1] and *Technical split* [1], therefore, strongly correlate with *Approved* [1] in practice.

Ultimately, *Aligned* describes overt in contrast to covert cases (Ferneley 2007; Haag and Eckhardt 2017). Cases with a value of *Aligned [1]*, therefore, resemble BMIT and cases with ~*Aligned [0]* resemble SIT (Kopper et al. 2018b). Because the management practices described for the measures *Aligned*, *Organizational split*, and *Technical split* aim to mitigate the risks of negative consequences for SIT/BMIT, we expect the condition *Aligned [1]* to contribute to the outcome *Positive [1]*.

4.1.2.1 Approved The measure *Approved* describes if there is overarching consent about the legitimacy of distinct instances of systems managed by BUs or general IT task responsibilities on an organizational level (coded *Approved [1]*). This can have varying degrees of formality, for example, formal policies for SIT/BMIT or a clear definition of split IT task responsibilities. On a more general level, it specifies if the case is complying or not complying with management intentions (Alter 2014). Chua et al. (2014), for example, determined in their cases if central IT was consulted and if the instances were legitimized by the organization and the IT department. Zimmermann et al. (2016a) similarly have described requirements to consult the IT department for BMIT and the importance of maintaining transparency. A prerequisite for approval might also include the need to adhere to technological (for example, infrastructure), data (for example, data structures), and process (for example, security) standards (Dittes et al. 2015).

A case is coded *Approved [0]* if there is no organizational approval for the observed system or setting (Köffer et al. 2014). Such systems exist outside predefined structures that regulate and control IT work (Behrens 2009). This similarly covers un-enacted projects that have never been subject to an official evaluation process and exist without awareness by top management (Blichfeldt and Eskerod 2008; Buchwald et al. 2014b; Buchwald and Urbach 2012). Most obviously, a case is not approved if policies exist that would clearly define IT task responsibilities, but they are deliberately ignored or violated (Haag and Eckhardt 2017).

4.1.2.2 Organizational split *Organizational split* [1] describes settings for which IT task responsibilities are split between the IT department and the BUs (Zimmermann et al. 2016a). As previously mentioned for the measure *General*, this is especially the case for a federated IT governance model (Andriole 2015; Weill and Ross 2004). In such a model it makes sense for the IT department to act as a repository of organizational-level requirements and coordinate them between BUs. This includes making

sure that systems remain aligned with the overall enterprise vision and architecture. It can also play an active role by guiding BUs toward effective implementation, operational, and procurement practices. Specifically, it can structure the implementation or vendor-selection process because it tends to be more aware of the broader implementation issues (for example, proper testing) and has more experience selecting and negotiating with vendors. Other critical areas where the IT department may take an active role or at least educate the BUs about are security, privacy, regulatory, or continuity (service operation) issues (Chua and Storey 2016).

Organizational split is simply coded [0] if the case does not involve split IT task responsibilities at all, and the BU is acting solely on its own without even making use of the experience of the IT department or asking for advice.

4.1.2.3 Technical split Organizational split indicates an active involvement of the IT department but Technical split [1] means that it is only passively involved by providing infrastructure or tools on a technical level (Andriole 2015). While BUs concentrate on requirements, rapid application development, and deployment, IT departments are responsible for the "transportation architecture" or infrastructure (networks, servers, databases, platforms) (Brown and Magill 1994). On the most abstract level, platform systems are provided as a layer that remains stable, while the components built on top vary over time (Bygstad 2016). Platforms (and lower layers such as databases and servers) enforce uniformity and allow to exploit knowledge that is embedded in the technology. This enables increased agility and fosters self-organizing, loosely coupled teams (Krancher and Luther 2015). Krancher and Luther (2015) have described two different abstraction levels for platforms where deployment PaaS (dPaaS) aims at software developers, and model-driven PaaS (mPaaS) aims at technology-savvy end users. The latter are also known as low-code platforms and allow to build applications by simply modifying metadata, without necessarily having to write code.

Technical split is coded [0] if BUs use their own infrastructure not provided by the IT department, which can also be independently sourced from public cloud providers. This is, for example, the case with shadow sourcing, as described by Haag (2015).

4.1.3 Local

The condition *Local* is determined by the measures *Specific* and *Standalone* and therefore determines if a system is locally encapsulated in a BU from a functional or integration perspective. Theory suggests that highly specific systems are more efficiently managed by BUs (Zimmermann et al. 2016a) and that strongly embedded/integrated systems (Fürstenau and Rothe 2014) require a more centralized approach (Bygstad 2016). We, therefore, expect *Local* [1] to contribute to the outcome *Positive* [1].

4.1.3.1 Specific A case is *Specific* [1] if the instance(s) it describes are very specific to a BU and its processes. Application specificity also means a higher level

of customization to the individual BU's needs (Behrens and Sedera 2004; Winkler and Brown 2014a). Zimmermann et al. (2017) found that when specificity is high, it is more efficient and faster to coordinate IT tasks within the BU because this omits costly and complex assignment procedures with the IT department (transaction costs). In contrast, non-specific tasks should be handled by the IT department because it, for example, can also consider reusability aspects or scale effects (Zimmermann et al. 2016a). A second aspect of the measure *Specific* is "scope of use" that defines the breadth to which an application is used within an organization, that is, if it is (or can potentially be) used only in one or multiple BUs (Fürstenau et al. 2016). Similar to specificity, Winkler and Brown (2014a) found that applications with a greater scope should be governed by the IT department due to the achievable scale effects.

Personalized spreadsheet-based applications would be an example for *Specific* [1] cases because they are specifically tailored by their creators for their needs and only cover a narrow scope of use (Fürstenau et al. 2016). They also allow flexibility to coordinate organizational and local requirements (Paulsson and Johansson 2013). However, also larger systems can be *Specific* [1] if they are limited to the requirements of a single BU. This, for example, applies to systems that are created for or offered to customers that are only served by one BU or systems that support the production processes of a single BU. In contrast to that, ERP systems, CRM systems, or more generally formal enterprise systems are not specific to a single BU and typically have a high scope of use (Huber et al. 2016). Cases that deal with such core systems are therefore coded with *Specific* [0].

4.1.3.2 Standalone *Standalone* refers to the degree of technical integration of a system. Bygstad (2016), for example, describes "Heavyweight IT" which represents fully integrated solutions that interact with the business logic and data access layers. In contrast to that, "Lightweight IT" describes non-invasive solutions that are usually initiated by users in cooperation with an external vendor. A lack of integration is commonly identified as one of the risks of SIT (Chua et al. 2014; Huber et al. 2016) which is caused by insufficient skills of conceptual IT development (Hetzenecker et al. 2012). Consequently, data integrity issues can arise (Silic and Back 2014). Fürstenau and Rothe (2014) therefore used "architectural embeddedness" to determine the criticality of a system. We similarly code *Standalone [0]* for systems with a high architectural embeddedness and vice versa.

Practically speaking, Heavyweight IT, or generally core backend solutions such as ERP systems, database servers, or integration software (Bygstad 2016) are coded *Standalone [0]*. Because, for example, ERP systems integrate many different organizational functions (Behrens and Sedera 2004) and have increased requirements to security and resilience, the argument for a centralized ownership by the IT department gets stronger (Bygstad 2016). Most applications realistically require at least a minimum degree of integration. Cases are therefore coded *Standalone [1]* if the respective systems are either completely isolated (for example, certain proof of concepts, product IT, or spreadsheet-based applications) or have a low degree of integration with other systems through clearly defined interfaces or APIs.

4.1.4 Simple

The condition *Simple* is described by the measures *Low-complexity* and *Low-risk*. Both are related, but *Low-complexity* puts a stronger emphasis on the size of a system and *Low-risk* on the criticality of the domain a system is built for. Due to the following theoretical fundaments, we expect *Simple [1]* to contribute to the outcome *Positive [1]*.

4.1.4.1 Low-complexity To determine the value for *Low-complexity*, we primarily rely on the criteria for "size of SIT" as outlined by Rentrop and Zimmermann (2012). One of their sub-criteria is how professionally a system is operated and how qualified employees are for IT tasks. The level of complexity is, therefore, based on the ability to handle systems with an increasing size. Other criteria include the number of users, the type of components (hardware, software, or a combination of components), and the maturity of service processes around the system. Additional criteria are drawn from Chua et al. (2014), who have described "Lightweight IT" as simple applications on cheap technology. Examples include mobile apps or personal productivity tools such as self-developed spreadsheets (Fürstenau et al. 2016). Those are typically applications that are "on the edge" and do not cover core functions (Chua et al. 2014). Workarounds (or workaround systems) are also considered as small-scale actions with a low complexity (Huuskonen and Vakkari 2013) that can be handled by BUs themselves.

More complex systems (*Low-complexity* [0]) with a broad scope, however, require more involvement of the IT department because higher risks and costs are expected (Zimmermann et al. 2017). Required resources (cost and time) are therefore also an indicator, but it needs to be evaluated in the respective context. Chua et al. (2014), for example, have described a case where 50–70 thousand dollar is considered a "significant" amount, and we similarly consider cases to be complex if costs exceed 200 thousand dollars or even millions. Project runtimes beyond one year can also indicate high complexity. ERP systems, for example, are considered to have a high complexity (Behrens and Sedera 2004; Jones et al. 2004) as well as other backend and transaction systems such as database servers or production control systems. A higher degree of integration is also associated with higher complexity (Bygstad 2016). We consider product development (R&D) to be generally complex as well in comparison to the examples described for *Low-complexity* [1].

4.1.4.2 Low-risk As for *Low-complexity*, we similarly draw on Rentrop and Zimmermann (2012) to determine the value for *Low-risk*. They used multiple sub-criteria to understand the relevance of a system. One of them is the strategic relevance for the company and another the criticality of the system. They have described that higher criticality also increases the risks and effects of erratic behaviors in areas such as IT security and compliance (Zimmermann et al. 2016a). Chua et al. (2014) have described examples of accepted SIT instances that do not cover critical areas, that is, they do not affect the core, regulated manufacturing business. Another

example has been mentioned in form of spreadsheets that are indeed used in a highly regulated healthcare environment, but only for non-critical tasks that do not involve patient data. Zimmermann et al. (2016a) have argued that for systems with an acceptable risk level, it is justified to keep certain IT tasks in the BUs. Keeping critical tasks in the IT department can reduce the risk if, for example, professional software testing is performed or increased operational stability can be expected due to the usage of standardized infrastructure.

As previously mentioned, some of the most-cited risks of SIT are related to privacy, security, and compliance which often must be resolved by the IT department (Chua et al. 2014; Fürstenau et al. 2017). The risk value is therefore high (*Low-risk* [0]), especially, when the system is affected by regulatory requirements (Gozman and Willcocks 2015) such as in the healthcare sector (Chua et al. 2014) or the banking industry (Fürstenau et al. 2017). Another risk is business continuity which can lead to disruption of operations in the event of a fault (Chua and Storey 2016; Györy et al. 2012). The risk is therefore considered high if a system covers highly critical areas with high availability requirements. Zimmermann et al. (2017) have, therefore, argued that high-risk tasks should be allocated to the IT department. Risk can also be high for a project if there is a mismatch between the complexity and the skills of the parties involved.

4.1.5 Volatile

The condition *Volatile* comprises the measures *Provisional* and *Ambiguous*. It describes cases with provisional, temporary systems and systems that have an exploratory character. Both measures can co-occur in experimental projects with short development cycles which also result in short product life cycles (Bygstad 2016).

4.1.5.1 Provisional *Provisional* is probably the most intuitive measure which indicates the temporality of a system. Workarounds or workaround systems are, for example, considered to be temporary (*Provisional* [1]) solutions until the obstacle disappears or a more complete fix for the system they cover is available (Alter 2014). Another example is prototypes that are not intended for long-term operation. Ebeling et al. (2013), for example, have described the decentral, iterative development of a shadow system to determine the requirements for a future core system. However, they have also recognized the system as a temporary solution until a proper, integrated system is available. Realistically, no system is built to last forever, but if the intention from the beginning is a limited foreseeable time of operation (for example, a single-use system for a project or a campaign that only lasts six months), the respective case is coded *Provisional* [1].

In all other cases, systems are not provisional (*Provisional* [0]). This is, for example, the case for systems that cover stable BU processes, for IT products developed for customers, or for core systems such as ERP or CRM. For those non-temporary systems, there are higher requirements on architectural standards, integration, and stability to ensure efficient long-term operation. They should, therefore, be in the responsibility of the IT department (Bygstad 2016). However, applying the logic of

transaction costs, it would be too expensive for temporary solutions to adhere to the same standards because the investment cannot pay off in the long term (Zimmermann et al. 2017). They are, therefore, candidates for being managed by the BUs themselves.

4.1.5.2 Ambiguous Cases are *Ambiguous* [1] if they deal with uncertainty, that is, systems for which the requirements are initially unclear (Zimmermann et al. 2016a). Similar to the concept of Bimodal IT (Horlach et al. 2017) or Lightweight IT (Byg-stad 2016), this describes experimental and agile environments with short development cycles and possibly short product life cycles. It especially covers exploratory projects that may have intangible and difficult to quantify benefits (Chua and Storey 2016). Zimmermann et al. (2017) have described that BUs engage in IT development to try new ideas that have unclear requirements. It can be justified to have responsibility for those instances in the BUs because involving the IT department would require extensive and inflexible coordination efforts that slow down the process. IT departments also may not want to spend resources on a properly designed and integrated system until the BU is certain of its future usage (Zimmermann et al. 2016b).

However, systems for which the above descriptions do not apply are *Ambiguous* [0]. This especially applies to systems that are focused on stability and efficiency, for example, backend and transactional systems (Bygstad 2016; Horlach et al. 2017) or more generally systems with clear requirements.

4.2 Solutions (RQ3)

Figure 6 shows the results of the QCA using the notation system from Ragin (2009). The sufficient configurations for the outcome *Positive [1]* and *~Positive [0]*, i.e., not positive, are based on the intermediate set of solutions. Each rectangle (e.g., S1-A) depicts a single solution/configuration in the respective solution set. Large, dark circles represent the presence of core conditions (which are also included in the parsimonious solution) and small, dark circles represent peripheral conditions. Crossed-out circles indicate the absence of a condition and blank spaces "do not care" situations, i.e., that it does not matter for the outcome if the condition is present or absent. Consistency, raw coverage, and unique coverage are also given for each configuration and overall solution consistency/coverage for the two solution sets *Positive*.

Figure 7 shows a Venn diagram with all cases in the respective intersections of conditions. Each box represents a condition being present [1]. If a case is outside of a box, the condition for it is absent [0]. The diagram also shows for each case if it is *Positive* or ~*Positive*. Overlaying all the different solutions allows linking the results back to the cases.

The following subsections are organized along the four sufficient configurations from Fig. 6 and within each subsection along the quadrants from Fig. 7. A quadrant is identified by the configuration it represents using standard notation conventions (*: logical AND; ~: negation) and the abbreviated format used in Fig. 7 represented by the truth values of the four conditions in the order Aligned, Local, Simple,



Fig. 6 Sufficient configurations for outcomes Positive and ~Positive based on intermediate solution



Fig. 7 Venn diagram of intermediate solution configurations and underlying cases

Volatile. Example: 1010 represents the configuration Aligned*~Local*Simple*~Vo latile, i.e., cases that are "aligned" and "simple" as extensively described above but not "local" and not "volatile". The 1010 quadrant can be found in the upper middle of Fig. 7. It includes 2 cases (P06-A, P11-A).

The solution configurations are presented together with the respective cases in a narrative and descriptive manner. Readers who prefer a more tabular format are referred to Table 3.

4.2.1 S1-A) Aligned*Local (Positive)

S1-A has the highest raw coverage (0.86) and unique coverage (0.38) of all configurations in the solution set for *Positive*. We, therefore, recognize it as the most important one with the most explanatory power, that is, it can explain 86% of all *Positive* outcomes and 38% of them uniquely (as they are not explained by any of the other configurations in the solution set). This is also supported by the results of the NCA where both *Aligned* and *Local* turn out to be "almost always necessary" for the outcome (with a consistency of 0.93) even if *Local* did not result in a core condition in the QCA process. To illustrate which cases are covered by the solution we structure them along all the possible configurations. Every one of those configurations is shown as a quadrant in the Venn diagram in Fig. 7. S1-A (Aligned*Local) contains (values for) two conditions which means the value of the two other conditions do not matter for this solution (they can either be [1] or [0]). As a result, S1-A covers four (2^2) different configurations which are also represented in the diagram as the four quadrants <u>1100</u>, <u>1101</u>, <u>1110</u>, and <u>1111</u>. The fixed condition values given by the solution are underlined for each configuration.

4.2.1.1 <u>11</u>11 (Aligned*Local*Simple*Volatile): 11 cases This configuration where all conditions have a value of [1] is not only part of S1-A, but also S1-B and S1-C. Furthermore, it is backed by the highest number of cases. This is consistent with the theoretical foundations previously described which indicate that all four conditions contribute to the outcome *Positive*.

Four of the cases are dealing with *general* governance settings. In *P07-A* for example, BUs are allowed to build their own solutions as long as their initiatives are approved in the architecture board. Approval is given if the system is not touching any core processes, only affecting isolated areas (involved applications and infrastructure are very local), and if the requirements are still unclear. For such cases the participant recognized cost benefits because they require an agile approach with a low process and coordination overhead. For *P01-C* the same conditions apply, but the participant more critically mentioned that BMIT is inevitable and therefore rules and conditions must be defined under which it is acceptable. For *P13-B* and *P15-C* the conditions were described in a similar form as the other two.

Another category is represented by local *Proof of Concept (PoC) solutions*. In *P16-C* (similarly also *P23-C*), PoCs are created locally and (if successful and if required) properly rebuilt afterwards centrally to be able to scale the solutions. Those initiatives are embedded in a central governance with clear responsibilities. Security requirements still need to be considered (for example, that central infrastructure

		ů	0,0	0		
Outcome	Sufficient configura- tions	Quadrants	Description (fixed condition values for respective solution are underlined)	Number of cases	Case numbers	Categories of cases (c.f. Fig. 8)
Positive	S1-A	1111	Aligned*Local*Simple*Volatile	11	P01-C, P07-A, P13-B, P15-C	General governance settings
					P16-C, P23-C, P29-A	Proof of Concept (PoC) solutions
					P05-B, P16-A, P26-B	Reporting
					P12-A	Other
		1100	Aligned*Local*~Simple*~Volatile	10	P02-A, P02-B, P15-A, P19- A, P23-A, P24-A	Product IT/R&D
					P23-B, P23-D	Larger local business systems
					P04-A, P17-A	General governance settings
		1110	<u>Aligned*Local</u> *Simple*~Volatile	3	P06-C, P16-D, P21-A	Smaller local business systems
		1101	<u>Aligned*Local</u> *~Simple*Volatile	1	P27-A	New ventures
	S1-B	1111	<u>Aligned</u> *Local* <u>Simple</u> *Volatile	11	P01-C, P07-A, P13-B, P15-C	General governance settings
					P16-C, P23-C, P29-A	Proof of Concept (PoC) solutions
					P05-B, P16-A, P26-B	Reporting
					P12-A	Other
		1110	<u>Aligned</u> *Local* <u>Simple</u> *~Volatile	3	P06-C, P16-D, P21-A	Smaller local business systems
		1010	<u>Aligned</u> *~Local* <u>Simple</u> *~Volatile	2	P06-A, P11-A	Cloud-based CRM workflows
		1011	<u>Aligned</u> *∼Local* <u>Simple</u> *Volatile	0	n/a	n/a
	S1-C	1111	Aligned*Local*Simple*Volatile	11	P01-C, P07-A, P13-B, P15-C	General governance settings
					P16-C, P23-C, P29-A	Proof of Concept (PoC) solutions
					P05-B, P16-A, P26-B	Reporting
					P12-A	Other
		0111	~Aligned*Local*Simple*Volatile	1	P01-A	Simple, temporary cloud solutions

Table 3 Intermediate solution configurations and underlying cases with case category descriptions

ble 3 (c	continued)					
utcome	Sufficient configura- tions	Quadrants	Description (fixed condition values for respective solution are underlined)	Number of cases	Case numbers	Categories of cases (c.f. Fig. 8)
Positive	S0-A	0000	~Aligned*~Local*~Simple*~Volatile	4	P01-B, P06-B, P12-B, P15-B	Critical core systems
		<u>0010</u>	~ <u>Aligned</u> *~ <u>Local</u> *Simple*~ <u>Volatile</u>	4	P03-B, P03-C, P13-A, P14-A	Unofficial, non-specific, small, permanent solutions
	S0-B	<u>0010</u>	~Aligned*~Local*Simple*~Volatile	4	P03-B, P03-C, P13-A, P14-A	Unofficial, non-specific, small, permanent solutions
		0110	~Aligned*Local*Simple*~Volatile	.0	P16-B, P18-A, P26-A	Unofficial, local, small, and permanent solutions
	n/a	<u>0100</u>	~Aligned*Local*~Simple~ <u>Volatile</u>	ε	P05, P18-B	Unofficial, local, complex or high-risk, permanent applications
					P03-A	Outlier

needs to be used), but fewer restrictions exist for the development itself because the teams should be able to move fast and to explore technical options. PoCs are also less critical because often it is not clear if there is a business case for the respective area, and they remain temporary solutions. Still, there are basic documentation, and support/continuity requirements, and the IT department supports with vendor selection. In *P29-A*, business-managed PoCs are also seen as a way for agile exploration, which makes them more efficient than formal projects with large process overhead. Know-how is created by experimenting with simple tools and, for example, by building spreadsheet-based prototypes that are at some point rebuilt together with the IT department when they reach a critical size. However, even before that, the IT department is available to support with complex tasks.

Reporting related cases form the third category of this configuration. In *P05-B* certain users are given read-only access to a central database. They can write SQL queries themselves to be able to quickly adapt to their changing local requirements and use self-developed spreadsheet-based tools to analyze/process the data and build reports. If an interface is required to transfer data back into the database, the IT department needs to be involved to retain data consistency. *P16-A* similarly provides a possibility for self-service reporting/analysis and allows advanced data analysts to build their own reports using a web-based Business Intelligence suite. *P26-B* was planning a data warehouse (DWH) self-service platform at the time of the interview.

A rather unusual case was described in *P12-A* where the IT department hired an external office automation expert that worked directly with the BUs to build local solutions together. The expert facilitated to efficiently cover local, initially unclear requirements that would otherwise have low priority in large system developments.

4.2.1.2 <u>1100(Aligned*Local</u>*~Simple*~Volatile): 10 cases The high number of cases for this configuration shows that, as the solution highlights, it is sufficient for *Aligned* and *Local* to be true for the outcome *Positive*, but not necessarily *Simple* and *Volatile*. More complex, stable instances can, therefore, also lead to a positive outcome as long as they are *Aligned* and *Local*.

The largest category of cases for this configuration is represented by product IT or more general R&D. The respective participants explained a general split of IT tasks and responsibilities, such as for P02-A. They described so-called "commercial IT" that is centrally governed, controlled, and supported by the IT department. This includes general infrastructure (e.g., workstations, network) and transactional or core systems (e.g., ERP, business warehouse). From a process perspective, the IT department is also responsible for central coordination of material management and production planning. However, so-called "shop floor IT" (management of CNC machines, systems at the production lines, and grapplers) is managed in the BUs. They are also responsible for "product IT", which is the development and maintenance of IT products or software in tangible products for customers. The participant also explained a specific project in P02-B for the development of a customer IT system where the IT department "would have only been an obstacle if it were involved". In P23-A, the participant similarly differentiated between "product IT" (including workflows in production facilities, production control, etc.) which is managed in the BU and "business IT" (ERP, DWH, etc.) which is managed by

the central IT department. The interfaces and responsibilities between the two are clearly defined. In *P19-A*, the participant emphasized that the task split should be commonly agreed to and documented but recognized that it is difficult to draw the line. They described "commodity systems" such as infrastructure (up to the integration layer), finance applications, or HR systems that should be run centrally by the IT department. In contrast to that, customer services, or generally systems that can provide a competitive advantage (which are a differentiation factor) can be managed by BUs. One reason for that is that usually no efficiencies or economies of scale can be gained by consolidating such systems centrally. In *P24-A*, a mostly centralized setup is described but R&D for IT products is still done in a separate unit. In *P15-A*, a product IT project is described where the BU and the IT department are working closely together.

Another category for this configuration is represented by *larger local business systems*. In *P23-B*, flexible frameworks are centrally provided in an "Application Service Provider Model" that can be customized by ecommerce teams for local country-specific requirements. In another example (*P23-D*) a BU created a new system for a locally specific customer process. The frontend was developed locally and with the help of the IT department integrated into the standard, centralized fulfillment backend through an interface. The BU also worked together with the central IT security officer to fulfill the required standards. The participant highlighted that the BU was able to move very fast and successful in this setting and that their IT department would not have had the resources or priorities to fulfill this requirement.

Two cases describe general governance settings with a split IT responsibility model. In *P04-A*, BMIT is "business as usual" because the company is structured rather decentral with hundreds of organizational units. BUs are responsible for product IT and locally specific systems. The central IT department provides shared IT services and is responsible for workplace IT, infrastructure (until the database layer in the technology stack), and overarching applications. The participant for *P17-A* similarly differentiated between "cross-section procedures" for which their department is responsible and "specialized procedures" for which the individual units are responsible. The latter are required to use the IT department's IT infrastructure until the database layer for their applications. The IT department for example also offers a voting platform that can be customized by local regions.

4.2.1.3 <u>11</u>10 (Aligned*Local*Simple*~Volatile): 3 cases This configuration is also part of S1-B and is represented by cases dealing with *smaller local business systems*. *P06-C* describes a small pricing engine that was developed and is maintained by some traders in the BU of a commercial banking company. They are working with an external vendor and are in close alignment with the IT department. Additionally, they adhere to commonly agreed processes, use development tools and secure infrastructure provided by the IT department, and work together with IT for testing and go-live. In *P21-A*, BUs are responsible for specialized, specific, small (non-core) tools for which they have the required knowledge and IT skills (e.g., VBA or Matlab). Some access controls are in place and they must coordinate requirements between teams. *P16-D* describes areas such as local (country-specific) customer interfaces, smaller local ecommerce systems, and local helpdesks which are decentralized.

4.2.1.4 <u>11</u>01 (Aligned*Local*~Simple*Volatile): 1 case *P27-A* deals with a general organizational setting for *new ventures*. For that, units (or startups) within the organization are formed to explore new business models. They consist of agile, integrated teams that can act relatively freely and detached from organization-wide systems and processes. The participant notes that they, for example, are not required to use the central ERP system in the beginning so that they are not dragged down in speed and able to concentrate on the development of the new business. Still, their decisions need to be reviewed by an overarching governance board, and the IT department has veto rights on IT architecture decisions. New systems must, for example, not have a (negative) impact on centrally controlled systems. This configuration is, therefore, comparable to the PoC category explained before but deals with settings that are more complex.

4.2.2 S1-B) Aligned*Simple (Positive)

The second solution (S1-B) in the solution set for *Positive* (S1) implies that *Aligned* and *Simple* together are sufficient for the outcome (independent of which values *Local* and *Volatile* have). S1-B has a lower raw coverage than S1-A (0.55 vs. 0.86) and a significantly lower unique coverage (0.07 vs. 0.38). The solution is, therefore, able to explain 55% of all *Positive* outcomes, but only 7% of them uniquely (no other solution can explain them). The latter are represented by cases with the configurations 1010 and 1011.

4.2.2.1 <u>1111(Aligned*Local*Simple*Volatile):</u> **11** cases This configuration has already been illustrated for S1-A, is also explained by S1-C and covers general governance settings that allow local BUs to build simple, volatile solutions, as well as local PoCs and adaptive reporting solutions.

4.2.2.2 <u>1</u>1<u>1</u>0(<u>Aligned</u>*Local*<u>Simple</u>*~Volatile): 3 cases As already illustrated for S1-A, this configuration represents smaller, local business systems.

4.2.2.3 1010 (Aligned*~Local*Simple*~Volatile): 2 cases Both cases for this configuration deal with management of *cloud-based CRM workflows*. In *P11-A*, BUs are empowered to create reports, dashboards, and workflows in a Salesforce CRM system. They are coached and supported by professionals to retain a consistent data model and organization-wide workflows. The participant explains this as a form of "business DevOps" that allows to be agile and to have 12 releases per month in comparison to 2–3 per year for the on-premise SAP environment which is managed by the IT department. Interestingly, the system emerged as a shadow instance and was then legitimized, improved, and properly integrated with other systems with the help of the IT department. Another participant was setting up a similar arrangement in *P06-A* at the time of the interview. In a new cloud-based CRM, which consolidates multiple older ones, BUs take over maintenance and further development/design of the system. This should allow them to employ more agile procedures without heavy project

processes and to solve challenges closer to the users who best understand their own needs. In this setting, the IT department is still responsible for large system changes, integration with other systems (e.g., email server), vendor/contract management, security, and retains budget control. The largest BU with the most complex requirements takes over coordination of different local requirements to enable synergies and to retain overarching workflow consistency. The participant noted that with cloud technologies the borders between IT and business blur because there is less technical knowledge necessary to operate them. Without having to maintain a datacenter or on-premise infrastructure, there are also fewer continuity risks.

It is interesting to note that these two cases are the exceptions that were responsible for the condition *Local* being only almost always necessary. The question arises why in these two cases locality was not needed. As argued above, this might be due to the way cloud technologies are operated. An essential part of IT responsibility does not enter the organization at all as it is taken over by an external cloud provider who handles the infrastructure/technology for a broad range of clients so that handling it all over one organization without having to limit it to local reach is usually no challenge from the provider perspective. Thus, the need for locality is exempted as there is a third party outside the respective organization handling most of the issues of global use. Additionally, this outside party, the provider, is specifically interested in handling or running its technology over a wide range of organizations and contexts in a standardized manner. Looking at these exceptions from the NCA perspective reveals that these are cases Dul (2015) calls "substitutes". In these cases, there are mechanisms that substitute or compensate the need for locality, which is the reason why locality is not needed in these cases.

4.2.2.4 <u>1011</u> (Aligned*~Local*Simple*Volatile): 0 cases No cases were observed for this possible configuration in solution S1-B which is a result of limited diversity in the data. However, three of the four conditions are present (*Aligned, Simple, Volatile*) that contribute to the outcome *Positive* based on theoretical foundations, and there is also no negative case with the same configuration. A hypothetical example in this category could be represented by a simple event management tool that is developed or sourced by a BU to organize a Christmas party and re-used all over the organization due to its usefulness. If such a case had existed, it again would have been contradicting that *Local* is a necessary condition (in the NCA). But as argued for 1010 where the need for locality can be compensated for even in the absence of volatility, here we assume that a system with a very short-term use and without operational relevance for the core business (as in the hypothetical example) may yield the outcome *Positive* in spite of global use. Again, this hints at locality being a condition that can be compensated for in specific constellations.

4.2.3 S1-C) Local*Simple*Volatile (Positive)

The third solution S1-C in the set S1 describes *Local*, *Simple*, and *Volatile* as sufficient for the outcome *Positive* (independent of which value *Aligned* has). It has a

similar raw coverage (0.41) as S1-B (0.55) and, therefore, a lower one than S1-A (0.86). The unique coverage (0.03) is similar to S1-B (0.07) significantly lower than the one for S1-A (0.38). This is caused by an outlier case for the configuration 0111.

4.2.3.1 <u>1111</u> (Aligned*Local*Simple*Volatile): 11 cases This configuration was already illustrated for S1-A and is also explained by S1-B. To recap, it covers general governance settings that allow local BUs to build simple, volatile solutions, as well as local PoCs and adaptive reporting solutions.

4.2.3.2 0111 (~Aligned*Local*Simple*Volatile): 1 case P01-A describes the only case that is part of a solution leading to the outcome *Positive* which is not *Aligned*.² It deals with a system required by a BU for a large customer project. Initially, the BU asked for a proposal from the IT department to implement the system. However, the BU deemed it to be too expensive and the implementation time too long. Only after three months, the IT department realized that the BU was already working with a standard solution they had procured externally by themselves in the meantime without alignment. It turned out that a simple, temporary cloud solution with an app frontend on tablets was sufficient for the BU to fulfill their temporary requirements during the project duration. The solution was not integrated with the internal systems and required some additional manual steps, but it was sufficient. One of the reasons for the mismatch of expectations between BU and IT department was that the IT department based their proposal on the assumption of building a customized application with full integration into the core ERP system. While the participant assumes that the IT department could have helped if the alignment and communication with the BU would have been better, they recognized it as a positive example for IT managed in the BU under these conditions (Local, Simple, Volatile).

While it is the only case in this category, we decided against just dismissing it because it is a plausible one. It stands for a potential category of positive cases that is just not reflected more broadly in our interviews with IT managers (and could be more prominent in interviews with business managers). Also, this case is the exception that was responsible for the condition *Aligned* being only almost always necessary. As with the exceptions from locality this seems to be due to a substituting mechanism. Again, a cloud provider takes over major parts of IT responsibility, which makes alignment less needed.

4.2.4 S0-A) ~Aligned*~Local*~Volatile (~Positive)

The solution set for ~*Positive*, i.e., a negative outcome, comprises two configurations. The first one described here (S0-A) indicates that ~*Aligned*, ~*Local*, and

² The only other case with outcome *Positive* which is not *Aligned* is *P03-A*. This is the case that caused the contradicting configuration (see Sects. 3.5.3 and 3.5.1) and it is also one of the two outlier cases that caused *Aligned* to be only almost always necessary instead of being always necessary (see Sect. 3.5.2). As a result, this case led to a configuration that is not described by any of the solutions (see Sect. 4.2.6).

245

~*Volatile* together are sufficient for the outcome ~*Positive* (independent of the values for *Simple*). Both S0-A and S0-B have similar raw coverages (0.62 vs. 0.54) and unique coverages (0.31 vs. 0.23). They have, therefore, a comparable explanatory power and are both important to consider for the overall solution set. The consistency value is a perfect 1.00 for S0-A and S0-B because the contradicting configuration from the outlier *P03-A* is excluded during truth table minimization as described in Sect. 3.5.3.

4.2.4.1 0000 (~Aligned*~Local*~Simple*~Volatile): 4 cases This configuration covers the development of permanent/non-volatile critical core systems for which the BU was not aligned with the IT department. P01-B describes the failed development of a large system for a core business process. The BU head decided to do the implementation by oneself together with an external vendor without aligning with the IT department first. They anticipated lower costs, had a strained relationship with the IT department, and wanted to avoid tedious discussions about guidelines, security, or integration. The project was paid with business budget instead of IT budget and ultimately failed after 1 1/2 years with a loss of several million euros. The system ended up with poor quality (documentation, modularization lacking) and poor integration/ interfaces with other systems. At this point, the IT department was involved to save the system and it required an additional year to fix the shortcomings. Despite the initial anticipation of cost savings by the BU head, the participant estimated that it would have been more than 50% cheaper if the IT department had been involved from the beginning. In P12-B a BU similarly implemented a new CRM system together with an external provider without aligning with the IT department. It was heavily customized and grew very complex. Ultimately, the effort was stopped and restarted. After that, the BU still worked together directly with the vendor, but the IT department took over vendor management and quality assurance of the project. In P15-B, a BU had the necessary budget and IT competencies to start implementing a core manufacturing execution system for the company without first aligning with the IT department (the relationship between the two was strained). The IT department was only involved at a very late stage and seen as a fulfillment provider instead of a partner. Due to inefficiencies caused by unclear responsibilities with an external provider and deficits in the solution design, the participant perceived it as a negative case. For P06-B the participant mentioned a central risk controlling system with many dependencies on other systems that was created by a BU without IT alignment. Here, they saw the risks (continuity, regulations, inefficiencies, etc.) outweighing the potential benefits.

4.2.4.2 <u>0010</u> (<u>~Aligned*~Local*Simple*~Volatile</u>): **4** cases This configuration contains four cases that can best be described as *unofficial, non-specific, small, permanent solutions*. The first one (*P03-C*) revolves around a solution for managing company goals which was procured independently without any involvement of the IT department. The requirements were not specific to the BU and did not involve any specific know-how the BU could contribute. The independently selected vendor turned out to be unqualified and system quality was poor (usability

and performance problems due to missing non-functional requirements). The BU initially wanted to avoid having to align with the IT department due to a perceived high effort of doing so. In *P13-A*, a small external provider worked directly with multiple BUs to implement small solutions. The BUs enjoyed its services because their unsolved issues were covered fast and cheap. However, the initiatives were not aligned with the IT department and created high follow up costs due to faulty system integrations and poor security concepts. The provider was then integrated into the IT department so that architectural impact analyses and quality control are possible. In *P14-A*, a doctor implemented a planning software in cooperation with a school project, but operational issues were ignored and remained unsolved. *P03-B* deals with non-standard devices that are not supported and pose a security risk (for example, smartphones or unsecured gateways connected to the internet).

4.2.5 S0-B) ~Aligned*Simple*~Volatile (~Positive)

As explained for S0-A, both S0-A and S0-B in the solution set S0 have similar raw and unique coverages and therefore similar explanatory power. However, they describe a different subset of cases for S0, but share one common configuration (0010).

4.2.5.1 <u>0010</u> (<u>~Aligned</u>*~Local*<u>Simple</u>*<u>~Volatile</u>): 4 cases This configuration (as described before) is shared with S0-A and includes *unofficial, non-specific, small, and permanent* solutions.

4.2.5.2 0110 (~Aligned*Local*Simple*~Volatile): 3 cases Three cases which caused continuity issues are representative for this configuration. These cases can best be described as unofficial, local, small, and permanent solutions. In P16-B, an employee developed a solution for a local business process. They managed to integrate it with a database but did not align with the IT department. Over time, the local BU's processes became dependent on the solution. After the employee left, they demanded money for continued support and even included a time lock in the application before leaving. In another case in the same organization, an employee unexpectedly died and left behind an unsupported, undocumented application they had built. P26-A represents a similar case where the author of an application left the company and it stopped working after some time. Because there was no source code available, no documentation, no versioning, or any professional development standards, it was necessary to completely rebuild the solution. For P18-A the participant denounced inefficient, hard to support and hard to secure Excel and Tableau reports that are created by BUs themselves for long-term solutions instead of requesting proper reports from the IT department.

4.2.6 Configuration not part of a solution

The configuration 0100 consists of three cases. Two of them (*P18-B* and *P05-A*) have an outcome of [0] (and would, therefore, be *~Positive*), but an outlier (*P03-A*) has an outcome of [1]. As described during the truth table minimization in Sect. 3.5.3, the consistency of this configuration for both *Positive* (0.33) and *~Positive* (0.66) lies below the threshold of 0.75, and it can, therefore, not be viewed as contributing to the outcome *Positive* or *~Positive*. As a result, this configuration is not described by any of the solutions. However, the theoretical foundations of the conditions would support an outcome of *~*Positive for this configuration.

4.2.6.1 <u>**0100**</u> (<u>~Aligned</u>*Local*~Simple*<u>~Volatile</u>): **3** cases This configuration is represented by two cases which can be best characterized as *unofficial*, *local*, *complex or high-risk*, *permanent* applications and one outlier case which was considered *Positive* by the participant. *P18-B* describes a critical application built with MS Access in a BU over years. The underlying technology was inadequate for the complex requirements and the solution, for example, did not support concurrent users and created data consistency issues. It took a high financial investment to properly rebuild the solution with the help of the IT department. In *P05-A* management accountants were using spreadsheets without permission instead of the officially mandated SAP system for calculations and were feeding inconsistent data back into the system. They were also not asking the IT department for help when changes to the SAP system was needed. This created a high level of operational and financial risks before the issue was addressed.

P03-A describes a one-year project for the development of a database extension. The solution had clear requirements and covered a critical business process, but the IT department was only shown the finished solution before hearing about the requirements. The solution had some operational issues, and integration with other systems was lacking (which the IT department planned to fix later), but the participant recognized it as a relatively well-made, positive example in the end. This case is, therefore, responsible for this configuration not being part of any solution because it has an outcome of [1] instead of [0] with the same conditions. It is also one of the two cases in the NCA that are responsible for *Aligned* being an almost always necessary condition instead of an always necessary one. Thus, *P03-A* is a contradictory case in several respects and an outlier in both analyses.

5 Discussion

5.1 Sufficient configurations of conditions for positive business-managed IT

In the results section we have described all the different observed categories of cases for the respective configurations. The QCA methodology effectively helped to structure and better understand the data through determination of condition variables, calibration of condition values, and categorization in a truth table.



Fig. 8 Solutions for Positive and ~Positive and observed categories of cases

Figure 8 summarizes the solutions for the outcomes *Positive* and *~Positive* and the underlying observed categories of cases. The numbers in circles reflect the number of cases for each category.

The first notable observation is that all but one case with the outcome *Positive* are *Aligned*. As described in Sect. 4.1.2, the condition Aligned ultimately describes overt cases and therefore BMIT as defined in the terminology section. The data, therefore, shows that instances of BMIT are generally positive and instances of (covert) SIT negative.

This observation is consistent with the QCA solutions. Through truth table minimization the QCA methodology also helps to generalize the findings in form of solutions. In the end, the observed categories of cases are just examples and the derived solutions can help identifying other cases with the same characteristics that may also result in a positive (or negative) outcome. The solution S1-A indicates that *Aligned* and *Local* together are sufficient for *Positive* (independent of the values for *Simple* and *Volatile*). It has the highest raw coverage (percentage of positive cases explained) and unique coverage (percentage of positive cases uniquely explained by this solution). We, therefore, deem it to be the most important solution. This is consistent with the results of the NCA which identified both *Aligned* and *Local* as "almost always necessary" for the outcome *Positive*. The solution S1-A already demonstrates the principle of "conjunctural causation" which is core to QCA as previously described. *Aligned* alone is not sufficient for *Positive*, but it is sufficient in combination with *Local*.

However, there are other solutions for *Positive* as well which demonstrates the principle of "equifinality". Besides S1-A, there are also S1-B and S1-C which form alternative causal paths to the outcome. Therefore, it is also sufficient for a positive outcome if *Aligned* and *Simple* are fulfilled (independent of the values for *Local* and *Volatile*). However, S1-B has a lower raw coverage than S1-A and a significantly lower unique coverage. Due to its low coverage we deem the last solution S1-C to be the least important. It indicates that *Local*, *Simple*, and *Volatile* are sufficient for *Positive*, independent of the value for *Aligned*. However, this case together with the two *~Local* cases deserve special attention. In combination with the results from the NCA that put forth the two almost always necessary conditions *Aligned* and *Local*

we see that these are the three exceptional cases that were responsible for assuming almost always and not always necessary conditions. All of the three are based on cloud technology, which indicates that cloud solutions indeed provide the opportunity to exempt the need for alignment and locality. It remains to be seen, whether with further dissemination of cloud technology the necessity assumption needs to be fully abandoned as these solutions gain further momentum.

We also calculated two solutions for *~Positive* (negative) that are different from just negating the solutions for *Positive*. This demonstrates that QCA allows to consider asymmetric relationships and does not automatically assume symmetry between occurrence and non-occurrence of conditions and outcome. S0-A indicates that it is sufficient for the outcome to be *~Positive* when *~Aligned*, *~Local*, and *~Volatile* are fulfilled (independent of the value for *Simple*). In other words, cases are negative when there was no alignment with the IT department and the system in question was built as an organization-wide, permanent solution. With those three conditions it makes no difference for the outcome if the case deals with a simple or a complex system. Similarly, S0-B (*~Aligned, Simple, ~Volatile*) indicates that cases are negative when there was no alignment with the IT department and it is about a simple, permanent solution (no matter if local or organization-wide). In terms of necessary conditions, the outcome *~Positive* clearly implies that a case is *~Aligned* and *~Volatile*.

Based on those results and the detailed findings during the interviews, it could be hypothesized that the types of cases we observed for *~Positive* could be either turned into positive examples of BMIT by including *Aligned* or should be fully managed by the IT department. Critical core systems, non-specific, organization-wide solutions, or workplace IT are areas that were frequently pointed out during the interviews as best being managed solely by the IT department. In other areas, the cases show some positive examples of BMIT. For most of them, the condition *Aligned* not only indicates that the initiative is officially approved, but there is a responsibility split between the BU and the IT department. Most commonly, the IT department is responsible for providing common infrastructure (incl. platforms), vendor management, security, and architecture standards because it has more experience in those areas and can take a global view.

5.2 Balancing local responsiveness and global standardization, autonomy, and control

Consistent with findings from literature, the observed companies are engaging in BMIT to enable BUs to rapidly react to changes in business conditions or market changes (Tambo and Bækgaard 2013; Zimmermann and Rentrop 2012). Ferneley (2007) has argued that covert systems emerge in response to dynamic environments. As described by Winkler and Brown (2014b) we found examples where product IT or R&D was managed in separate BUs because they require high innovation through IT. Zimmermann et al. (2017) have concluded that uncertainty and unclear requirements can make it more efficient to find solutions to local requirements directly in the BUs. We also found that BMIT enables BUs to pursue smaller, possibly exploratory projects that may otherwise be neglected by the IT department (Chua and Storey 2016). This could be due to a lack

of business acumen or resources in the IT department (Koch and Peters 2017). Gartner (2016) has even argued that CIOs should focus on their core competencies and be aware of or give up control of areas they cannot pursue profitably or successfully.

However, there are risks and benefits to (partially) giving up IT control to BUs. Silic et al. (2016) found that there may be increased costs associated with IT managed in the BUs but they seem to be outweighed by associated benefits. Winkler and Brown (2014b) also found that companies with well-balanced IT decision rights show better Business-IT Alignment and achieve superior firm performance. They have suggested IT reference frameworks such as COBIT to apply overarching accountability schemes. Urbach and Ahlemann (2016) have argued that a separation of business and IT is no longer adequate and that in the future, IT innovation will come through close collaboration between business and IT. Fürstenau et al. (2016) have discussed aspects of achieving a balance between granting BUs the autonomy and flexibility to manage their own IT solutions and leveraging organization-wide economies of scale through centralization and standardization. Too much autonomy results in redundancy and incompatible solutions and too little autonomy prevents the agility to react to local demands (Kretzer 2015; Winkler and Brown 2014b). Too strict IT policies may diminish an organization's ability to rapidly react to changing requirements (Taal et al. 2016). Tiwana and Konsynski (2010) also found that there is a relationship between decentralization of IT governance and increased IT agility.

In our study, we observed different forms of co-governance models for positive cases of BMIT. They resemble the "federal" IT governance model described by Weill and Ross (2004). In practice, those are generalizations of allocated IT task responsibilities as described for individual SIT instances by Zimmermann et al. (2016a). Consistent with a finding from Silic et al. (2016), we, for example, observed that IT security, privacy, and compliance are usually in the responsibility of the IT department which makes sure those areas are properly considered in BMIT projects. Integration of solutions with other systems is another area that is adequately managed by the IT department due to its global perspective and expertise. Bygstad and Iden (2017) describe "lightweight" solutions that are developed in separate processes but subject to the "heavyweight" regime when integrated with other systems. Winkler and Brown (2014b) have stated that it is a widely adopted pattern that infrastructure decisions are centralized and business application decisions made by the BUs. Specifically, the standardized "transportation architecture" for which synergies and scale-effects can be achieved (servers, databases, networks, etc.) is managed and provided by the IT department in this model (Brown and Magill 1994; Zimmermann et al. 2016b).

As we also found in our study, platforms are a form of infrastructure that is provided by IT departments for BMIT initiatives. Krancher and Luther (2015) have explained that platforms (PaaS) can be used to enforce uniformity, but also allow to enhance agility through exploitation of knowledge embedded in the technology (and through abstracting further from the complex layers of the technology stack). Kretzer and Maedche (2014) have highlighted the generative characteristics of platforms and have compared them to the research stream on SIT. Both similarly deal with supplementing systems with new features and solutions. As we also found in our study, they have highlighted the area of BI as an example. Besides BI systems, ERP systems are increasingly serving as a platform. They fulfill their purpose of data homogenization (Spierings et al. 2012) but also allow flexible re-programmability (generativity) if tools can be added to take advantage of the shared data resources (Yoo et al. 2012).

To conclude, IT managers face the challenge of balancing control and autonomy in their IT governance model. As our research shows, BMIT provides potential benefits to organizations. We also show sufficient conditions for positive manifestations of BMIT that can help defining IT governance strategies. Even if not recognizing the potential benefits, a total control and centralization of IT tasks might not be feasible. Chua and Storey (2016) have argued that the IT department must deal with the emerging reality of BUs being able to procure and operate their own solutions because, for example, alternatives are not made available in time or not timely enough. This also raises questions about the potential future role of the IT department or the CIO in the organization. The IT department may increasingly become a coordinating and advisory body (Chua and Storey 2016) and the CIO may play an orchestrating role by being a boundary spanner between different intra-organizational and inter-organizational networks (Peppard 2016). Gartner (2016) expects that IT managed within BUs will increase because technology becomes more deeply embedded in all business activities. Also, with cloud technologies gaining more momentum, our results already indicate that what is considered necessary for successful SIT/BMIT today, may no longer be tomorrow.

6 Conclusion

In this paper, we applied QCA to differentiate positive and negative forms of SIT/ BMIT (RQ1) and to determine influential conditions that can be observed for instances of SIT/BMIT (RQ2). We also derived sufficient configurations of conditions for positive and negative forms of those instances (RQ3). To do so, we conducted interviews with 29 IT managers from different companies and industries who described 42 different cases of SIT/BMIT. The conditions determined through a dialog between data and theory are Aligned, Local, Simple, and Volatile. The QCA method revealed multiple configurations that can lead to outcome Positive. Ssufficient configurations for positive cases of IT managed by BUs: S1-A) Aligned*Local, S1-B) Aligned*Simple, and S1-B) Local*Simple*Volatile. Due to its highest explanatory power and because both Aligned and Local resulted as "almost always necessary" conditions in the NCA, we deem S1-A as being the most important one. We also found two solutions—S0-A) ~Aligned*~Local*~Volatile, S0-B) ~Aligned*Simple*~Volatile-which are sufficient for a negative outcome (~Positive), both of them having similar explanatory power. Because almost all positive cases are also *Aligned* (which implicates overtness), we call BMIT as generally positive in contrast to generally negative ones in form of (covert) SIT. In spite of this general tendency, we want to stress that Volatile systems, if Simple and Local, yield a positive outcome regardless of their alignment. This shows that our configurational analysis might be in line with general expectations but it also provides a more nuanced understanding of potential alternatives that may also yield positive results and may become even more important in the future.

The paper makes a theoretical contribution by advancing the research stream of SIT/BMIT. Our insights help to better understand covert and overt instances of systems

managed by BUs and can be used to further advance IT governance approaches and build new theories. Similarly, our contribution to practice is that IT managers can use our findings and examples to make informed decisions about their IT governance setup. Specifically, they can use the sufficient configurations for positive BMIT to determine the distribution of IT task responsibilities for certain types of systems. As the use of QCA in IS is not very common, we also claim a small methodological contribution by demonstrating and operationalizing the usage of QCA in the field of IS, especially when using qualitative data in the form of interviews. However, considering the underlying qualitative data we must concede that our approach suffers from similar limitations as any study using an open coding approach with human coders.

Further potential limitations need to be kept in mind. For our study, we primarily interviewed CIOs. One might argue that by definition CIOs cannot know SIT instances and therefore SIT is not considered unless BU managers are interviewed. However, as most SIT instances might be detected at some point during their lifecycle, we argue that not uncovering them early in their lifecycle might be a drawback for our study, but not a severe one. If the uncovered instances stay in place, they become BMIT or regular systems managed by IT, if they are decommissioned, they cease to exist, but the respective CIO knows they existed in the past. In any of these cases, the CIO can report on them. Another limitation related to relying on CIOs for interview data is a potential bias in our study. CIOs tend to be more critical with respect to SIT/ BMIT (Koch et al. 2014) and are less likely to endorse participatory governance in comparison to business professionals (Andriole 2015). Almost all our interviews were conducted with CIOs from the EU, but most of them were from the DACH region which might have an influence on the results due to regional and cultural differences. Future scholars interested in advancing our research could, therefore, involve more business representatives in their studies and broaden their geographical scope for data collection. They could also increase the number of examined conditions, though this would require a significant increase of underlying cases and it needs to be carefully considered which ones to add. In a study about allocation of decision rights for SaaS, Winkler and Brown (2014a) for example found no influences due to industry and size of organizations. Future research could also compare different forms of IT governance and development methodologies that closely involve the BUs. Scholars can, for example compare BMIT and the included co-governance models with setups where the IT department employs fully agile teams directly in the BUs. They may do so to reach a similar goal of being able to react fast to changing business requirements. A possible outcome could be that both converge at some point on the way to a setting where, as one of the participants described it, "everything is IT".

Acknowledgements Open Access funding provided by Projekt DEAL.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission

directly from the copyright holder. To view a copy of this licence, visit http://creativecommons.org/licen ses/by/4.0/.

References

Alter S (2014) Theory of workarounds. Commun ACM 34:1041–1066

Andriole SJ (2015) Who owns IT? Commun ACM 58:50-57

- Basurto X, Speer J (2012) Structuring the calibration of qualitative data as sets for qualitative comparative analysis (QCA). Field Methods 24:155–174
- Behrens S (2009) Shadow systems: the good, the bad and the ugly. Commun ACM 52:124–129
- Behrens S, Sedera W (2004) Why do shadow systems exist after an ERP implementation? Lessons from a case study. In: PACIS 2004 proceedings
- Berg-Schlosser D, de Meur G (2008) Comparative research design: case and variable selection. In: Rihoux B, Ragin CC (eds) Configurational comparative methods: qualitative comparative analysis (QCA) and related techniques. Sage Publications, Thousand Oaks, pp 19–32
- Blichfeldt BS, Eskerod P (2008) Project portfolio management: there's more to it than what management enacts. Int J Project Manag 26:357–365
- Brown CV, Magill SL (1994) Alignment of the IS functions with the enterprise: toward a model of antecedents. MIS Q 18:371–403
- Buche A, Carstensen J (2009) Qualitative comparative analysis: Ein Überblick. In: Kriwy P, Gross C (eds) Klein aber fein!: quantitative empirische Sozialforschung mit kleinen Fallzahlen. VS Verlag für Sozialwissenschaften, Wiesbaden, pp 65–99
- Buchwald A, Urbach N (2012) Exploring the role of un-enacted projects in IT project portfolio management. In: ICIS 2012 proceedings
- Buchwald A, Urbach N, Ahlemann F (2014a) Business value through controlled IT: toward an integrated model of IT governance success and its impact. J Inf Technol 29:128–147
- Buchwald A, Urbach N, Ahlemann F (2014b) Understanding the organizational antecedents of bottom-up un-enacted-projects. In: ECIS 2014 proceedings
- Bygstad B (2016) Generative innovation: a comparison of lightweight and heavyweight IT. J Inf Technol 32:180–193
- Bygstad B, Iden J (2017) A governance model for managing lightweight IT. In: WorldCIST 2017 proceedings
- Capgemini (2016) IT-Trends-Studie 2016. https://www.capgemini.com/de-de/resources/it-trends-studi e-2016/. Accessed 4 Nov 2017
- Chua CEH, Storey VC (2016) Bottom-up enterprise information systems: rethinking the roles of central IT departments. Commun ACM 60:66–72
- Chua C, Storey V, Chen L (2014) Central IT or Shadow IT? Factors shaping users' decision to go rogue with IT. In: ICIS 2014 proceedings
- Chun M, Mooney J (2009) CIO roles and responsibilities: twenty-five years of evolution and change. Inf Manag 46:323–334
- Corbin J, Strauss A (2008) Basics of qualitative research, 3rd edn. Sage Publications, Thousand Oaks
- Crilly D, Zollo M, Hansen MT (2012) Faking it or muddling through? Understanding decoupling in response to stakeholder pressures. Acad Manag J 55:1429–1448
- Dittes S, Urbach N, Ahlemann F, Smolnik S, Müller T (2015) Why don't you stick to them? Understanding factors influencing and counter-measures to combat deviant behavior towards organizational IT standards. In: Wirtschaftsinformatik 2015 proceedings
- Dul J (2015) Necessary condition analysis (NCA). Organ Res Methods 19:10-52
- Dul J (2019) Conducting necessary condition analysis: for business and management students. Sage Publications, Thousand Oaks
- Dul J, Vis B, Goertz G (2018) Necessary condition analysis (NCA) does exactly what it should do when applied properly: a reply to a comment on NCA. Sociol Methods Res. https://doi. org/10.1177/0049124118799383
- Ebeling B, Köpp C, Breitner M (2013) Diskussion eines Prototyps für das dezentrale Management von Forschungsressourcen an deutschen Hochschulinstituten. In: Wirtschaftsinformatik 2013 proceedings

- El Sawy OA, Malhotra A, Park Y, Pavlou PA (2010) Seeking the configurations of digital ecodynamics: it takes three to tango. Inf Syst Res 21:835–848
- Ferneley EH (2007) Covert end user development: a study of success. J Organ End User Comput 19:62-71
- Fiss PC (2007) A set-theoretic approach to organizational configurations. Acad Manag Rev 32:1180-1198
- Fiss PC (2011) Building better causal theories: a fuzzy set approach to typologies in organization research. Acad Manag J 54:393–420
- Forsythe PJ (2012) Profiling customer perceived service quality expectations in made-to-order housing construction in Australia. Eng Construct Architect Manag 19:587–609
- Fürstenau D, Rothe H (2014) Shadow IT systems: discerning the good and the evil. In: ECIS 2014 proceedings
- Fürstenau D, Sandner M, Anapliotis D (2016) Why do shadow systems fail? An expert study on determinants of discontinuation. In: ECIS 2016 proceedings
- Fürstenau D, Rothe H, Sandner M (2017) Shadow systems, risk, and shifting power relations in organizations. Commun Assoc Inf Syst 41:43–61
- Gartner (2016) Coming to terms with business unit IT to prepare for digital business. https://www.gartn er.com/doc/3288932/coming-terms-business-unit-it. Accessed 12 Jan 2020
- Gartner (2017) Make the best of Shadow IT. https://www.gartner.com/smarterwithgartner/make-the-bestof-shadow-it/. Accessed 12 Jan 2020
- Gartner (2018) Gartner survey reveals many digital workers think senior managers are out of touch with their technology needs. https://www.gartner.com/en/newsroom/press-releases/2018-06-05-gartn er-survey-reveals-many-digital-workers-think-senior-managers-are-out-of-touch-with-their-techn ology-needs. Accessed 12 Jan 2020
- Goertz G (2003) Cause, correlation, and necessary conditions. In: Goertz G, Starr H (eds) Necessary conditions: theory, methodology, and applications. Rowman and Littlefield, Lanham
- Goertz G (2006) Social science concepts: a user's guide. Princeton University Press, Princeton
- Gozman D, Willcocks L (2015) Crocodiles in the regulatory swamp: navigating the dangers of outsourcing, SaaS and Shadow IT. In: ICIS 2015 proceedings
- Györy A, Cleven A, Uebernickel F, Brenner W (2012) Exploring the shadows: IT governance approaches to user-driven innovation. In: ECIS 2012 proceedings
- Haag S (2015) Appearance of dark clouds? An empirical analysis of users' shadow sourcing of cloud services. In: Wirtschaftsinformatik 2015 proceedings
- Haag S, Eckhardt A (2017) Shadow IT. Bus Inf Syst Eng 59:469-473
- Haworth-Hoeppner S (2000) The critical shapes of body image: the role of culture and family in the production of eating disorders. J Marriage Fam 62:212–227
- Hetzenecker J, Sprenger S, Kammerer S, Amberg M (2012) The unperceived boon and bane of cloud computing: end-user computing vs. integration. In: AMCIS 2012 proceedings
- Horlach B, Drews P, Schirmer I, Boehmann T (2017) Increasing the agility of IT delivery: five types of bimodal IT organization. In: HICSS 2017 proceedings
- Huber M, Zimmermann S, Rentrop C, Felden C (2016) The relation of shadow systems and ERP systems: insights from a multiple-case study. Systems 4:11
- Huuskonen S, Vakkari P (2013) "I did it my way": social workers as secondary designers of a client information system. Inf Process Manag 49:380–391
- Jones D, Behrens S, Jamieson K, Tansley EE (2004) The rise and fall of a shadow system: lessons for enterprise system implementation. In: ACIS 2004 proceedings
- Kappelman L, Johnson V, Torres R, Maurer C, McLean E (2018) A study of information systems issues, practices, and leadership in Europe. Eur J Inf Syst 21:1–17. https://doi.org/10.1080/09600 85X.2018.1497929
- Kearns GS, Lederer AL (2003) A resource-based view of strategic IT alignment: how knowledge sharing creates competitive advantage. Decis Sci 34:1–29
- Klotz S, Kopper A, Westner M, Strahringer S (2019) Causing factors, outcomes, and governance of Shadow IT and business-managed IT: a systematic literature review. IJISPM 7:15–43
- Koch H, Peters U (2017) Shadow analytics. In: AMCIS 2017 proceedings
- Koch H, Zhang S, Giddens L, Milic N, Yan K, Curry P (2014) Consumerization and IT department conflict. In: ICIS 2014 proceedings
- Köffer S, Ortbach K, Niehaves B (2014) Exploring the relationship between IT consumerization and job performance: a theoretical framework for future research. Commun Assoc Inf Syst 35:261–283

- Köffer S, Ortbach K, Junglas I, Niehaves B, Harris J (2015) Innovation through BYOD? Bus Inf Syst Eng 57:363–375
- Kopper A (2017) Perceptions of IT managers on Shadow IT. In: AMCIS 2017 proceedings
- Kopper A, Westner M (2016a) Deriving a framework for causes, consequences, and governance of Shadow IT from literature. In: MKWI 2016 proceedings
- Kopper A, Westner M (2016b) Towards a taxonomy for Shadow IT. In: AMCIS 2016 proceedings
- Kopper A, Westner M, Strahringer S (2017) Kontrollierte Nutzung von Schatten-IT. HMD Praxis der Wirtschaftsinformatik 54:97–110
- Kopper A, Fürstenau D, Zimmermann S, Rentrop C, Rothe H, Strahringer S, Westner M (2018a) Business-managed IT: a conceptual framework and empirical illustration. In: ECIS 2018 proceedings
- Kopper A, Fürstenau D, Zimmermann S, Klotz S, Rentrop C, Rothe H, Strahringer S, Westner M (2018b) Shadow IT and business-managed IT: a conceptual framework and empirical illustration. Int J IT/ Bus Align Govern 9:53–71. https://doi.org/10.4018/IJITBAG.2018070104
- Krancher O, Luther P (2015) Software development in the cloud: exploring the affordances of platformas-a-service. In: ICIS 2015 proceedings
- Kretzer M (2015) Linking report individualization and report standardization: a configurational perspective. In: ECIS 2015 proceedings
- Kretzer M, Maedche A (2014) Generativity of business intelligence platforms: a research agenda guided by lessons from Shadow IT. In: MKWI 2014 proceedings
- Lasrado L, Vatrapu R, Andersen KN (2016) A set theoretical approach to maturity models: guidelines and demonstration. In: ICIS 2016 proceedings
- Levy Y, Ellis TJ (2006) A systems approach to conduct an effective literature review in support of information systems research. Inf Sci Int J Emerg Transdiscipl 9:181–212
- Liu Y, Mezei J, Kostakos V, Li H (2015) Applying configurational analysis to IS behavioural research: a methodological alternative for modelling combinatorial complexities. Inf Syst J 27:59–89
- Lund-Jensen R, Azaria C, Permien FH, Sawari J, Bækgaard L (2016) Feral information systems, shadow systems, and workarounds: a drift in IS terminology. Proc Comput Sci 100:1056–1063
- Magunduni J, Chigona W (2018) Revisiting Shadow IT research: What we already know, what we still need to know, and how do we get there? In: ICTAS 2018 proceedings
- Melo C, Moraes J, Ferreira M, Figueiredo R (2017) A method for evaluating end-user development technologies. In: AMCIS 2017 proceedings
- Myers MD, Newman M (2007) The qualitative interview in IS research: examining the craft. Inf Organ 17:2–26
- Ortbach K, Brockmann T, Stieglitz S (2014) Drivers for the adoption of mobile device management in organizations. In: ECIS 2014 proceedings
- Panko RR, Port DN (2012) End user computing: the dark matter (and dark energy) of corporate IT. In: HICSS 2012 proceedings
- Park Y, El Sawy OA, Fiss PC (2017) The role of business intelligence and communication technologies in organizational agility: a configurational approach. J Assoc Inf Syst 18:648–686
- Paulsson W, Johansson B (2013) Spreadsheets use in budgeting: a dialectic process theory perspective. In: AMCIS 2013 proceedings
- Peppard J (2016) Rethinking the concept of the IS organization. Inf Syst J 28:76-103
- Ragin CC (2009) Redesigning social inquiry: fuzzy sets and beyond. University of Chicago Press, Chicago
- Ragin CC (2017) User's guide to fuzzy-set/qualitative comparative analysis. University of Arizona, Tucson
- Ragin CC, Shulman D, Weinberg A, Gran B (2003) Complexity, generality, and qualitative comparative analysis. Field Methods 15:323–340
- Rantala K, Hellström E (2001) Qualitative comparative analysis and a hermeneutic approach to interview data. Int J Soc Res Methodol 4:87–100
- Rentrop C, Zimmermann S (2012) Shadow IT evaluation model. In: FedCSIS 2015 proceedings
- Rihoux B (2006) Qualitative comparative analysis (QCA) and related systematic comparative methods: recent advances and remaining challenges for social science research. Int Sociol 21:679–706
- Rihoux B, Ragin CC (eds) (2008) Configurational comparative methods: qualitative comparative analysis (QCA) and related techniques. Sage Publications, Thousand Oaks
- Röder N, Wiesche M, Schermann M, Krcmar H (2014) Why managers tolerate workarounds: the role of information systems. In: AMCIS 2014 proceedings

- Schneider MR, Eggert A (2014) Embracing complex causality with the QCA method: an invitation. J Bus Mark Manag 7:312–328
- Schneider CQ, Wagemann C (2010) Standards of good practice in qualitative comparative analysis (QCA) and fuzzy-sets. Comp Sociol 9:397–418
- Schneider CQ, Wagemann C (2012) Set-theoretic methods for the social sciences: a guide to qualitative comparative analysis. Cambridge University Press, Cambridge
- Sedera D, Lokuge S, Grover V, Sarker S, Sarker S (2016) Innovating with enterprise systems and digital platforms: a contingent resource-based theory view. Inf Manag 53:366–379
- Siewert MB (2017) Qualitative comparative analysis. In: Jäckle S (ed) Neue trends in den Sozialwissenschaften: innovative Techniken f
 ür qualitative und quantitative Forschung. Springer Fachmedien, Wiesbaden, pp 273–305
- Silic M, Back A (2014) Shadow IT: a view from behind the curtain. Comput Secur 45:274-283
- Silic M, Silic D, Oblakovic G (2016) Influence of Shadow IT on innovation in organizations. Complex Syst Inf Model Q 8:68–80
- Singh H (2015) Emergence and consequences of drift in organizational information systems. In: PACIS 2015 proceedings
- Spierings A, Kerr D, Houghton L (2012) What drives the end user to build a feral information system? In: ACIS 2012 proceedings
- Taal A, Le J, de Leon AP, Sherer JA, Jenson KS (2016) Technological and information governance approaches to data loss and leakage mitigation. Comput Sci Inf Technol 4:181–187
- Tambo T, Bækgaard L (2013) Dilemmas in enterprise architecture research and practice from a perspective of feral information systems. In: EDOCW 2013 proceedings
- Techtarget (2014) What is embedded IT? http://searchcio.techtarget.com/definition/embedded-IT. Accessed 11 Jan 2020
- Techtarget (2016) What is citizen development? http://searchsalesforce.techtarget.com/definition/citizendevelopment. Accessed 11 Jan 2020
- Tiwana A, Konsynski B (2010) Complementarities between organizational IT architecture and governance structure. Inf Syst Res 21:288–304
- Urbach N, Ahlemann F (2016) Schatten-IT als gelebte Praxis: IT-Innovationen werden in interdisziplinären Teams in den Fachabteilungen erarbeitet. In: Urbach N, Ahlemann F (eds) IT-Management im Zeitalter der Digitalisierung. Springer, Berlin, pp 67–75
- Vis B, Dul J (2016) Analyzing relationships of necessity not just in kind but also in degree: complementing fsQCA with NCA. Sociol Methods Res 47:1–28
- Wagemann C, Buche J, Siewert MB (2016) QCA and business research: work in progress or a consolidated agenda? J Bus Res 69:2531–2540
- Weill P, Ross JW (2004) IT governance: how top performers manage IT decision rights for superior results. Harvard Business Press, Boston
- Wendler R, Bukvova H, Leupold S (2013) Qualitative comparative analysis in information systems and Wirtschaftsinformatik. In: Wirtschaftsinformatik 2013 proceedings

Winkler T, Brown CV (2014a) Horizontal allocation of decision rights for on-premise applications and software-as-a-service. J Manag Inf Syst 30:13–48

- Winkler T, Brown CV (2014b) Organizing and configuring the IT function. In: Topi H, Tucker AB (eds) Computing handbook, 3rd edn. Chapman and Hall, Boca Raton, pp 57.1–57.14
- Yin RK (2003) Case study research: design and methods, 3rd edn. Sage Publications, Thousand Oaks
- Yoo Y, Boland RJ Jr, Lyytinen K, Majchrzak A (2012) Organizing for innovation in the digitized world. Organ Sci 23:1398–1408
- Zimmermann S, Rentrop C (2012) Schatten-IT. HMD Praxis der Wirtschaftsinformatik 49:60-68
- Zimmermann S, Rentrop C, Felden C (2014) Managing Shadow IT instances: a method to control autonomous IT solutions in the business departments. In: AMCIS 2014 proceedings
- Zimmermann S, Rentrop C, Felden C (2016a) Governing identified Shadow IT by allocating IT task responsibilities. In: AMCIS 2016 proceedings
- Zimmermann S, Rentrop C, Felden C (2016b) Governing IT activities in business workgroups: design principles for a method to control identified Shadow IT. In: Abramowicz W, Alt R, Franczyk B (eds) Bus Inf Syst. Springer, Cham, pp 252–264

Zimmermann S, Rentrop C, Felden C (2017) A multiple case study on the nature and management of shadow information technology. J Inf Syst 31:79–101

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.