

Theory-generating design science research

Roman Beck · Sven Weber · Robert Wayne Gregory

Published online: 3 February 2012
© Springer Science+Business Media, LLC 2012

Abstract A frequently mentioned challenge in design science research (DSR) is the generation of novel theory above and beyond information technology artefacts. This article analyzes the DSR process and extends established frameworks for theory generation to exemplify improvements to theory generation through methods of grounded theory development. On a conceptual base, we developed a theory-generating DSR approach which integrates methods of grounded theory development with established DSR methodology. This combination enables a design theorist to generate theoretical knowledge that extends the applicable knowledge base. We do not elaborate this combination on a meta-level, but rather provide a process model for researchers in form of an extension of a well-known DSR model to combine both methods in a pluralistic research design. With this suggested research approach, scholars can draw theoretical insights from analytical abstractions and can improve the development of IT artefacts in a structured way to avoid failure or repair loops.

Keywords Design science research · Behavioural science research · Grounded theory method · Pluralistic research method

1 Introduction

In recent years, design science research (DSR) has attracted more information systems (IS) research attention, because it deals with the generation of information technology (IT) artefacts and their evaluation, which is just as important to IS research as is research into the impacts of IS (Benbasat and Zmund 2003; Hevner et al. 2004). In general, DSR provides models and guidelines to researchers that enable them to create, improve, and evaluate IT artefacts (Hevner et al. 2004; Holmström et al. 2009; March and Smith 1995; Weber et al. 2012). Yet many existing DSR research attempts pay less attention in generating an original theoretical contribution that goes beyond problem-solving IT artefacts (Carlsson 2006; Gregory and Muntermann 2011; Hevner et al. 2004; Winter 2008).

We therefore investigate potential avenues for combining both aims, using a conceptual approach based on an extensive analysis of DSR and design theory literature. As a result of our analysis, we propose a theory-generating DSR approach that is informed by behavioural science elements used for theorizing, such as techniques from the grounded theory method (GTM) (e.g., Glaser 1978; Glaser 1998; Glaser and Strauss 1967). Therefore, our proposed approach closes some methodological gaps in DSR to achieve insights at a higher level of analytical abstraction (Yadav 2010). Certain techniques from GTM are remarkably appropriate, in that they generate additional theoretical insights in DSR. The proposed theory-generating DSR approach combines

R. Beck · S. Weber (✉)
Institute of Information Systems, Goethe University Frankfurt,
Grüneburgplatz 1,
60323 Frankfurt am Main, Germany
e-mail: svweber@wiwi.uni-frankfurt.de

R. Beck
e-mail: rbeck@wiwi.uni-frankfurt.de

R. W. Gregory
University of Göttingen,
Platz der Göttinger Sieben 5,
37073 Göttingen, Germany

R. W. Gregory
e-mail: gregory@wiwi.uni-goettingen.de

elements from DSR and GTM for simultaneous problem solving and theory generation.

Calls for more pluralistic research in IS (e.g., Mingers 2001) have prompted some prior attempts to combine different methods. For example, Goldkuhl (2004) uses behavioural science techniques in a DSR approach and suggests that three types of grounding (internal, empirical, and theoretical) enhance DSR as a means to generate practical knowledge. Holmström et al. (2009) accumulates DSR with a second research cycle that features the development of substantive and formal theory (the focus of GTM) to contribute to the knowledge base, but they also use behavioural science elements to develop an explanatory, theory-testing approach instead of a means to develop new theoretical insights. Kuechler and Vaishnavi (2008a, b) address theory development and theorizing in the context of DSR but without combining any specific behavioural science research method. A more advanced approach is presented by Sein et al. (2011), who integrate DSR with action research in one model. Their model argues that design science needs to go beyond the traditional focus on the IT artefact and recognize its organizational embeddedness. By integrating design science with action research lenses, the authors are able to emphasize the organizational interventions through IT artefact deployment as well as the learning from this interventions in terms of contributions to the knowledge base. In this critical reflection and learning step, which is somewhat disconnected from the traditional build and evaluate DSR cycle, the researchers abstract the key theoretical findings that go beyond solving an individual problem. Finally, some other studies combine DSR with action research (e.g., Allen et al. 2000), ethnography (e.g., Baskerville and Stage 2001), or soft design science (Baskerville et al. 2009) with the goals of improving theoretical abstraction and knowledge generation within a DSR project.

In summary, existing research frameworks guided towards the simultaneous use of DSR and behavioural science but fall short to provide specific processes for how to conduct a multi method DSR project (Carlsson 2006). Yet both Goldkuhl (2004) and Holmström et al. (2009) conclude that DSR and behavioural science may coexist within a single project but rather as an additional and independent step. In this paper, we extend this view and develop a process model for theory-generating DSR that relies on the simultaneous and not separated use of both DSR and GTM elements.

The structure of the remainder of this article follows the building blocks of conceptual research described by Yadav (2010, p. 3). In the next two sections, we provide an overview of the two research approaches (DSR and GTM) before we outline how design theory components map onto the DSR approach. In the subsequent section, we present a detailed description of the theory-generating DSR approach, followed by an overview on theory-generating DSR. We

also detail how the IS design theory components relate to our proposed approach and provide a first illustration how theory-generating DSR can be applied in DSR projects. We conclude with a discussion section and an outlook for further research.

2 Design science research

The primary focus of IS research is the IT artefact (Benbasat and Zmud 2003), whereas DSR centres on the design and creation of the artificial (Simon 1969), especially IT artefacts (in the form of a construct, model, method, instantiation, or combination thereof; (March and Smith 1995)). Therefore, DSR encompasses the creation of an innovative construct that has not existed before and can be used to serve human purposes (March and Smith 1995). Its historical origins stem from engineering (Au 2001; March and Smith 1995), but DSR also relates to several other academic disciplines, such as architectural science and computer science. A common ground across these disciplines is a dual focus on both the practical application of the IT artefact and the scientific abstraction and learning (Baskerville 2008). In this context, theorizing in DSR is worthy of discussion and at least as important as problem-solving itself (Lee et al. 2011).

Most DSR researchers also recognize that the developed IT artefact can provide theoretical contributions, provided that key DSR guidelines are followed (Gregor 2002, 2006) and so-called kernel theories, drawn from the natural and social sciences through a creative translation process, are applied (Hevner et al. 2004; Markus et al. 2002; Walls et al. 1992). Derived principles and concepts of such kernel theories may provide a basis for advancing DSR by specifying requirements and generating an IT artefact (Walls et al. 1992). However, Gregor and Jones (2007) explore that theorization is a key goal in theory-generating DSR which may eventually lead to an IS design theory. Therefore, problem solutions and knowledge contributions can be derived from either existing kernel theories and IT artefacts or the development of a new IT artefact and subsequent theorization (Holmström et al. 2009).

In the DSR process, the initial step must be the search for a problem that has practical relevance (Hevner et al. 2004). In other words, “a DSR approach seeks a solution to a real-world problem of interest to practice” (Kuechler and Vaishnavi 2008a, p. 492), which requires a differentiation between products (IT artefact) and processes (activities that lead to the IT artefact) in the DSR cycle (March and Smith 1995; Walls et al. 1992). The product outcome is inevitably embedded in some place, time, and community and must undergo theorization to meet innovative and progressive demands (Orlikowski and Iacono 2001).

The DSR process in turn consists of two basic processes: building and then evaluating the IT artefact (Baskerville et al. 2009; Hevner and March 2003; March and Smith 1995). In the building process, a sequence of activities aims to produce “something new,” then in the evaluation process, the created IT artefact undergoes evaluation to produce feedback and generate new knowledge about the problem. The newly generated insights serve to improve the quality of the IT artefact and the design process itself (Hevner et al. 2004). These processes take place partly in parallel and involve multiple iterations, which enables the IT artefact to be generated such that it fully satisfies the researchers and practitioners who later make use of it (Markus et al. 2002). In total then, DSR offers a rigorous and meaningful contribution to practice and theory, in the form of an IT artefact and its evaluation (Gregor 2006).

3 Grounded theory method

The supporting component for theory-generation in our model is GTM. Since it was introduced to sociology by Glaser and Strauss (1967), the method has been widely developed and applied in various disciplines (Bryant and Charmaz 2007a; Urquhart 2007). In a grounded theory study, the focus lies on the discovery or generation of theory grounded on empirical evidence (Glaser and Strauss 1967). A grounded theory study considers the process of discovering concepts and categories and the relationships between them (Bryant and Charmaz 2007b), with the end result of a substantive or grounded theory (Glaser 2007).

Over time, GTM has evolved in different directions. For example, in the mixed methods approach, GTM combines with other techniques and methods, such as case study research (Eisenhardt 1989; Eisenhardt and Graebner 2007). The most predominant approaches are those propagated by the originators, Glaser and Strauss, who offer “Glaserian” and “Strausserian” forms of grounded theory (Ketokivi and Mantere 2010). This distinction also has been called the emerging versus forcing debate. The proponents of Glaserian grounded theory place more emphasis on the principle of emergence; grounded theory should emerge from the data and existing theories or concepts, and coding schemes should not be forced onto the data (Udo 2005).

The term “grounded theory” refers to both the end product and the process. Conducting grounded theory research involves various techniques prescribed by GTM (Glaser 1998; Strauss and Corbin 1990), including—among others—theoretical sampling and the constant comparative method (Suddaby 2006). Theoretical sampling means that insights from the initial data collection and analysis guide subsequent data collection and analysis, so the grounded theory can emerge over time through iterative cycles of

deeply intertwined data collection and analysis. Urquhart et al. (2010) provide an in-depth discussion of this key grounded theory concept and refer to this as deciding on analytic grounds where to sample from next. Over time, researchers achieve theoretical saturation because additional data collection and analysis efforts do not yield new findings (Eisenhardt 1989). In the constant comparative method, the researcher constantly compares instances of data labelled as a particular category with other instances of data in the same category as a means to substantiate these categories and build theory. This method is discussed at length by Urquhart et al. (2010) and the authors refer to this as an important tool for exposing generated analytical insights to rigorous scrutiny and building theory. Thereby, all kinds of “slices” of data (e.g., primary data such as qualitative interviews, secondary data such as documentation and extant literature) are used to reach higher levels of abstraction and advance conceptualization. The relations identified among the categories and the theoretical integration lead to the grounded theory.

Grounded theory method can in principle be used within a wide range of epistemological stances and research approaches (Bryant and Charmaz 2007a; Madill et al. 2000). It provides the ability to generate an in-depth understanding of a phenomena by analyzing qualitatively the collected data from different sources, sorting it into consistent categories, and emerging a grounded insights of it (Urquhart et al. 2010). For an overview of the process of grounded theory development, we recommend, for example, Fernandez (2004). In the past, Weedman (2008) used this multi-epistemological nature of GTM to analyze an IT artefact design project (called Sequoia) with elements of GTM. While GTM is never mentioned, her analyzing techniques were very similar to theoretical sampling and the constant comparison method. In addition, Baskerville and Pries-Heje (1999) combined action research with grounded theory to add rigor and reliability to the theory formulation process of action research. As an contrary example, Arazy et al. (2010) used qualitative methods to evaluate and test a social recommender systems that utilizes data regarding users’ social relationships by filtering relevant information to users. However, this approach is more on theory testing and not building. Hence, GTM, as a behavioural method provides suitable instruments for an in-depth understanding of the problem space and its environmental factors to theorize and not only test in DSR.

4 IS design theory and design science research

On the one hand, DSR provides scholarly contributions based on the design, evaluation, generalization, and/or theorization of the IT artefact (Gregor 2006; Gregor and Jones 2007; Orlikowski and Iacono 2001). On the other hand, it provides a practical contribution to practitioners in the form of the useful

IT artefact (Hevner et al. 2004). Hence, both the IS design theories as well as extant DSR models have to be taken into account to develop a new theory-generating DSR approach.

4.1 IS design theory

As a theoretical lens, our literature review focused on fundamental components of IS design theory. Our initial DSR understanding pushed us toward Walls et al. (1992) who laid the basic foundations for developing a precise understanding about the nature and anatomy of a design theory. They developed seven components of an IS design theory guided by prior literature (e.g., Dubin 1978; Nagel 1961) and separated them into product and process components.

The product components encompass meta-requirements, meta-design, existing kernel theories governing the requirements, and a test of whether the meta-design satisfies the meta-requirements. Most existing DSR publications spend little time considering the requirements or assume they are already specified (e.g., Aalst and Kumar 2003; Abbasi and

Chen 2008; Umapathy et al. 2008; for further information please see the Appendix). Because requirement specification is a central part of developing an IT artefact, as well as of subsequent satisfaction with its functionality, we regard this requirements specification phase critical for DSR. In addition, a lot of information becomes available from a system when we consider its purpose, the problem it aims to solve, and the overall intentions behind building it. Therefore, we searched for an appropriate theoretical lens to integrate requirements into our theory-generating DSR approach.

The process components of Walls et al. (1992) represent the design method for the IT artefact and integrate existing kernel theories that govern the design process. They also describe the underlying knowledge for the design process and guide research projects.

In turn, using the components of design theory identified by Walls et al. (1992), Gregor and Jones (2007) suggest eight components that an IS design theory should include. We build on these eight components to analyze existing

Table 1 Eight components of IS design theory in different DSR approaches

IS Design Theory Component	Description	Comparison with Existing DSR Frameworks	Improvements Required for Theory-Generating DSR
1. Purpose and scope	“What the system is for,” that is, the set of meta-requirements that specifies the type of artefact to which the theory applies and also defines the scope or boundaries of theory	Meta-requirements (Walls et al. 1992)	A more detailed requirements analysis increases our fundamental understanding of the IT artefact and its sub-parts
2. Constructs	Representations of entities of interest in the theory (design sub-parts)	Partly covered by the design cycle: build and evaluate (March and Smith 1995)	
3. Principle of form and function	The abstract blueprint or architecture that describes an IT artefact, whether a product or method/intervention	Different frameworks provide guidelines and recommendations for the design process (e.g., Hevner et al. 2004)	
4. Artefact mutability	Changes in the state of the artefact anticipated by theory; that is, what degree of artefact change is encompassed by theory?	Partly covered by theorizing about the IT artefact (Orlikowski and Iacono 2001)	Detailed theoretical analysis of the IT artefact and its design process to adapt to changing environments and allow for evolution over time
5. Testable propositions	Truth statements about the design theory	Evaluation of whether the design satisfies requirements (Markus et al. 2002)	Evaluation and theory generation through theory building from behavioural science (e.g., Holmström et al. 2009)
6. Justification knowledge	The underlying knowledge or theory from natural or social sciences that gives a basis and explanation for the design (kernel theories)	Underlying kernel theories (Walls et al. 1992)	Extend underlying kernel theories with behavioural science aspects (Gregor and Jones 2007)
7. Principles of implementation	Description of processes for implementing the theory (either product or method) in specific contexts	Partly covered by the people involved in a given or changing environment (Hevner et al. 2004)	A detailed analysis of influential factors (e.g., interaction between involved people)
8. Expository instantiation	A physical implementation of the IT artefact that can help represent the theory as an expository device and for the purposes of testing	A viable IT artefact in the form of a construct, model, method, or instantiation (Hevner et al. 2004)	

DSR frameworks and determine if they include all the components needed to provide theoretical insights (see Table 1). In the first and second columns of Table 1, we outline each required design theory component and then provide information about existing DSR frameworks. Finally, we illustrate ways to overcome the challenges we have identified to generate additional theoretical insights from DSR. Although many IS design theory components have been addressed, others might be improved or strengthened by behavioural science elements, particularly to formulate and evaluate the theoretical insights derived from IT artefact development and use.

4.2 Extant DSR models

To develop our theory-generating DSR approach, we searched for a established DSR model that could serve as a basis for our approach. Several scholarly contributions focus on the epistemological positioning of DSR. For example, Winter (2008) differentiates design science from design research: The former encompasses reflection on and guidance for the IT artefact construction and evaluation process while the latter encompasses the creation and evaluation of a specific IT artefact. Hevner et al. (2004) focus on the process of IT artefact development. Thus, IS research appears influenced by both the environment (people, organizations, existing technology) and the knowledge base (Hevner et al. 2004). This classification is closely related to Winter’s (2008) point of view where an IT artefact results from the constant iteration of refinement and assessment. In this regard, Hevner et al. (2004), suggest seven guidelines for conducting DSR, but many other examples of DSR frameworks and models exist, including Peffers et al. (2008), Pries-Heje and Baskerville (2008), March and Smith (1995), and Nunamaker et al. (1991). These DSR frameworks build the foundation for many researchers conducting DSR but do not provide explicit prescriptions how to achieve theoretical abstraction from an IT artefact in a structured way.

A somewhat different approach is provided by Vaishnavi and Kuechler (2008) as well as Kuechler and Vaishnavi (2008a, b) who propose a design cycle for DSR that address theory development and theorizing. Moreover, they provide a general model describing each process step in DSR as illustrated in Fig. 1.

According to the model of Vaishnavi and Kuechler (2008) as well as Kuechler and Vaishnavi (2008a, b), problem awareness is the starting point of DSR which is reflected by an initial proposal depicting the problem that has to be solved. This step is similar to Hevner et al. (2004) and their demand for an initial problem that has to be solved. The next phase is the suggestion phase where it is tested if the formulated proposal can be transferred into a tentative

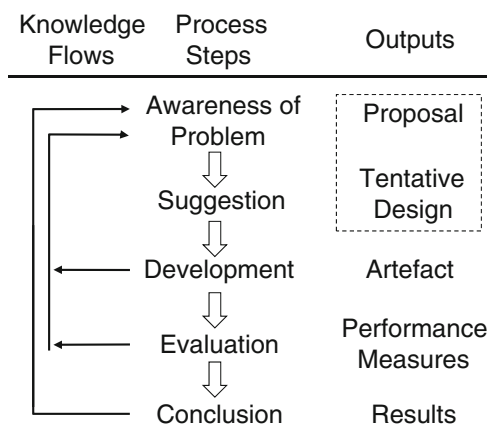


Fig. 1 General design cycle of DSR (Kuechler and Vaishnavi 2008a, b; Vaishnavi and Kuechler 2008)

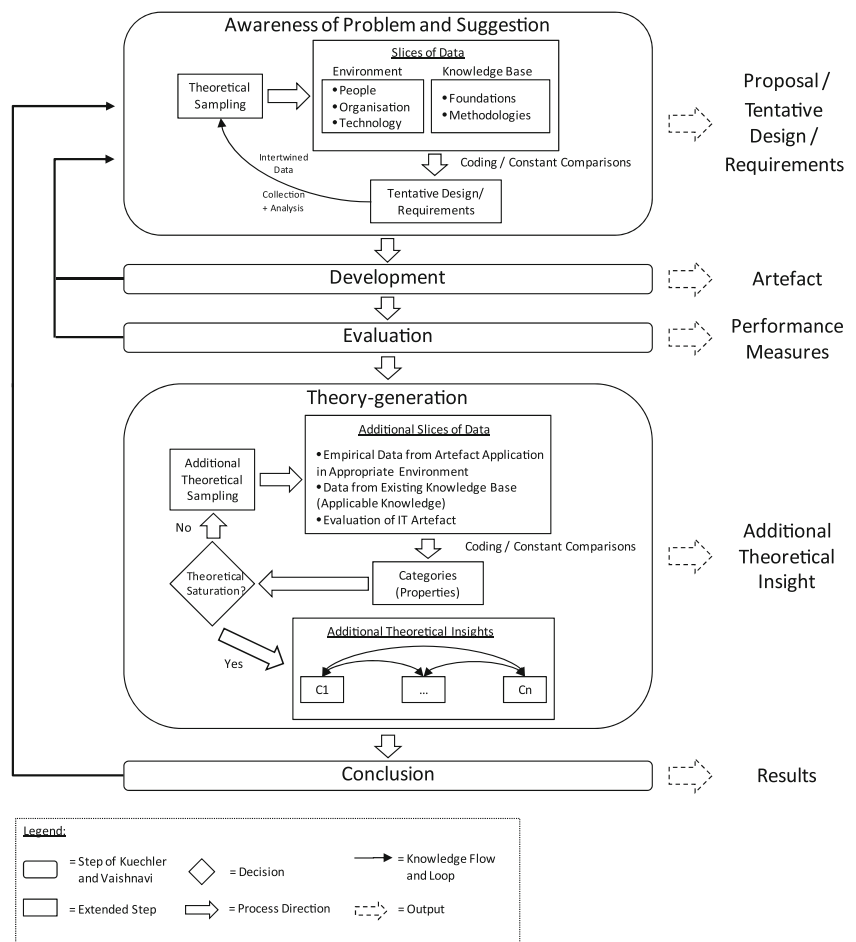
design or not. Thereafter, the IT artefact is developed and evaluated and conclusions are drawn from the IT artefact as a result of the problem solving process. However, the whole process is highly repetitive in the sense that every process step can lead to a repetition and improvement of prior steps through the knowledge flows depicted in Fig. 1.

Departing from the described DSR model, we used its basic elements to structure and develop our theory-generating DSR approach which incorporates the findings from Table 1, as illustrated in the following.

5 Theory-generating design science research

On the basis of prior findings and identified components in DSR design theory development, as well as the ways they might be improved with behavioural science elements, we developed our theory-generating DSR approach, which combines DSR and GTM techniques, as illustrated in Fig. 2. The proposed approach extends the general design cycle of DSR from Kuechler and Vaishnavi (2008a, b). Therefore, the first steps encompass typical methods used already in DSR: the awareness of the problem as well as to develop and evaluate the IT artefact. These steps are complemented and specified by supportive GTM techniques, such as detailed analyses of the tentative design and requirements, theoretical sampling, and the constant comparative technique. A subsequent theory-generation step which is an extension of Kuechler and Vaishnavi’s (2008a, b) model encompass the simultaneous use of DSR and GTM to generate additional theoretical insights, such as by collecting additional data from the application of the IT artefact in an appropriate environment. A complementary research approach can create new insights and results in the conclusion step to be integrated into existing knowledge, in addition to the developed IT artefact. The loop back to the awareness of the problem to refine the tentative design

Fig. 2 Process of generating additional scholarly contributions by theorizing in DSR



and the requirements therefore closes. Moreover, these new insights and knowledge provide a theoretical basis for upcoming DSR projects.

In the following sections, we discuss in detail the different steps of our proposed theory-generating DSR approach (Fig. 2).

5.1 Awareness of problem and suggestion

The awareness of the problem and the initial definition of the research lens is critical, in the sense that the underlying theories of different research approaches (e.g., design or behavioural science) justify and explain the researcher's design process (Gregor and Jones 2007; Kuechler and Vaishnavi 2008a, b). Unfortunately, serious methodological problems can occur in this step, for example, a researcher with a strict GTM lens could primarily focus on exploring the theoretical implications behind the requirements for the IT artefact and pay less attention to problem-solving. In contrast, a DSR researcher might deemphasize important behavioural issues by focusing only on the actual problem. Theory-generating DSR therefore tries to address both practical relevance and theoretical contributions. In contrast to Kuechler and Vaishnavi's (2008a, b) model, we merged the

awareness of the problem and the suggestion in one step. As they mention themselves, both complement each other and are highly intertwined (Kuechler and Vaishnavi 2008a, b). Hence, both can be combined with our theory-generating approach in one step to explore the requirements of the IT artefact.

5.1.1 Theoretical sampling

A first tentative design of the IT artefact and well-defined requirements are key for problem solving, so this step is heavily influenced by the real-world problem as it tries to determine the proper requirements and entities of interest with regard to the IT artefact (Gregor and Jones 2007). Researchers therefore need to focus on the actual problem, but they cannot lose perspective on potentially emergent problems and must actively integrate solutions into the development of the IT artefact. By focusing on both goals simultaneously, researchers can establish a basis for a satisfying IT artefact. Theoretical sampling in GTM means that insights from initial data collection and analysis efforts guide subsequent ones. In other words, understanding of the phenomena (or business requirements) emerges over time through iterative cycles of data collection and analysis

that are deeply intertwined (Glaser 1978). Urquhart et al. (2010) define theoretical sampling as “deciding on analytic grounds where to sample from next” (p. 371). Thereby, theoretical sampling assists the classification of data, the characterization of relationships between data, and to clarify these relationships. Without this method, it is nearly impossible for the researcher to generate theoretical insights about the phenomenon (Urquhart et al. 2010). Accordingly, subsequent data collection and analysis efforts over time must build upon one another for cumulative generation of theoretical insights. Over time, researchers reach a kind of saturation, such that additional data collection and analysis efforts do not yield any new findings (Eisenhardt 1989). Moreover, theoretical sampling increases the researchers’ flexible ability to look for new aspects that emerge during the process. It also provides a means to consider potential problems with the IT artefact in advance, thus enhancing the likelihood of creating innovative and progressive IT artefacts for testing.

5.1.2 Slices of data

The slices of data to be collected and analyzed relate to the environment, including people, the organization, and technology, as well as the knowledge base or extant literature (Hevner et al. 2004). However, in DSR, slices of data mainly refer to extant literature or models (e.g., Aalst and Kumar 2003), collected data (e.g., Albert et al. 2004), or kernel theories (e.g., Markus et al. 2002) (for a detailed discussion, see the Appendix). The inclusion of extant literature alongside the empirical data is a technique widely adopted by IS grounded theorists (e.g., Fernandez 2004; Levina and Vaast 2005) as a means to raise the overall analysis to a higher conceptual level. In addition, the process of systematic data collection and analysis helps to identify and clarify the requirements and entities of interest (Gregor and Jones 2007), which provide the basis for the subsequent development or adaptation of an IT artefact. This process is inherently iterative, in that data collection and analysis on the one hand and IT artefact creation and evaluation on the other hand are deeply intertwined.

5.1.3 Tentative design and requirements

A first tentative design and well-defined requirements are extremely important to DSR, because they ensure the IT artefact’s relevance to a real-world problem (Gregor and Jones 2007; Hevner et al. 2004; Pries-Heje and Baskerville 2008; Walls et al. 1992). In our theory-generating DSR approach, these requirements and entities emerge from the intertwined process of theoretical sampling and the collection of slices of data. Most DSR projects integrate the IT artefact’s requirements as an essential step (e.g., Aalst and Kumar 2003; Albert et al. 2004; Markus et al. 2002; for a

detailed discussion please see the Appendix), because without clearly defined requirements, the IT artefact will not be useful and cannot offer a satisfying solution.

5.2 Development

The tentative design and the explored requirements lead to the development and creation of the IT artefact which is the outcome of this step (Kuechler and Vaishnavi 2008a, b). Because DSR has its roots in engineering and the science of the artificial (Au 2001; Baskerville 2008; Hevner et al. 2004; McKay and Marshall 2005), the creation and evaluation of the IT artefact are frequently at the focus of researcher’s attention. They must exist in any theory-generating DSR approach. In addition, DSR provides additional theoretical contributions by representing both an expository device and the purpose of the testing (Gregor and Jones 2007), as we describe next.

5.3 Evaluation

In the evaluation step, the researcher must evaluate if the IT artefact meets the requirements and solves the real-world problem (Gregor and Jones 2007), because “What works and doesn’t work will evolve over time based upon feedback and learning from applying the ideas and analyzing the results” (Basili 1996). This step tests if the criteria that were explored in the awareness of the problem and suggestion step are accomplished. The outcome of this step is represented by performance measures for the IT artefact (Kuechler and Vaishnavi 2008a, b). Both the creation and evaluation of the IT artefact are highly intertwined. Without a problem solution, the preceding steps must be repeated, until the evaluation actually indicates a satisfactory problem solution. Thus, theory-generating DSR includes a refinement cycle of data collection, development, and creation. For traditional approaches, a solution signals the end (Aalst and Kumar 2003; Umaphy et al. 2008), such that the contribution to the knowledge base is the development and evaluation of the IT artefact (Gregor 2006; Hevner et al. 2004). However, theory-generating DSR goes beyond that border.

5.4 Theory-generation

Beside the simultaneous use of GTM and DSR in the awareness and suggestion step, theory-generating DSR enables the researcher to conduct an additional theory-generation step. In our model, this step represents an extension of Kuechler and Vaishnavi’s (2008a, b). It enables additional theoretical insights, beyond the developed IT artefact and therefore represents an intertwined process of problem solving and theorizing. In particular, these additional insights present the outcomes of this extension. It also

uncovers the potential of GTM to offer valuable research advice, e.g., techniques such as theoretical sampling and constant comparative method, to a growing area of interest in IS research that focuses on the IT artefact, as demanded by leading scholars (e.g., Hevner et al. 2004; Orlikowski and Iacono 2001). GTM focuses on generating a behavioural understanding of the focal phenomenon and making a theoretical contribution to the knowledge base, whereby in the prior steps, DSR solved a real-world problem and developed the basis for this analysis; the IT artefact.

While GTM enables DSR to offer an additional theoretical contribution to the knowledge base beyond the IT artefact (Hevner et al. 2004), DSR enables GTM to bring the IT artefact to the centre of scholarly attention (Orlikowski and Iacono 2001).

5.4.1 Additional theoretical sampling

Several additional steps provide new insights about the usage and performance of the IT artefact. A created IT artefact can change users' behaviours and expectations, which would define new requirements for follow-up projects and improvements of the IT artefact. In this sense, the approach has created a reusable contribution to the knowledge base. Weedman (2008) thus describes a DSR project focusing on the evaluation and theory-generating part supported by GTM techniques.

Additional theoretical sampling involves further data collection, after the creation of the IT artefact, with support from GTM. These additional data enable the researchers to explore the performance, usability, and assimilation of the IT artefact. This step increases understanding of the usage of the IT artefact and provides a means to explore changes in people's behaviour after they use the IT artefact.

Again, the data collection and analysis guide any subsequent data collection. Therefore, the understanding of the phenomena, the usage of the IT artefact, and its impact on people's behaviour emerge over time, through iterative cycles of data collection and analysis (Glaser 1978). Over time, the researcher achieves theoretical saturation (Eisenhardt 1989). This extremely important step in theory-generating DSR bridges the gap between design and behavioural sciences (Holmström et al. 2009; Orlikowski and Iacono 2001). Some prior research used additional theoretical sampling but unfortunately only to evaluate the created IT artefact, without clarifying the connection between the approaches (e.g., Albert et al. 2004; Markus et al. 2002).

5.4.2 Additional slices of data

Additional data may come from, though are not limited to, the application of the IT artefact in the appropriate environment, applicable knowledge from existing literature, and

results from the IT artefact evaluation. The overall analysis moves up to a higher conceptual level (Levina and Vaast 2005). Furthermore, DSR provides unique data that cannot be used in an ordinary GTM approach, in the form of so-called throw-away prototypes (e.g., Markus et al. 2002) and data from IT artefact testing. Not only does the usage of an IT artefact depend on its environment, but IT artefacts explicitly can change their environment and people's behaviours and thus the requirements for future IT artefacts. However, throw-away prototypes offer only instantiations as IT artefacts. Thus, in theory-generating DSR, additional slices of data can contain more than literature, including prototypes and other innovative data.

5.4.3 Categories

An essential step that condenses core categories during the analysis involves the structured process of coding and analysis (Glaser 1978). Different models for coding exist, but in the Glaserian version, the researcher starts with so-called open coding, such that he or she groups indicators from the data or initial IT artefacts into concepts and then categories over time, when it becomes clear which themes are of central interest to yield the desired theoretical insights. The coding process itself evolves and changes over time to become more selective, such that prior conceptualizations and codes guide subsequent steps. To generate these categories (and their properties) from the collected data, researchers undergo several iterations of coding and analysis and employ constant comparison (Glaser 1978). In this sense, constant comparison is the process of constantly comparing instances of slices of data with other instances of data to generate the categories. This comparison can also be applied to define and refine existing categories. It contributes to the development of theory by structuring the analytic properties of the data and categories (Urquhart et al. 2010). For instance, Markus et al. (2002) use throw-away prototypes as sorted categories, refined over such iteration steps. Thus, researchers can reveal any clear fit of the IT artefacts with primary or secondary data, including people's changed behaviours. The integration of extant literature into the coding and conceptualization process can enrich the process of creating and exploring categories. As a result, the researcher must not "force" additional theoretical insights onto the data. The additional data collection and analysis continues until a point of theoretical saturation. This point is reached when the researcher believes further data will not lead to any more insights.

5.4.4 Additional theoretical insight

Condensing the emerged categories allows for analyses of the relationships among them and thus increases insights (Glaser 1978). The end result, such as an added contribution

to the domain of study in the form of grounded theory about the developed IT artefact, expands the knowledge base (e.g., Weedman (2008) contributes to social science theory). After identifying and defining the core categories, researchers must assess the relationships among them to generate additional theoretical insights. An important step to achieve the final theoretical contribution of this study entails extensive comparisons across the generated insights and prior published work in the same domain. The emergent contribution and its theoretical insights then can be integrated into follow-up DSR projects as new requirements to consider.

5.5 Conclusion

To encourage a cumulative research tradition and benefit from existing problem solutions, the results of the proposed research approach should be used in further projects or research cycles, integrated as an existing knowledge base and slice of data in initial iterations.

In summary, theory-generating DSR integrates simultaneously techniques from DSR and GTM to construct an IT artefact and undertake an evaluation through conceptualizations and theory building (Hevner et al. 2004; Winter 2008). Prior DSR research has contributed to the knowledge base with designs and actions affiliated with the IT artefact itself (Gregor 2006). We see further potential derived from the design and action processes. Our intertwined theory-generating approach provides a process model for such research, though the sequence of the procedure naturally must be defined by the researcher and his or her intentions. That is, theory-generating DSR is highly dynamic and should lead to a mutable IT artefact that can be easily adapted to changing environments (Gregor and Jones 2007).

6 Evaluation of our approach to theory-generating design science research

To evaluate the developed theory-generating DSR approach, we present first a theoretical evaluation in which we describe how the developed model fits to the IS design theory components of Gregor and Jones (2007) and second, illustrate with the help of an exemplary case of the extant DSR literature how some of the theory-generating DSR elements can be applied.

6.1 Theoretical evaluation

In Table 2 we provide a comparison of the proposed steps and activities against the eight IS design theory components provided by Gregor and Jones (2007). Beyond descriptions of our developed theory-generating DSR approach, we offer information about the activity in and legitimation for each

step. Finally, we illustrate how the IS design theory components of Gregor and Jones (2007) relate to the steps of theory-generating DSR.

The missing design theory component not addressed by our theory-generating DSR approach is component 4 from Gregor and Jones (2007): artefact mutability. Despite this apparent gap, we believe the overall process of generating additional theoretical insights from DSR finding leads to a more mutable IT artefact that can be adapted easily to changing environments. Furthermore, the detailed analysis of the IT artefact's requirements and its functionality in a given or changing environment makes the IT artefact more accountable, flexible, and mutable; it is not necessarily embedded in some specific place, time, or community (Orlikowski and Iacono 2001). Therefore, though our approach does not explicitly deal with IT artefact mutability, which may be regarded as a limitation, we believe it addresses artefact mutability implicitly, through the process of iterating IT artefact cycles (problem solving, theoretical saturation) as the IT artefact goes through repeated cycles of change and reflections. However, because our theory-generating DSR approach aims to achieve new insights, testing artefact mutability with an upfront chosen kernel theory admittedly is not possible.

6.2 Illustration of our model through an exemplary case

Weedman (2008) analyzed a design project from a behavioural and social science perspective. This design project was about the implementation of 'Sequoia 2000' which represents an interactive information system based on the data-handling needs in global change science. Earth scientists need to analyze a huge mass of data of different types, e.g., satellite data, text, raster, and vector with Sequoia 2000 as the platform that allows to analyze and share the data among them. The development of Sequoia 2000 can be used to illustrate several steps of our model even though they are not explicitly followed by Weedman (2008).

Weedman (2008) entered the field and positioned their research among others on the design theories of Gregor and Jones (2007). She conducted an extensive literature review on DSR and design theories to *be informed (awareness) of the problem*. However, their problem was grounded in the need for a collaboration and analysis tool for earth scientists. The *requirements* for the huge data-handling problem were officially stated as: vastly increased storage, much faster networking, visualization techniques for modelling, and a new database management system to replace the existing flat file system. The expectation towards the system was that it would change the way how global change researchers work, for example by allowing much more data transfer than in the past based on increased storage and network speed. However, initial *theoretical sampling*, such as

Table 2 Eight components of IS design theory in different DSR approaches

Step	Derived from	Activity	Reason	Supported Design Theory Components
Awareness of problem and suggestion	DSR (Kuechler and Vaishnavi 2008a, b; Peffers et al. 2008)	Definition of research lens	Need for appropriate combination of design and behavioural science	Component 6: Underlying theories to justify and explain the design
Theoretical sampling	GTM (Glaser 1978)	Systematic collection and analysis of data	Provides the means to identify requirements and a basis for generating theory from an IT artefact	Component 2: Through a detailed analysis, the entities of interest can be derived and clearly communicated
Slices of data	GTM (Fernandez 2004; Levina and Vaast 2005)	Identification of necessary slices of data for both DSR and behavioural understanding	Allows for combination of requirement identification/business needs and theoretical conceptualization	
Tentative design and requirements	DSR (e.g., Hevner et al. 2004)	Shaping requirements and business needs	Fosters the relevance of the problem solution	Component 1: Defines what the system is for
Development	DSR (Hevner et al. 2004; Kuechler and Vaishnavi 2008a, b)	Development of the IT artefact	Strengthens the practical contribution of the DSR finding	Component 8: The physical implementation of the IT artefact
Evaluation	DSR (Basili 1996; Kuechler and Vaishnavi 2008a, b)	Evaluation of the IT artefact	Confirms whether the problem is solved	Component 5: Evaluates whether the IT artefact satisfies the requirements
Additional theoretical sampling	GTM (Glaser 1978)	Collection and analysis of additional data based on IT artefact embedded in the environment	Extends the scope of the analysis beyond business needs/requirements toward a behavioural understanding based on preliminary theoretical sampling	Component 7: Helps implement additional theoretical insights about the IT artefact through additional data collection and analysis
Additional slices of data	GTM (Fernandez 2004; Levina and Vaast 2005)	Identification of additional slices of data for behavioural understanding	Strengthens the theoretical conceptualization process within the DSR process	
Categories	GTM (Glaser 1978)	Generation of conceptual categories (and properties) from the intertwined data collection and analysis	Forces DSR researchers to conceptualize and reach theoretical abstraction	
Conclusion	DSR (Hevner et al. 2004; Kuechler and Vaishnavi 2008a, b)	Embedding theoretical insights into DSR literature	Closes the loop by stimulating further DSR research that builds on the generated theoretical findings	Component 3: An additional theoretical contribution of the IT artefact and its behaviours can function as a blueprint for follow-up projects

interviews with the prospective users, pointed out that their understanding of the problem was somewhat different. They thought that it would only provide them with more advanced hard- and software, without however triggering a fundamental change in earth science. For instance, a programmer working for earth scientists pointed out that he does not understand that this project is beneficial for him. To meet these moving targets, a questionnaire was sent out to all participants and several interviews were conducted. Thereby, the participants and the collected data represent environmental factors as *slices of data*. For instance, the interviewees were asked about potential problems with Sequoia 2000. They pointed out that some problems with the handling of the technology could occur. The interviewees based these statements on experiences

with the programmers who emphasized that assistance in the use of the technology was not within scope. As a consequence, the development was adapted to this insight and earth scientists were included prior to the implementation. This implied that the potential user became accustomed to the technology which resembles an *intertwined process* of data collection and analysis to refine the *tentative design* and the *requirements* of the technology.

After solving these problems, Sequoia 2000 was *developed* and *evaluated* stepwise. The first prototypes were fed with data from the earth scientists to test the performance. Several problems occurred in this process. For instance, the network went down several times or the file system broke down because of the huge mass of data. As a consequence,

additional theoretical sampling and *slices of data* were collected from the participants. The programmers saw the earth scientists involved in the build and evaluate step as a kind of test bed for early systems testing and evaluation. However, the earth scientists saw their role as customers not being interested in acting as test persons but rather receive a stable and ready-to-use technology that enables them to conduct their research. The understanding of the earth scientists as customers rather than as participants in the design reinforced their research goals as being more important than the design goals. Hence, Sequoia 2000 had to go through several other improvement loops because of additional deliverables that were not fixed in the initial requirements analysis.

The entire Sequoia project can be seen as an example of including behavioural science aspects into the design and development of a technology. This project generated different *conclusions* and thereby additions to the knowledge base. In particular, Weedman (2008) developed two different types of contributions: First, she developed an explanation of how the meta-requirements of handling large or massive amounts of data in the domain of earth science could be matched with different kinds of solution components, including a cross-disciplinary metadata schema and associated chunking strategy for database management, principles of data visualization, and the principle of data reusability for running different analysis tasks. According to Baskerville and Pries-Heje (2010), this type of contribution resembles an ‘explanatory design theory’, i.e. the design product (Walls et al. 1992). However, the second and probably more important contribution of Weedman (2008) was the development of categories and relationships between them regarding the design process of solving ill-defined or wicked problems. Such a ‘design process theory’ (Baskerville and Pries-Heje 2010) is represented in the following Fig. 3.

Accordingly, different *categories* emerged from Weedman’s (2008) analysis which are related to the core

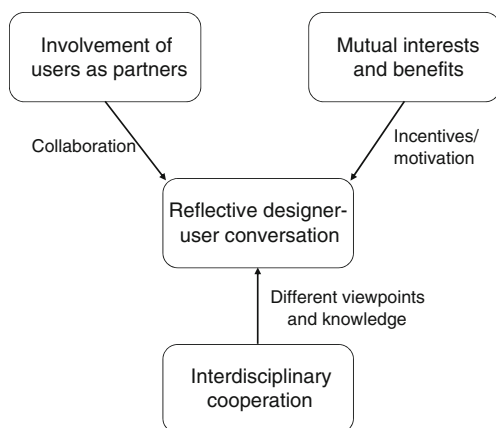


Fig. 3 Derived design process theory drawn from Weedman (2008)

notion of collaborative DSR projects with ‘reflective designer-user conversations’ at its centre. Adapted from Schön (1983, 1992), it represents the idea that designers and users engage in interactive conversations about both problem formulation and solving. Such a conversation stimulates a kind of reflection-in-action and shared understanding between the designers and users which leads to enhanced design outcomes. This central aspect of the collaboration is leveraged and enabled through three further concepts. First, designers and users are motivated by placing mutual interest and benefits at the centre of team composition. According to Weedman (2008), this can be further leveraged through appropriate incentive structures. Second, the concept of interdisciplinary cooperation provides different viewpoints and knowledge into the design process. Therewith, different knowledge types to be combined, including domain expert knowledge and design knowledge. Finally, users are involved as partners in the design process to stimulate collaboration with designers. In summary—drawing from Weedman (2008)—a design process theory can be derived that addresses motivational (i.e. mutual interests and benefits), cognitive (i.e. interdisciplinary knowledge), and behavioural (i.e. designer-user collaboration) aspects.

7 Discussion

While DSR has become an established area of research within IS, it still lacks the maturity of more accepted research approaches in terms of set of research instruments used or evaluation of theoretical contributions made. Since the seminal work of Hevner et al. (2004) there is an even stronger debate how DSR can be combined with other research methods to increase its rigor.

Theory-generating DSR focuses on the creation and evaluation of theoretical abstractions from an IT artefact by combining elements of constructive research with key concepts and techniques from interpretative research, thus allowing for grounded theorizing. Our suggested approach extends the general design cycle for DSR proposed by Kuechler and Vaishnavi’s (2008a, b). Different from Holmström et al. (2009) and Sein et al. (2011) we do not include behavioural science as an additional and independent component into DSR but rather develop a process model that enables a simultaneous and highly intertwined use of both. In so doing, we can combine the development of an IT artefact that solves a class of real-world problems in a new and innovative way with rigorous theorizing, such as about how the newly developed IT artefact interacts with its environment.

In particular, the awareness of the problem and the suggestion step can be enhanced through a detailed analysis of the environment and the extant knowledge base. Through

theoretical sampling, slices of data, and the constant comparison method researchers may be able to explore the problem and the phenomenon more precisely and analyze potential failures in the development in advance. A tentative design and the requirements of the IT artefact are more detailed through this refined step in theory-generating DSR. In addition, the extension of theory-generation after the evaluation of the IT artefact enables researchers to create an additional theoretical insight, e.g., from the application of the IT artefact in the appropriate environment, applicable knowledge from existing literature, or results from the IT artefact evaluation. This additional theoretical component offers another research outcome, but it also complements the underlying DSR approach and thereby meets the requirements for a IS design theory, as specified by Gregor and Jones (2007). As an exemplary case from extant literature, we used Weedman (2008) who used different behavioural science elements to analyze their developed IT artefact in more detail.

From a scientific point of view, theory-generating DSR focuses on the IT artefact and its improvement, as well as offering additional theoretical insights based on the use of the IT artefact. Thus, its reusable contribution adds to existing knowledge bases. Furthermore, it designs and creates IT artefacts as means to discover new knowledge (Baskerville et al. 2009), which distinguishes theory-generating DSR from, e.g., action research, which aims to create change in an organizational setting and studies the subsequent effects (Baskerville et al. 2009). Moreover, action researchers take an action and apply extant theories within the course of the research project (Coglan and Coughlan 2002; McKay and Marshall 2001). It thus explores the research project from an internal perspective; the action researcher works with the people directly affected by the action or who have the potential to influence the action in their environment (Avison et al. 1999). In contrast, a theory-generating DSR researcher observes phenomena and interacts only within the scientific environment. Accordingly, theory-generating DSR provides findings about the potential improvements to an IT artefact, and the researcher simply observes.

8 Conclusion

We have developed and presented a conceptual approach for a complementary use of DSR and behavioural science research elements, on the basis of Kuechler and Vaishnavi's (2008a, b) model and comparison with components necessary for IS design theories. For our initial theory development, we used interrelations to combine previously unconnected bodies of knowledge (DSR and GTM). In the course of this combination, we identified gaps in their conceptualizations and added missing components to an

already existing DSR model from Kuechler and Vaishnavi's (2008a, b). However, the proposed approach is a general process model, rather than a strict recipe, and has not yet been challenged or tested through application to actual DSR projects.

According to Hevner et al. (2004), theoretical contributions to the knowledge base represent an important and necessary part of DSR. We therefore propose the combination of DSR and GTM to unite design and behavioural aspects. In particular, we illustrate how design and behavioural research elements combine effectively in a pluralistic research design, which responds to calls to find potential research improvements (e.g., Mingers 2001).

Future research in this direction should evaluate whether such a combined research method is applicable or not in DSR projects. The used behavioural science method in this paper uses elements from GTM which is only one out of many methods in the field. Hence, this paper presents a new attempt to combine design with behavioural research. Future research should evaluate further the simultaneous use of behavioural and DSR methods into a pluralistic research design.

In summary, for a scientific discipline, generating contributions to its knowledge base is at least as important as solving real-world problems. We strongly believe that the proposed theory-generating DSR approach as one possible combination of design and behavioural science can support this goal by providing a process model for researchers who strive to follow and accomplish this aim in a research setting. From a theoretical perspective, this model can provide a fruitful contribution to the DSR community and expect to be very interesting to researchers focusing on DSR from a methodological point of view. From a practical perspective, managers could use the developed model for their DSR projects and thereby improve the development of prototypes in a structured way to avoid failure or repair loops.

Appendix: Review of DSR Articles

In the course of our literature research, we found several articles beside Weedman (2008) that offer support for various steps of our theory-generating DSR approach. Each article depicts a usage of DSR methods. Many articles present the IT artefact as an instantiation, though as Hevner et al. (2004) state: "IT artefacts can also be represented by constructs, models, methods, instantiations, or a combination thereof" (see also, March and Smith 1995). In Table 3, we list articles that discuss typical examples of DSR and that provided some basic ideas for our theory-generating DSR approach.

Table 3 Literature using DSR as a research approach

Article	Developed IT-Artefact	Description
Aalst and Kumar (2003)	Interorganizational routing language based on XML	The authors describe a standard problem-oriented approach. They use DSR methods accidentally to create a new interorganizational routing language. They test this language with a prototype implementation. The article depicts the typical engineering and problem-oriented approach from which DSR is derived.
Abbasi and Chen (2008)	CyberGate: A design framework and system for text analysis of computer-mediated communication	The authors conduct a procedure analogous to Aalst and Kumar's (2003) and develop CyberGate, an IT artefact for text analysis techniques.
Albert et al. (2004)	GIST: A model of the design and management of content and interactivity of customer-centric Web sites	The authors describe a standard problem-oriented approach. The IT artefact is derived from existing models, methods, and additional data. The requirements are obtained from an analysis of the existing environment. The authors conduct an elaborate evaluation of the IT artefact by collecting interview and survey data among users to get a deeper insight on its usability. Moreover, they conduct a gap analysis to identify potential gaps and missing values in their created model. Unfortunately, they do not use the data they gathered to improve the IT artefact.
Umapathy et al. (2008)	A model for knowledge reuse for designing enterprise integration solutions	The authors describe a standard problem-oriented approach. The requirements of the IT artefact are described by an analysis of the existing environment. The evaluation of the IT artefact involves a comparison of the old IT artefact with the new one.
Markus et al. (2002)	A design theory for systems that support emergent knowledge processes	The authors use DSR to create their IT artefact without mentioning it. Their research offers a closely intertwined combination of DSR and behavioural science. The authors build their IT artefact on existing kernel theories and conduct interviews to ensure the usability of their IT artefact. Unfortunately, these steps were not sequential. Several iterations of building and testing the IT artefact were conducted with so-called throw-away prototypes. The article provides extremely fruitful suggestions for our theory-generating DSR approach.

References

- Aalst, W., & Kumar, A. (2003). XML-based schema definition for support of interorganizational workflow. *Inform Syst Res*, 14(1), 23–46.
- Abbasi, A., & Chen, H. (2008). CyberGate: a design framework and system for text analysis of computer-mediated communication. *MIS Quarterly*, 32(4), 811–837.
- Albert, T. C., Goes, P. B., & Gupta, A. (2004). GIST: a model for design and management of content and interactivity of customer-centric web sites. *MIS Quarterly*, 28(2), 161–182.
- Allen, D. K., Colligan, D., Finnie, A., et al. (2000). Trust, power and interorganizational information systems: the case of the electronic trading community TransLease. *Inform Syst Res*, 10(1), 21–40.
- Arazy, O., Kumar, N., & Shapira, B. (2010). A theory-driven design framework for social recommender systems. *J Assoc Inform Syst Online*, 11(9), 455–490.
- Au, Y. A. (2001). Design Science I: the role of design science in electronic commerce research. *Comm Assoc Inform Syst*, 7(1), 1–15.
- Avison, D., Lau, F., Myers, M., et al. (1999). Action research. *Comm ACM*, 42(1), 94–97.
- Basili VR (1996) The role of experimentation in software engineering: past, current, and future. *Proceedings of the International Conference on Information Systems (ICIS)*, Ohio, USA.
- Baskerville, R. (2008). What design science is not. *Eur J Inform Syst*, 17(5), 441–443.
- Baskerville, R., & Pries-Heje, J. (1999). Grounded action research: a method for understanding IT in practice. *Account Manag Inform Tech*, 9(1), 1–23.
- Baskerville, R., & Pries-Heje, J. (2010). Explanatory design theory. *Business & Information Systems Engineering*, 2(5), 271–282.
- Baskerville, R., & Stage, J. (2001). Accommodating emergent work practices: ethnographic choice of method fragments. In B. Fitzgerald, N. Russo, & J. I. DeGross (Eds.), *Realigning research and practice: the social and organizational perspectives*. Boston: Kluwer Academic Publishers.
- Baskerville R, Pries-Heje J, Venable J (2009) Soft design science methodology. *Proceedings of Design Science Research in Information Systems and Technology (DESRIST)*, Philadelphia, USA.
- Benbasat, I., & Zmund, R. W. (2003). The identity crisis within the IS discipline: defining and communicating the discipline's core properties. *MIS Quarterly*, 27(2), 183–194.
- Bryant, A., & Charmaz, K. (2007a). Grounded theory in historical perspective: an epistemological account. In A. Bryant & K. Charmaz (Eds.), *The SAGE handbook of grounded theory*. London: SAGE Publications.
- Bryant, A., & Charmaz, K. (2007b). Grounded theory research: methods and practices. In A. Bryant & K. Charmaz (Eds.), *The SAGE handbook of grounded theory*. London: SAGE Publications.
- Carlsson SA (2006) Towards an information systems design research framework: a critical realist perspective. *Proceedings of Design Science Research in Information Systems and Technology (DESRIST)*, Claremont, USA.

- Coglan, D., & Coughlan, P. (2002). Action research for operations management. *Int J Oper Prod Manag*, 22(2), 220–240.
- Dubin, R. (1978). *Theory building*. New York: Free Press.
- Eisenhardt, K. M. (1989). Building theories from case study research. *Acad Manage Rev*, 14(4), 532–550.
- Eisenhardt, K. M., & Graebner, M. E. (2007). Theory building from cases: opportunities and challenges. *Acad Manage J*, 50(1), 25–32.
- Fernandez WD (2004) The grounded theory method and case study data in IS research: issues and design. *Proceedings of Information Systems Foundations Workshop: Constructing and Criticising*, Canberra, Australia.
- Glaser, B. G. (1978). *Theoretical sensitivity: advances in the methodology of grounded theory*. Mill Valley: Sociology Press.
- Glaser, B. G. (1998). *Doing grounded theory: issues and discussions*. Mill Valley: Sociology Press.
- Glaser, B. G. (2007). *Doing formal theory*. London: SAGE Publications.
- Glaser, B. G., & Strauss, A. L. (1967). *The discovery of grounded theory: strategies for qualitative research*. Chicago: Aldine Publishing Company.
- Goldkuhl, G. (2004). Design theories in information systems—a need for multi-grounding. *J Inform Tech Theor Appl*, 6(2), 59–72.
- Gregor, S. (2002). Design theory in information systems. *Aust J Inform Syst*, 10(1), 14–22.
- Gregor, S. (2006). The nature of theory in information systems. *MIS Quarterly*, 30(3), 611–642.
- Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *J Assoc Inform Syst Online*, 8(5), 312–335.
- Gregory, R. W., & Muntermann, J. (2011) Theorizing in design science research: inductive versus deductive approaches. In: Thirty second international conference on information systems, Shanghai, China.
- Hevner, A. R., & March, S. T. (2003). The information systems research cycle. *IT Systems Perspective*, 36(11), 111–113.
- Hevner, A., March, S., Park, J., et al. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75–105.
- Holmström, J. B., Ketokivi, M., & Hameri, A. P. (2009). Bridging practice and theory: a design science approach. *Decision Sci*, 40(1), 65–87.
- Ketokivi, M., & Mantere, S. (2010). Two strategies for inductive reasoning in organizational research. *Acad Manage Rev*, 35(2), 315–333.
- Kuechler, B., & Vaishnavi, V. (2008a). On theory development in design science research: anatomy of a research project. *Eur J Inform Syst*, 17(5), 489–504.
- Kuechler B, Vaishnavi V (2008b) Theory development in design science research: anatomy of a research project. In *Proceedings of Design Science Research in Information Systems and Technology (DESRIST)*, Atlanta, GA.
- Lee JS, Pries-Heje J, Baskerville R (2011) Theorizing in design science research. *6th International Conference on Design Science Research in Information Systems and Technology*, Milwaukee, WI.
- Levina, N., & Vaast, E. (2005). The emergence of boundary spanning competence in practice: implications for implementation and use of information systems. *MIS Quarterly*, 29(2), 335–363.
- Madill, A., Jordan, A., & Shirley, C. (2000). Objectivity and reliability in qualitative analysis: realist, contextualist and radical constructionist epistemologies. *Br J Psychol*, 91(1), 1–20.
- March, S. T., & Smith, G. (1995). Design and natural science research on information technology. *Decis Support Syst*, 15(4), 251–266.
- Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly*, 26(3), 179–212.
- McKay, J., & Marshall, P. (2001). The dual imperatives of action research. *Inform Tech People*, 14(1), 46–59.
- McKay J, Marshall P (2005) A review of design science in information systems. *Proceedings of the Australasian Conference on Information Systems (ACIS)*, Sydney, Australia.
- Mingers, J. (2001). Combining IS research methods: towards a pluralist methodology. *Inform Syst Res*, 12(3), 240–259.
- Nagel, E. (1961). *The structure of science*. New York: Harcourt, Brace & World.
- Nunamaker, J. F., Chen, M., & Purdin, T. D. M. (1991). Systems development in information systems research. *J Manag Inform Syst*, 7(3), 89–106.
- Orlikowski, W. J., & Iacono, C. S. (2001). Research commentary: desperately seeking the ‘IT’ in IT research—a call to theorizing the IT artifact. *Inform Syst Res*, 12(2), 121–134.
- Peppers, K., Tuunanen, T., Rothenberger, M. A., et al. (2008). A design science research methodology for information systems research. *J Manag Inform Syst*, 24(3), 45–77.
- Pries-Heje, J., & Baskerville, R. (2008). The design theory nexus. *MIS Quarterly*, 32(4), 731–755.
- Schön, D. A. (1983). *The reflective practitioner: how professionals think in action*. New York City: Basic Books.
- Schön, D. A. (1992). Designing as reflective conversation with the materials of a design situation. *Knowl Base Syst*, 5(1), 3–14.
- Sein, M. K., Henfridsson, O., Puaro, S., et al. (2011). Action design research. *MIS Quarterly*, 35(1), 37–56.
- Simon, H. A. (1969). *The science of the artificial*. Cambridge: MIT Press.
- Strauss, A., & Corbin, J. (1990). *Basics of qualitative research: grounded theory procedures and techniques*. London: Sage Publications.
- Suddaby, R. (2006). From the editors: what grounded theory is not. *Acad Manage J*, 49(4), 633–642.
- Udo K (2005) ‘Emergence’ vs. ‘Forcing’ of empirical data? A crucial problem of ‘grounded theory’ reconsidered. *Forum Qualitative Social Research*, 6(2), Art. 27.
- Umamathy, K., Puro, S., & Barton, R. R. (2008). Designing enterprise integration solutions: effectively. *Eur J Inform Syst*, 17(5), 518–527.
- Urquhart, C. (2007). The evolving nature of grounded theory method: the case of the information systems discipline. In A. Bryant & K. Charmaz (Eds.), *The SAGE handbook of grounded theory*. London: SAGE Publications.
- Urquhart, C., Lehmann, H., & Myers, M. D. (2010). Putting the ‘theory’ back into grounded theory: guidelines for grounded theory studies in information systems. *Inform Syst J*, 20(4), 357–381.
- Vaishnavi, V., & Kuechler, W. (2008). *Design science research methods and patterns—innovating information and communication technology*. Boca Raton: Auerbach Publications Taylor & Francis Group.
- Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Inform Syst Res*, 3(1), 36–59.
- Weber, S., Beck, R., & Gregory, R. W. (2012) Combining design science and design research perspectives - Findings of three prototyping projects. In: 45 Hawaii international conference on system sciences, Wailea, Maui, HI.
- Weedman, J. (2008). Client as designer in collaborative design science research projects: what does social science design theory tell us? *Eur J Inform Syst*, 17(5), 476–488.
- Winter, R. (2008). Design science research in Europe. *Eur J Inform Syst*, 17(5), 470–475.
- Yadav, M. S. (2010). The decline of conceptual articles and implications for knowledge development. *J Market*, 74(1), 1–19.

Prof. Dr. Roman Beck is the E-Finance and Services Science Chair at Goethe University, Frankfurt, Germany. His research focuses on the creation, sourcing, and management of knowledge intensive IT services and has been published in journals such as Communications of AIS, Electronic Markets, IEEE Transactions on Software Engineering, Information Systems Frontiers, Information Technology & People, and others.

Sven Weber is a research assistant and doctoral student at the E-Finance Lab at Goethe University, Frankfurt, Germany. His research focuses on the combination of design science and behavioural science research and the management of communication systems. His research

has been published at several IS conferences (e.g., AMCIS, DESRIST, HICSS).

Dr. Robert Wayne Gregory is an Assistant Professor at the Chair of Electronic Finance and Digital Markets at the University of Göttingen. He holds a PhD in Information Systems from University of Frankfurt and has a focus on both behavioural and design science-oriented research. Robert has been involved in a wide range of research projects that have been conducted in collaboration with industry, especially with financial service providers and banks. His research has been published in outlets such as Information Technology & People and ICIS.