SHADOW IT SYSTEMS: DISCERNING THE GOOD AND THE EVIL

Daniel Fürstenau
Freie Universität Berlin, Berlin, Germany, daniel.fuerstenau@fu-berlin.de

Hannes Rothe
Freie Universität Berlin, Berlin, Berlin, Germany, hannes.rothe@fu-berlin.de

Daniel Fürstenau and Hannes Rothe, 2014, "SHADOW IT SYSTEMS: DISCERNING THE GOOD AND THE EVIL", Proceedings of the European Conference on Information Systems (ECIS) 2014, Tel Aviv, Israel, June 9-11, 2014, ISBN 978-0-9915567-0-0
http://aisel.aisnet.org/ecis2014/proceedings/track15/9

SHADOW IT SYSTEMS: DISCERNING THE GOOD AND THE EVIL

Complete Research

Fürstenau, Daniel, Freie Universität Berlin, Germany, daniel.fuerstenau@fu-berlin.de

Rothe, Hannes, Freie Universität Berlin, Germany, hannes.rothe@fu-berlin.de

Abstract

Shadow IT is becoming increasingly important as digital work practices make it easier than ever for business units crafting their own IT solutions. Prior research on shadow IT systems has often used fixed accounts of good or evil: They have been celebrated as powerful drivers of innovation or demonized as lacking central governance. We introduce a method to IT managers and architects enabling a more nuanced understanding of shadow IT systems with respect to their architectural embeddedness. Drawing on centrality measures from network analysis, the method portrays shadow IT systems as most critical if they hold a central position in a network of applications and information flows. We use enterprise architecture data from a recycling company to demonstrate and evaluate the method in a real project context. In the example, several critical and yet disregarded shadow IT systems have been identified and measures were taken to govern them decently.

Keywords: Shadow IT systems, IS architectures, Network analysis, Centrality measures, Design science research

Introduction and Motivation

Shadow IT systems are software applications or extensions to existing software (1.) neither developed nor (2.) controlled by an organization's central IT department (cf. Rentrop and Zimmermann, 2012). They are mainly developed or commissioned by a business unit to support specific business processes (Rentrop and Zimmermann, 2012; György et al., 2012). They may therefore fill a gap between solutions offered by a central IT department and the needs of users in a business unit (cf. Jones et al., 2004; Behrens, 2009).

Are shadow IT systems the good or the evil? We find existing research dispersed: Either shadow IT systems are celebrated, as important drivers of innovation (cf. Raden, 2005; Panko, 2006; Behrens, 2009), or demonized, as lacking central governance and control (cf. Rentrop and Zimmermann, 2012). We believe research has not yet moved towards a nuanced understanding of when shadow IT systems are harmful or beneficial for organizations. It is subject to much debate whether innovation potentials overcompensate for lacking central control over systems. This is important as the magnitude of shadow IT is more and more increasing: As technical competencies in business units are more widely available and "software as a service"-solutions are readily accessible, non-IT units are in a much better position today crafting new digital workflows themselves. Furthermore, business units' practices get entangled with digital tools more intensively than ever. Consequently, it becomes increasingly difficult for IT managers to govern the increasing variety of IT systems. Thus, a need for research arises on factors guiding the focus on critical IT systems within the architecture.

Our central theme is that shadow IT systems' roles depend on how they position in organizations' IS architectures. We argue that strongly embedded shadow IT systems are a potential threat as their failure may cause architectural decay, unintended ripple effects, and service interruptions. Shadow IT systems embedding weakly may, however, promote user-driven innovation and solve local challenges.

In this paper, we suggest a method to identify shadow IT systems and assess their potential effects on the application architecture. To construct the method, we build on a design science tradition (Peffers et al., 2007). The method is intended to support IT managers and architects in IT consolidation projects or as part of their continuous transformation efforts. We proceed as follows: To begin with, we review the existing literature on shadow IT systems (sec.2). We then motivate the need for a new method and build the method's blocks (sec.3). Next, we apply a case study in a recycling company to demonstrate and evaluate the method (sec.4). We finally discuss findings, limitations, and future directions (sec.5).

1 Shadow IT Systems – the Good and/or the Evil?

Several fields of information systems (IS) research discuss shadow IT: IS security, IT alignment and the IS architecture branch of the IS strategy and organization literature. We position our paper in the latter stream. Our starting point is a literature review by György et al. (2012) showing that research on shadow IT systems is still infant. Up to now, researchers label shadow IT systems differently: "Shadow IT", "rogue IT", "shadow systems" or "bolt-on" systems, mostly reflecting on a specific attribute for the line of argument of the pertaining article. We stick to "shadow IT systems" as it emphasizes an attribute that is important for our study as we explain below. Additionally, the term has been used most frequently throughout various publications. In the following, we discuss opportunities and risks of shadow IT systems – from an IS strategy and organization perspective.

What is the rationale behind shadow IT systems? Sometimes internal or external factors force business units to react instantly. For instance, a project coordinator may need to create a controlling instrument in short time as the management board expects her or him to present the project. Often, Microsoft Excel becomes the software of choice, as it is inexpensive, ubiquitous and leads to fast results (Raden, 2005). In these cases, a business unit may find shadow IT systems more effective than awaiting protracted decision-making by a central IT department (cf. Panko, 2006; György et al., 2012). Sometimes the IT department lacks the competences or is – perceived as – unable to create a service solution (cf. Smyth and Freeman, 2007). Even the mere ability of a business unit to create software solutions on their own raises the probability of using shadow IT systems (cf. Raden, 2005). This is also the case if business units can outsource services easily, e.g. with a 'software as a service' supplier (Rentrop and Zimmermann, 2012). Shadow IT systems are autonomously established by business units: They deeply understand their target groups' demands and create solutions solving their specific issue (cf. György et al., 2012). This bottom-up approach may result in a creative solution. Shadow IT systems are even more likely to diffuse in an organization and to become accepted by the involved units (cf. Behrens, 2009).

Why may shadow IT systems be perceived as threats? We pointed out shadow IT systems are (1.) not developed by a central IT department. Consequently, employees in a business unit spend a significant amount of resources to develop and maintain shadow IT systems (cf. Raden, 2005). If their development lacks quality assurance and monitoring, inconsistent business logics or assumptions underlying algorithms or data models can become inscribed into the application. Furthermore, shadow IT systems are (2.) distributed in a decentral way throughout the organization. Hence, they are often redundant to applications offered in other parts of the organization. As a result of their distribution, shadow IT systems lack central control in multiple ways: First of all, a central IT department is often times unaware of how extensive shadow IT systems are used in the organization.

Additionally – even if shadow IT usage is transparent – developing and maintaining shadow IT systems burdens a business unit's budgets (cf. György et al., 2012). Those units will hardly be eager to relinquish control over an application that already created sunk costs. Additionally, those systems are often created by only a few employees who feel emotionally attached to their solution. Hence, shadow IT systems can become subject to (micro-) political conflicts within the organization (cf. Behrens, 2009). Finally, shadow IT systems affect the evolution of an organization's IS architecture. Shadow IT systems become embedded in work routines. They are complemented by a social dimension of the IS architecture (Behrens, 2009). Furthermore, business units create interfaces to transfer data from the shadow IT systems to other applications. Thus, business units invest in specific shadow IT systems and in turn disinvesting becomes less desirable. As a flipside, the complexity of the overall architecture rises. Especially an increase of the variety of applications and their interactions drive the (partly hidden) maintenance costs for the overall architecture (cf. Schneberger and McLean, 2003).

Shadow IT systems may be an (unanticipated) risk for an organization but also "diamonds in the rogue" (cf. Behrens, 2009). We will demonstrate that not every shadow IT system is an equally high threat to the overall information systems architecture. In the following, we introduce our method taking both sides into consideration.

2 A Method to Detect and Evaluate Shadow IT Systems

We chose a design science approach (Peffers et al. 2007). Design science research is an important and widely accepted form of conducting research in the IS community (Gregor and Hevner, 2013; Hevner and Chatterjee, 2010; Hevner et al., 2004). Our research followed a step-wise, iterative procedure as outlined by Peffers et al. (2007:54): Problem definition, definition of scope and objectives, design and development, demonstration, evaluation, and communication. We develop a method to identify shadow IT system's and assess their importance with respect to their architectural embeddedness; we demonstrate and evaluate the method's application using a case method (Yin, 2013) in a company from the recycling domain. We spoke with potential target users (IT managers) to judge the method's effectiveness. With respect to the design science literature such interviews represent a legitimate way to evaluate our design science artifact (Hevner et al. 2004; Venable et al. 2012). Although our primary objective is methodological, we embrace the role of design science research for theory building, reflecting on the potential impact of the study on shadow IT research in every stage (cf. Kuechler and Vaishnavi, 2012).

2.1 Defining Objectives of the Method

We argued that not each shadow IT system is an equally high threat: Methodological support should thus enable us to effectively discern shadow IT systems. Our aim is two-fold: First, our method supports IT managers and architects to detect shadow IT systems in an organizations' IS architecture. Second, it facilitates assessing their effects on the application landscape. While methods have been proposed to investigate application landscapes, e.g. to support service-oriented redesign (e.g., Aier and Winter, 2009; Baumann et al., 2009), to assess interface and component complexity (e.g. Schütz et al., 2012) or to enhance business capability support (e.g. Freitag et al., 2011), techniques, measures, and procedures for assessing IS architectures with respect to shadow IT systems are lacking.

2.2 Network-analytic Visualizations and Measures to Evaluate Shadow IT

In the following, network analysis is introduced as a perspective visualizing and assessing application landscapes. Networks consist of a set of nodes and edges. We suggest applications – enterprise resource planning systems (ERP) as well as autonomous billing modules based upon MS Access – to

represent systems interacting with each other. In network analysis terminology, we define them as nodes. To create a network for viable analysis, we need to operationalize these interactions. We could construe a bipartite (also called a two-mode) network (Wasserman and Faust, 1994). In such network, an edge between two applications may be construed if both are used in the same business process. Hence, we would need to assess relevant processes. Our object of research (shadow IT systems) is characterized as being highly independent from central surveillance. Many shadow IT systems will thus not contribute primarily to standardized business process logics. Instead of assessing the systems by using centrally defined processes, we find a one-mode network based upon actual information flows, in forms of integrated interfaces, superior for our analysis. In such a network, edges materialize in implemented and actually used interfaces between two systems (cf. Dreyfus and Iyer, 2008).

Typically we distinguish between different levels of network analysis. (1.) Macro analysis focuses on patterns of interconnections. They are normally visualized graphically or with the help of adjacency matrices (Lerner, 2010: 355f.). One of the most prominent coefficients for macro analysis is the density of a network. It can be determined by the strength and quantity of connections between dyads and triads of interconnected nodes. In an absolutely dense network every node shares a tie with every other (Borgatti and Everett, 1997: 253). Additionally, we may focus on the overall centralization of networks. Centralized networks can be identified by a high number of links that emanate from a few individuals in the core of the network. Relationship analysis (2.) is based on the types of edges and the (non-)existence of relationships. It is highly concerned with cliques, structural holes or 'boundary spanners' (Cross and Prusak, 2002: 9f.). Finally (3.), a micro analysis narrows the scope to the attributes of a single node (Lima, 2007). As mentioned before, we interpret applications as nodes within a network representing the IS architecture. Our main objective is finding and evaluating specific nodes which we claim to represent shadow IT systems. Hence, we will concentrate on the micro level (3.). This level helps us to assess the importance of particular nodes within a network.

In their paper, Dreyfus and Iyer (2008) found that "[a]pplications with high positional value may be important because they influence many other applications". Following suit, we focus on finding metrics for the influence of applications on others. In network analysis, a variety of centrality measures are used to assess a node's influence. We focus on three of them, which are degree, betweenness, and Eigenvector centrality. We chose these metrics by two means. First of all, they are used most often throughout research based upon centrality in network analysis on a micro-level. Secondly, they allow for distinct interpretations. We will evaluate these metrics following the widely accepted SMART criterion for decision processes consisting of five attributes: (s)pecific, (m)easurable, (a)ttainable, (r)ealistic, (t)imely (cf. Doran, 1981; Wright, 2008). (S)pecificity concentrates on a clear target to be improved. Our target is to assess the centrality of (shadow) IT systems. A (m)easurable item offers quantifiable indicators. Metrics are used to define (a)ttainable goals. They need to be as simple as possible to understand clearly their direct implications. Metrics should also clarify their reach, to be used in (r)ealistic decision processes. Finally the (t)ime between data collection and decision need to be minimized. All centrality measures use the same source of network data. Hence, (m)easurability and (t)imely data collection do not differ across them. Both attributes are therefore not used to evaluate the usability of the metrics. They are essential for our further analysis nonetheless. Therefore, we will focus on them in the next chapter, which is concerned with the procedural model. **Degree centrality** defines (actor) centrality on a micro level most simply (cf. Wasserman and Faust, 1994). In a non-directional network, degree centrality (C_D) is measured as the sum of direct ties (x) between one node (i) and any other within the network. Standardized by the size of a network (g), degree centrality is defined in Equation 1 as:

$$C'_D(n_i) = \frac{\sum_j x_{ij}}{g-1} \tag{1}$$

It focuses on the direct neighbors of a node. Hence, although the metric is easy to interpret we can only vaguely assess the influence of such a node on the overall network. A shadow IT system with a degree centrality of zero shares no data with any other application. A high degree of centrality may indicate that the application is integrated into a dense cluster of systems which is strongly interconnected. Nevertheless, it may also hint to a system reaching far between a diversity of systems of the IS architecture and spans boundaries to many different applications. Degree centrality is very comprehensible, as it only counts the amount of used interfaces. (A)ttainable and (r)ealistic decisions could be made on this indicator. For our purpose – to find central applications within an IS architecture – it leaves a lot of room for ambiguous interpretations as it only accounts for direct neighbors. Hence, it lacks the according (s)pecificity.

Another frequently discussed metric is **betweenness centrality**. It is a path-based centrality measure and particularly accounts for indirect ties between nodes (Freeman, 1977; Wasserman and Faust, 1994). Betweenness centrality measures the probability that a node (*i*) lies on a shortest path between two other nodes (*j* and *k*). We sum over the probabilities for every constellation within the network. Betweenness centrality is mostly discussed when it comes to boundary spanning. Nodes with a high betweenness centrality are likely to be the only link between cliques and clusters in a network. Their importance is driven by the fact that if they are removed, the whole network may fall apart. As shown in Equation 2, we also standardize betweenness for the overall network (*g*) as follows:

$$C'_B(n'_i) = \frac{\sum_{j<k} g_{jk(n_i)} / g_{jk}}{(g-1)(g-2)/2} \tag{2}$$

With regards to a visualization of an IS architecture, betweenness centrality may be a good indicator for indirect dependence of applications. We take it as a point of departure for (a)ttainable and (r)ealistic decision making. One may begin with asking the right questions: Does an ERP *A* rely on an application *C* to get data from *B*? If so, application *C* becomes a 'boundary spanner' or 'gatekeeper' (Wasserman and Faust, 1994). While the answers to such questions may be very insightful, the metric itself lacks an important attribute; which degree centrality already contributed for. It does only partly account for the effects of direct links. Even with a low betweenness centrality, the application concerned may have many links to other systems, which by themselves may be interconnected. Moreover, little certainty exists that data streams between applications always take the shortest path, which is a key assumption underlying betweeness centrality.

Finally, we discuss **Eigenvector centrality,** frequently used to discuss the power of a node within a network (Bonacich, 1987). It is also similar to the PageRank used by Google to assess the importance of web pages within the World Wide Web. Eigenvector centrality combines attributes of the two before mentioned approaches (cf. Bonacich, 2007). It does not only take the direct ties of a node into consideration but also the neighborhood of these ties. Additionally, the centrality of a node rises with the centrality of its direct neighbors. The recursive function – referring to Newman (2008) – in standardized form is given by Equation 3:

$$C_\lambda(i) = \frac{1}{\lambda}\sum_j x_{ij} c_\lambda(j) \tag{3}$$

It uses the adjacency matrix *A*, in which $x_{ij} = 1$, if node *i* and *j* are tied to each other. The eigenvalue λ is a constant. Transferred to information systems, we may assume that an application *i*, sharing data with a very central ERP *j*, is also more central in the overall IS architecture. The measure takes direct and indirect connections into account. Therefore, it fulfills our (s)pecific need to find central actors within a network. Due to its recursive definition, the causes of its centrality remain unclear compared to the other metrics. Thereupon, it is rather difficult to define (a)ttainable and (r)ealistic decisions based upon this metric.

In our following analysis, we will therefore use degree centrality and betweenness centrality to assess the importance of a shadow IT system within the IS architecture. We find both measures simple enough to derive direct decisions as well as comprehensive enough to complement each other.

2.3 Procedural Model

In this section, we suggest a procedural model as an additional design artifact. The procedure includes the steps shown in Figure 1: Initiation, data collection, data analysis and conception, and measure implementation.

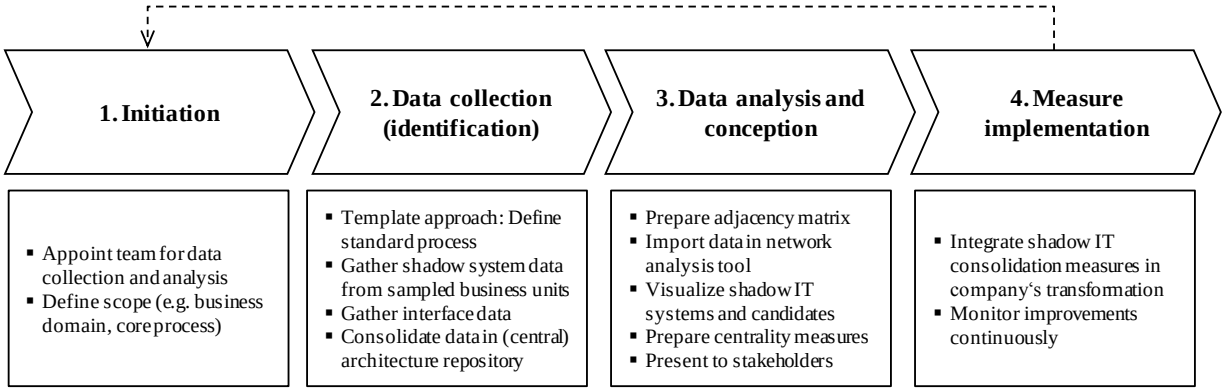

Figure 1. Steps to identify and evaluate shadow IT systems

Project initiation. First, a project team is appointed and the scope is defined. The staffing should account for two kinds of complementary skills: Business analysis competencies are essential throughout the data collection phase to align IT and business perspective; the business analyst acts as a boundary spanner. In later phases, team members need to bring in skills in advanced data analysis, network analysis and statistical modeling. Team members should also be experienced with IS-architectural solution patterns to appraise the data at hand. The project's breadth and depth should be limited; one must keep in mind the resources to collect missing data. The case company, for instance, concentrated its efforts to one business domain with twenty-nine subsidiaries.

Data collection. We found it most important to start with a standard process during data collection. We suggest a template approach; a template supports screening the subsidiaries for potential shadow IT systems by asking: What is the IT support for process step "X"? The team prepares the template together with an experienced business expert. By doing so, the team ensures formulating the template in a comprehensible vocabulary and grounds it in the organization. After a pre-test, a fair fraction of subsidiaries/units should be sampled purposefully. During workshops, the business analyst steps through the standard process with the business units and notes systems supporting each process step. We thereby suggest sticking to the standard process wherever possible while staying reflexive to sense deviations. For each new application, a small sheet with additional characteristics is filled (i.e. description, owner, operator and business units). In a next step, additional workshops with IT staff and architects are performed to map the systems' interfaces. We suggest using a template with source, target, transferred business object, and type of interface (e.g. online, manual, semi-manual). As we suggest storing the data in a central architecture repository, the data is reusable for other projects. As result of the data collection phase, the team should hand over the as-is IT master plan, an application list (as described above), and an interface document for the domain under consideration.

Data analysis. The data analysis team cleans and reconciles the data. This includes removing duplicates, checking the consistency of the data and preparing the adjacency matrix. To identify shadow IT candidates, the analyst should design a set of rules when to classify an application as

shadow IT system. As a starting point, we found it valuable to identify central IT units. Following our definition of shadow IT systems, we categorized each system that is neither owned, operated and used by a central IT unit as a shadow IT candidate. We further distinguished between systems owned, operated and used by the same non-central IT unit and systems for which ownership and usage diverged. We labeled the first set of applications as 'candidates' and the latter 'shadow IT'. This is because some professionalization will be involved when a system is hosted for another unit, which might not be the case for systems thriving in an encapsulated environment. As an optional step, we suggest stepping through the IT master plan for functional analysis; following our definition, shadow IT systems are specific to a business unit's need; hence, these systems are legitimate drivers of innovation if a business unit lacks alternatives on a corporate level. If alternatives exist, however, the situation looks bad as the system overlaps with existing corporate systems. One may thus additionally mark the shadow IT system as redundant. After importing the data to a network analysis tool (e.g. Gephi or *R*), one should start discussing and revising the findings with stakeholders by using network visualizations and measures as discussed earlier. After appreciating the feedback, the team may prepare a concept for coping with shadow IT systems and presents it to stakeholders.

Measure implementation. Measures to consolidate or re-design the architecture should be integrated in the company's overall transformation program. Improvements should be monitored continuously.

3 Recycle Inc.: Demonstration and Evaluation

We use a case method (Yin, 2013) in a private company in the recycling industry to demonstrate and evaluate our method. Recycling Inc. has approximately 9,000 employees and its main areas of business are waste operations, recyclables trading, services, steel and metals recycling among others. Our point of contact was one IT unit in the waste operations business domain, which employed 15 people at the time of our research. There we gained access to a comprehensive data set from the companies' IT architecture group: Data was gathered in a real requirements engineering project during a three-month period in 2011 preparing a major reorganization. The data set mainly contained an overview of the company's current applications, information flows and business processes. The IT master plan gave insights into the business process support for 29 business entities and their 73 business process steps. The core waste management process decomposed into three main steps:

(i) Distributing and pricing waste operations services (e.g. different quality containers)
(ii) Operating and disposing waste including tour planning & weighting and
(iii) Invoicing, accounting and controlling services

We expected to the data to reflect that process but we were surprised of the variety of different applications and information flows. The observed fragmentation led us to perform additional analyses on shadow IT candidates and potential harmful organizational consequences.

In a first step, we ran through a list of approximately 400 applications. The list held the applications' owner, IT operations unit, business units and supplier. Altogether, we had to process 1,650 entries/rows. We defined rules when to mark an application as "shadow IT candidate" (highlighted as yellow nodes in Figure 2, Figure 3 and Figure 4). We concluded to do so if an application had one sole owner, IT operations unit and business unit. An example is an application for gas station terminals that was run, owned and used by Recycle Inc. Building Materials North. We applied the rule only for business units. We did not consider applications run, owned and used by a central IT unit as shadow IT candidates. We identified three central IT units, one on a corporate level and two of which situated within subsidiaries. Next, we filtered applications run and owned by a business unit (not a central IT unit) and used by another business unit as 'potential shadow IT candidates'. Recycle Inc. Cottbus for instance operates a program to plan tours used by Recycle Inc. Service Lusatia and the administrative unit of Recycle Inc. Cottbus among others. We concluded that potential shadow IT candidates must

underlie some kind of governance but often grow in an uncontrolled way as they often do not adhere to central IT standards and architectural principles. The rest of the applications were not marked as shadow IT candidate. In a next step, we brought together the applications and the information flows. As information flows among applications were gathered in a separate data collection step, we had to reconcile the list of applications and the list of information flows. We defined a mapping procedure showing for each application holding information flows whether it was a (potential) shadow IT candidate. As we were interested in applications connected within the application landscape, we excluded applications without information flows from further considerations. We ended with a total of 212 applications of which we identified 24 shadow IT candidates (11%), 101 potential shadow IT candidates (48%) and 87 non-shadow IT systems (41%).

Next, we depicted the data with *Gephi*, a network analysis tool. We plotted applications as nodes and information flows as non-directed and dichotomous edges because we were interested in the degree of integration among different applications. Figure 2 shows that we plotted shadow IT systems in the network visualization in red and shadow IT candidates in yellow. We highlighted *central* shadow IT systems on the network plot by the size of the nodes. We compiled lists of the twenty most central IT systems with respect to degree centrality and betweenness centrality as depicted in Figure 3A and Figure 4D and discussed them with the responsible IT manager. The IT manager used our findings to prepare an IT board presentation on the CIO level coping with the extent of shadow IT in Recycle Inc.'s IS architecture.

We were surprised by the extent to which shadow IT systems embed in Recycle Inc.'s IS architecture. We found a significant fraction of systems in Recycle Inc.'s IS architecture adhering to the category of shadow IT systems (cf. Figure 3A and Figure 4D): Among the twenty most central IT systems, we identified three shadow systems with high degree centrality and four systems with high betweenness.

Example *A*, as depicted in Figure 4, is a tool for tour planning in various subsidiaries in the company's southern region. Tour planning is an important activity in the process landscape of Recycle Inc.; it enables allotting collection vehicles, vehicle personnel, and generating tour plans. Industry-specific standard software traditionally lacks capabilities in this area as tour planning involves integrating complex subtasks as strategic planning, operational planning, and allotment. Tour planning is often complemented by organizational resources in the form of specific planning personnel performing the task on a day-to-day basis. To overcome limitations of existing IT tools, subsidiaries developed an in-house solution together with an external vendor. High betweenness centrality shows the solution connects several parts of Recycle Inc.'s IS architecture. The system acts as a gateway connecting several instances of RANO – an ERP/logistics system used in the south.

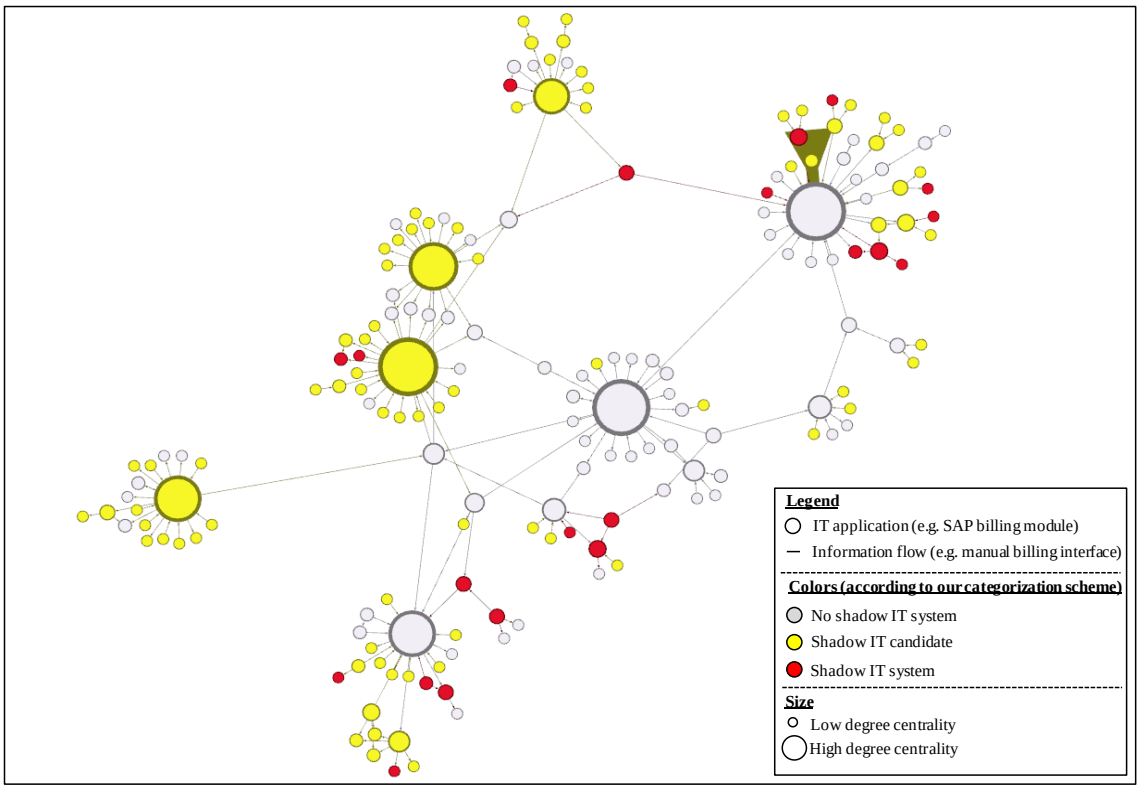

Figure 2. IS architecture of Recycling Inc. including applications and flows of information. The colors indicate whether the application is considered a shadow IT system.

Interestingly, we found results deviating for the two centrality measures: Two additional shadow IT systems appeared in the "top 20" for betweenness centrality that have not been listed when we analyzed degree centrality. We found that both measures deliver complementing information: Degree centrality is a good starting point to determine the importance of shadow IT systems on immediate neighbors and betweenness centrality is vital when it comes to evaluate the bridges between clusters.

Results got amplified when we additionally took shadow IT candidates into consideration (cf. yellow nodes in Figure 2, Figure 3 and Figure 4). We found that some candidates scaled up to the extent of proper hubs with multiple peripheral systems. One example is RANO, mentioned earlier in the analysis. Several instances of the system, e.g. the one depicted in Figure 4B, are used within different subsidiaries in the south, particularly because the system lacks multi-client capabilities; they were added to Recycle Inc.'s IS architecture when the company acquired businesses from a competitor. At the time of our research, the systems had not yet been fully integrated and transferred to the central IT unit's governance. Another typical example for a shadow IT candidate is a dashboard for managerial accounting: In this connection the two large yellow nodes, as depicted in the box within Figure 4C, both represent instances of RANO were the dashboard extracts input data.

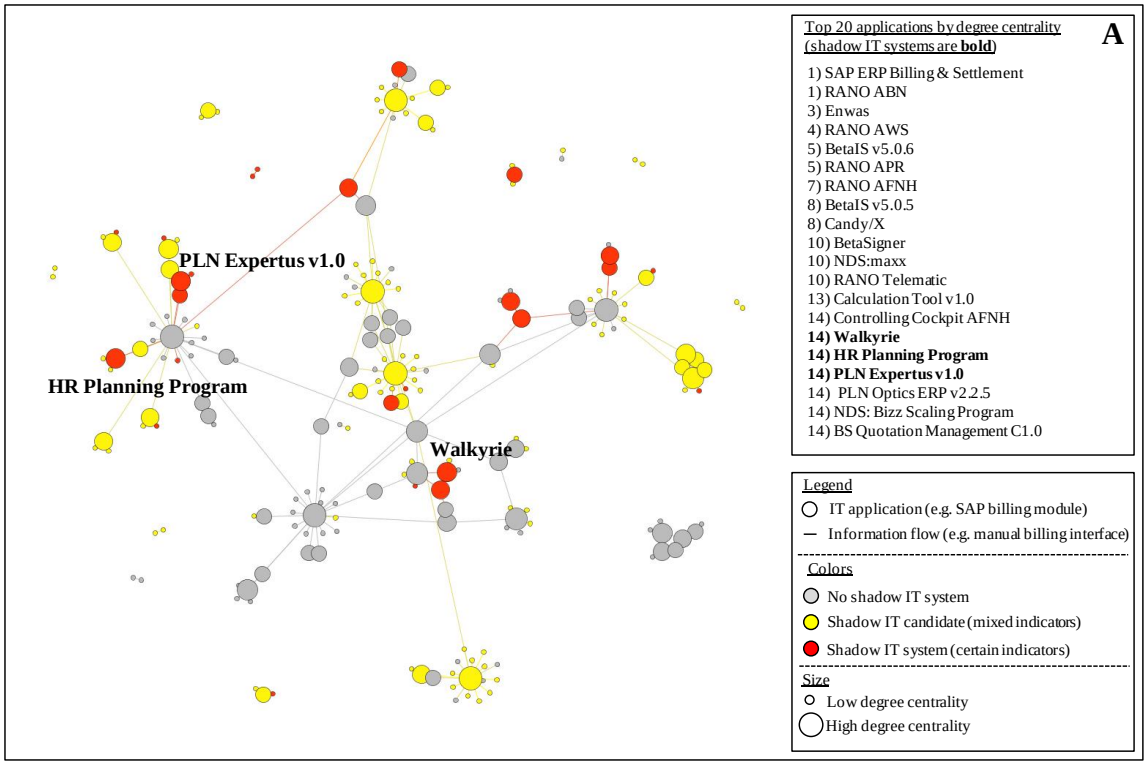

Figure 3. *Recycle Inc.'s IS architecture: The size of the nodes indicates the degree centrality. Shadow IT systems are highlighted yellow (candidates) and red.*

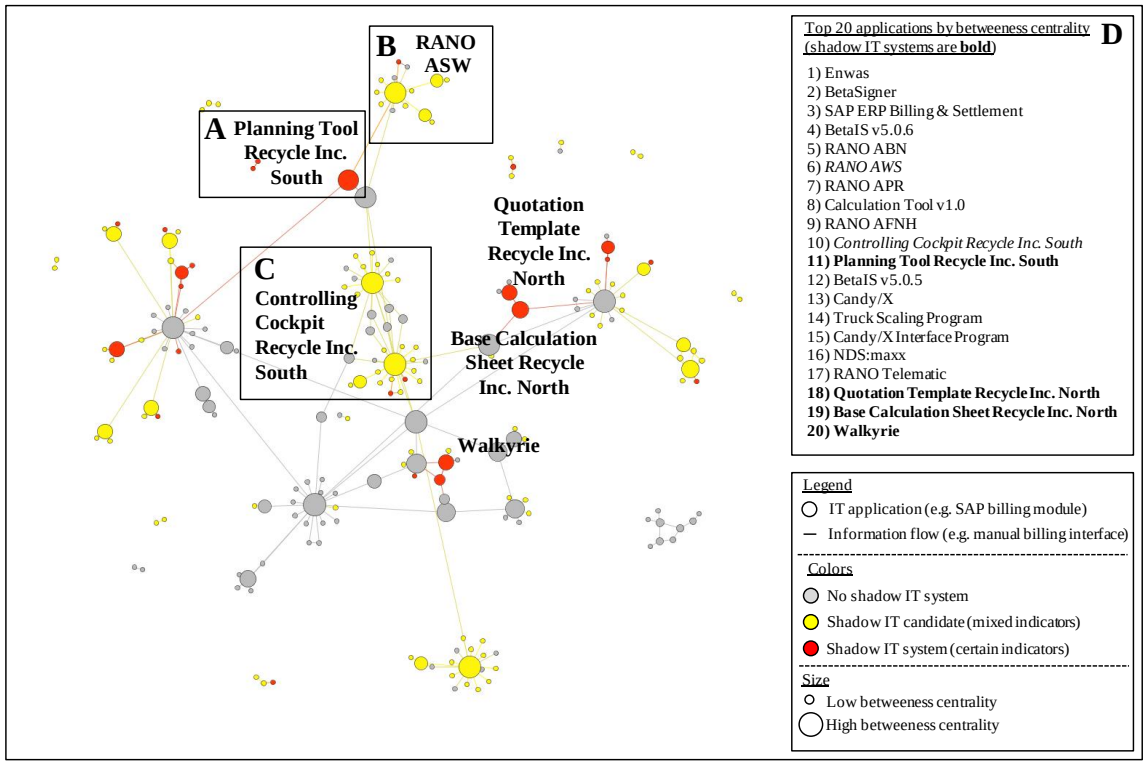

Figure 4. *Recycle Inc.'s IS architecture: The size of the nodes indicates the betweenness centrality. Shadow IT systems are highlighted yellow (candidates) and red.*

4 Concluding Remarks

Our aim in this article has been to advance our understanding on the consequences of shadow IT systems for organizations. Approaches presuming that shadow IT systems are good or evil fall short with respect to their architectural impacts. We accept the multiplex nature of shadow IT systems and discern them by evaluating their position in a network of applications and information flows. To achieve this, we suggest using centrality measures from network analysis. When a shadow IT system is characterized by high betweenness centrality, IT management should be alert: The overall IS architecture is at risk to break down in case the shadow IT system fails to provide its services. Moreover, in case of a high degree centrality, a large number of other systems depend directly on one shadow IT system; if the organization's ability to maintain the system drains or the support is discontinued, organizational measures to replace the system may be drawn back; it is easy to imagine situations in which systems embed so strongly in the organization that serious inertia to discontinue the system arise (e.g. because of difficulties to transfer interfaces or business logics inscribed in the system). Thus, we believe our work is a theoretical step forward towards a more nuanced understanding of shadow IT systems' consequences on a company's IS architecture.

Before sketching practical implications, we emphasize three conditions that limit the generality of our approach. First, data quality issues let us suggest using undirected instead of directed edges. This appears like an appropriate characterization for shadow IT systems acting as a gateway between different parts of the IS architecture; it may, however, mischaracterize shadow IT systems merely extracting data from various source systems. Consider in this connection the dashboard for managerial accounting, depicted in Figure 4C. Even though this shadow system requires appropriate (central) governance controlling its growth, its overall architectural impact may be limited. In connection to the previous point, we earmark the importance of avoiding interlocking; special attention should be devoted to constellations in which systems mutually depend on each other and interfaces are more than one-way streets. Consider for instance the planning tool introduced in Figure 4A: Personnel planning outcomes are fed back to the transactional systems for payroll accounting, timekeeping and other tasks creating strong interdependencies. Limitations of our data, however, prevented us to perform more advanced analysis into that direction. As another limitation, the betweenness centrality measure from network analysis assumes that information between nodes is transferred on shortest paths. Using the measure in other settings should include elucidating whether this fact gives rise to controversies. We, however, tried to mitigate concerns about information bypassing shadow IT systems by discussing our doubts carefully with the respective architects of the company.

Our analysis enables us to draw interesting managerial implications. First and foremost, we conclude that IT managers are better of concentrating on *central* shadow IT systems. By limiting attention to shadow IT systems having reached a critical mass and risk position in the organization, managers are in a better position to obtain the necessary resources and to take appropriate measures how to govern these critical systems. It should be noted that shadow IT systems playing a central role in an IS architecture are very likely to be fundamental for a multitude of business processes. They most likely filled a – previously unaddressed – organizational need. An organization concentrating on such systems will also heave innovative potentials of systems which have already demonstrated to be of effective use. In connection to that point, our approach aims to support managers in their decisions by providing a standardized procedure for detecting and evaluating shadow IT systems. This analysis brings together dispersed organizational knowledge and complements architects' experiences. Finally, we conclude that visualizing shadow IT systems, as demonstrated in our approach, helps to inform stakeholders about the state of the IS architecture in terms of shadow IT systems in a comprehensible way. A dashboard solution could build up on our approach.

In addition to applying our approach in other settings, especially in those in which more centralized IS architectures will be expected (e.g. the banking or airline industry), we see two particularly promising

ways to proceed further. First, the network data we have demonstrated show remarkable traces of legacy, accumulated over long periods. Future studies exploring the evolution of shadow IT systems over time will hence provide interesting challenges. One possible approach would be to collect further network data at different points in time. By connecting different slices of data, one may be able to identify relative movements in the importance of certain shadow IT systems: When and where are shadow IT systems drifting out of control? Another approach to investigate the network dynamics would be to model the underlying growth logic by the means of simulation. The network data we have demonstrated suggests a non-random growth process. We particularly believe it is promising to fit the empirical data to simulation data from non-random (hybrid) growth models as suggested by Jackson and Rogers (2007). This could illuminate whether and to what extent today's central shadow systems become exponentially more critical over time as preferential attachment governs their growth. Second, moving from the level of micro analysis to relationship analysis – as described in our second chapter – would be another next step. We would then focus on architectural patterns, like clusters, of applications. The case data we have presented, suggests that shadow IT systems integrate in the overall IS architecture in different ways: Some shadow IT systems take the form of insular applications coping with specific local challenges, some complement centrally-managed applications in a hub-and-spoke fashion, and in some constellations shadow IT systems even form self-contained ecosystems. This presents interesting challenges for future research.

Acknowledgements

The authors thank Lauri Wessel and Lutz Kirchner for helpful comments on early drafts of this paper. We are also indebted to the Associate Editor and three anonymous ECIS reviewers for their substantial suggestions for improving the former draft of this paper. Daniel Fürstenau gratefully acknowledges funding by the doctoral program "Research on Organizational Paths" (http://www.pfadkolleg.de), which has been generously supported by DFG – German Research Foundation.

References

Aier, S. and Winter, R. (2009). Virtual Decoupling for IT/Business Alignment - Conceptual Foundations, Architecture Design and Implementation Example. Business & Information Systems Engineering, 1 (2), 150–163.

Baumann, A., Engels, G., Hofmann, A., Sauer, S. and Willkomm, J. (2009). A Holistic Software Engineering Method for Service-Oriented Application Landscape Development. In Advances in Enterprise Engineering II (Proper, E., Harmsen, F. and Dietz, J. Eds.), pp. 1–17, Springer, Berlin Heidelberg.

Behrens, S. (2009). Shadow systems: The good, the bad and the ugly. Communications of the ACM, 52 (2), 124–129.

Bonacich, P. (1987). Power and Centrality: A Family of Measures. American Journal of Sociology, 92 (5), 1170–1182.

Bonacich, P. (2007). Some unique properties of eigenvector centrality. Social Networks, 29 (4), 555–564.

Borgatti, S.P. and Everett, M.G. (1997). Network analysis of 2-mode data. Social networks, 19 (3), 243–269.

Cross, R. and Prusak, L. (2002). The people who make organizations go-or stop. Harvard Business Review, 80 (6), 104–112.

Doran, G.T. (1981). There'sa SMART way to write management's goals and objectives. Management Review, 70 (11), 35–36.

Dreyfus, D. and Iyer, B. (2008). Managing architectural emergence: A conceptual model and simulation. Decision Support Systems, 46 (1), 115–127.
Freeman, L.C., 1977. A set of measures of centrality based on betweenness. Sociometry, 40 (1), 35–41.
Freitag, A., Matthes, F., Schulz and C., Nowobilska, A. (2011). A Method for Business Capability Dependency Analysis. In Proceedings of the International Conference on IT-enabled Innovation in Enterprise (ICITIE 2011), paper 9.
Gregor, S. and Hevner, A.R. (2013). Positioning and Presenting Design Science Research for Maximum Impact. Management Information Systems Quarterly, 37 (2), 337-355.
György, A., Cleven, A., Uebernickel, F. and Brenner, W. (2012). Exploring the shadows: IT governance approaches to user-driven innovation. In Proceedings of the 20th European Conference on Information Systems (Pries-Heje, J. et al. Eds.), Barcelona, Spain.
Hevner, A.R., March, S.T., Park, J. and Ram, S. (2004). Design science in information systems research. Management Information Systems Quarterly, 28 (1), 75–105.
Hevner, A. and Chatterjee, S. (2010), Design science research in information systems, In Design Research in Information Systems: Theory and Practice (Integrated Series in Information Systems) (Hevner, A. and Chatterjee, S. Ed.), pp. 9–22, Springer.
Jackson, M.O. and Rogers, B.W. (2007). Meeting Strangers and Friends of Friends: How Random Are Social Networks? The American Economic Review, 97 (3), 890–915.
Jones, D., Behrens, S., Jamieson, K. and Tansley, E. (2004). The rise and fall of a shadow system: Lessons for enterprise system implementation. In ACIS 2004 Proceedings. Paper 96.
Kuechler, W. and Vaishnavi, V. (2012). A Framework for Theory Development in Design Science Research: Multiple Perspectives. Journal of the Association for Information Systems, 13 (6), 395–423.
Lerner, J. (2010). Beziehungsmatrix (in German). In Handbuch Netzwerkforschung (Stegbauer, C. and Häußling, R. Eds.), pp. 355–364, VS Verlag für Sozialwissenschaften, Wiesbaden.
Lima, M. (2007). Visual complexity: Mapping Patterns of Information. Princeton Architectural Press.
Newman, M.E.J. (2008). The mathematics of networks. In The new palgrave encyclopedia of economics (Durlauf, S.N. and Blume, E. Eds.), pp. 1–12, 2nd Edition. Palgrave Macmillan.
Panko, R.R. (2006). Spreadsheets and Sarbanes-Oxley: Regulations, risks, and control frameworks. Communications of the Association for Information Systems, 17 (29), 647–676.
Peffers, K., Tuunanen, T., Rothenberger, M. a. and Chatterjee, S. (2007). A Design Science Research Methodology for Information Systems Research. Journal of Management Information Systems, 24 (3), 45–77.
Raden, N. (2005). Shedding light on shadow IT: Is Excel running your business? Hired Brains Inc., Santa Barbara. Santa Barbara, CA. URL http://isis-solution.com/pdfs/Raiden-Excel%20Running%20Your%20Business.pdf, Last Accessed 21 March 2014.
Rentrop, C. and Zimmermann, S. (2012). Shadow IT-Management and Control of Unofficial IT. In Proceedings of the 6th International Conference on Digital Society (ICDS), pp. 98–102, Valencia, Spain.
Schneberger, S.L. and McLean, E.R. (2003). The complexity cross: implications for practice. Communications of the ACM, 46 (9), 216–225.
Schütz, A., Widjaja, T. and Kaiser, J. (2013). Complexity in Enterprise Architectures - Conceptualization and Introduction of a Measure from a System Theoretic Perspective. In ECIS 2013 Completed Research., Paper 202.
Smyth, K. and Freeman, J. (2007). Blue Prism Rogue IT Survey 2007. URL http://www.blueprism.com/download_file.php?file=dl_13_1_rogue_it_survey_white_paper.pdf, Last Accessed 21 March 2014.
Venable, J., Pries-Heje, J. and Baskerville, R. (2012). A Comprehensive Framework for Evaluation in Design Science Research. In Design Science Research in Information Systems. Advances in

Theory and Practice (Peffers, K., Rothenberger, M. and Kuechler, B. Eds.), pp. 423–438, Springer Berlin Heidelberg.
Wasserman, S. and Faust, K. (1994). Social network analysis: methods and applications, 19th Edition. Cambridge University Press, New York, NY, USA.
Wright, C. (2008). Chapter 7 Policy Issues and Fundamentals. In The IT Regulatory and Standards Compliance Handbook (Wright, C.S. Ed.), pp. 149–159, Syngress, Burlington.
Yin, R.K. (2013). Case Study Research: Design and Methods (Applied Social Research Methods). 5th Edition. Sage Publications.