DESERV IT: A Method for Devolving Service Tasks in IT Services

Florian Baer · Kurt Sandkuhl · Michael Leyer · Birger Lantow

Abstract Nowadays, IT operations devolve many tasks in IT services to internal customers (i.e., IT self-service). The rationale for this service task devolvement is often to reduce the IT personnel's workload. However, prior research has shown that IT operations often fail to achieve this goal. Existing methods for modeling and analyzing services fall short of supporting service providers in identifying and specifying service tasks suitable to be devolved to (internal) customers. This paper presents, therefore, the first method for devolving service tasks in IT services (DESERV IT). DESERV IT is a compound of four method components encompassing a joint meta-model, a visual notation for modeling IT services, and procedural recommendations. The DESERV IT meta-model extends the meta-model of service blueprinting by means of concepts required to analyze service task devolvement. DESERV IT is evaluated in four evaluation episodes. The results of the evaluation episodes show that DESERV IT is perceived as effective, useful, complete, and generalizable by experts in the IT service management and enterprise architecture discipline. This paper contributes to enterprise modeling by demonstrating the feasibility of DESERV IT in an example case and describing DESERV IT's evolution during the evaluation episodes. DESERV IT supports practitioners (e.g., request fulfillment managers) in modeling and analyzing IT services.

F. Baer (✉) · K. Sandkuhl · B. Lantow
Chair Business Information Systems, University of Rostock,
Albert-Einstein-Straße 22, 18051 Rostock, Germany
e-mail: Baer.Florian@gmail.com

K. Sandkuhl
e-mail: Kurt.Sandkuhl@uni-rostock.de

B. Lantow
e-mail: Birger.Lantow@uni-rostock.de

M. Leyer
Chair Service Operations, University of Rostock, Ulmenstraße 69, 18051 Rostock, Germany
e-mail: Michael.Leyer@uni-rostock.de

1 Introduction

Within enterprises, IT services are performed by IT operations (Marrone and Kolbe 2011; Steinberg et al. 2013). In recent years, internal customers have become actively involved in the performance of IT services due to the proliferation of IT self-services (Zaza and Junglas 2016). In IT self-services, IT operations devolve service tasks, such as the reset of passwords, to internal customers, so that the internal customers are enabled to perform these service tasks on their own (Kumar and Telang 2012; Scherer et al. 2015). In particular, with the ongoing integration of software development and IT operations (DevOps) (Roche 2013), more and more service tasks are devolved by IT operations to software engineers.

The rationale for IT self-services is not the reduction of operational costs, because such a cost reduction would only transfer costs from IT operations to other organizational functions of the same enterprise (Campbell et al. 2010).

Table 1 Method requirements that must be fulfilled by DESERV IT

Specific functional method requirements	
Method requirement 1 (MR1)	The method must support IT operations in identifying and analyzing failures which occur in the IT self-service due to service task devolvement
Method requirement 2 (MR2)	The method must support IT operations in identifying and analyzing capability gaps in the IT self-service
Method requirement 3 (MR3)	The method must support IT operations in identifying and analyzing those IT self-services whose outcomes rely on IT resources and are produced for free for the internal customers
Method requirement 4 (MR4)	The method must support IT operations in identifying and analyzing the need to adopt solutions which comprise sets of the identified behavioral patterns, to prevent failures from occurring
Generic environmental method requirements	
Method requirement 5 (MR5)	The method must be effective, i.e., it must support IT operations in devolving service tasks to internal customers in order to reduce the IT personnel's workload in the resulting IT self-services
Method requirement 6 (MR6)	The method must be useful to IT operations
Method requirement 7 (MR7)	The method must be complete, i.e., it must include all method components required to be effective (see MR5)
Method requirement 8 (MR8)	The method must be generalizable, i.e., it must be relevant not only for local practice but also for global practice

Instead, self-services are about increasing customer satisfaction by ensuring a high convenience of services (Bitner et al. 2002; Collier and Kimes 2012), while freeing the service personnel from performing recurrent, routine service tasks (Baer et al. 2018). Within enterprises, IT operations are an organizational function that supports the primary business processes. Therefore, from an IT operations' perspective, IT self-services are introduced to reduce the time the IT personnel (i.e., IT operations' personnel) must spend on operational tasks that support long-established business processes. In times when IT-supported decision-making and the development of artificial intelligence become increasingly important for enterprises when designing product and service innovations, IT operations must perform a more strategic function in enterprises. By reducing the workload that the IT personnel must spend on supporting long-established business processes, IT operations are enabled to identify ways to actively support and enable business processes that drive product and service innovation.

However, information systems research and our own observations have shown that the reduction of the workload of the (IT personnel) service personnel in (IT) self-services is not self-evident (Kumar and Telang 2012; Baer et al. 2018). Although scholars have called for methods that can guide managerial decisions on service task devolvement (Scherer et al. 2015), research in this domain has primarily taken a customer perspective (e.g., customer satisfaction and adoption in the self-service context). Hence, to date, self-service research from the service provider perspective is scarce, with the result that there is no method that supports (IT operations) service providers in devolving service

tasks to (internal) customers in (IT) services while considering the (IT personnel's) service personnel's workload as the target variable. Scholars have designed several methods for modeling and analyzing services in general (e.g., Becker et al. 2013; Trkman et al. 2015). However, none of these methods fulfills all the requirements identified as relevant for making well-founded decisions about service task devolvement, such as the identification and analysis of capability gaps (Baer et al. 2016, 2018). To fill this research gap, we investigated IT self-services from the perspective of IT operations and address the following research question: *How can IT operations be supported methodically in devolving service tasks to internal customers in IT services in such a way that the IT personnel's workload is reduced?*

The research question was addressed by designing a method for the DEvolvement of SERVice tasks in IT services (DESERV IT) that is an extension to service blueprinting (Shostack 1982). The design of DESERV IT has adopted a design science research approach (Johannesson and Perjons 2014). The objective of this paper is to contribute to enterprise modeling by presenting DESERV IT, demonstrating its feasibility in an example case, and describing DESERV IT's evolution during the evaluation episodes. The research findings demonstrate that IT operations can, when methodically supported by DESERV IT, devolve service tasks, which are complex in their structure and place heavy intellectual demands, to internal customers in IT services effectively.

Table 2 An example of a service catalog matrix

IT service	Description	Class	Outcome	Options
VM provisioning	Provisioning of a customized VM	Purchase and transaction	VM for testing and personal training and education	CPU, memory, and storage can be customized at will

Table 3 Suggestions for where to specify fail points in IT self-services

Fail point	Specification suggestion
Self-service failure	Must be specified for each devolved each service task for which capability gaps exist (see Sects. 7.2.3 and 7.3.2)
Ambiguous information	Must be specified in each "information seeking" (see Sect. 7.1.1) if information about subjects, of which the internal customers have little knowledge, is provided
Intentional misperformance	Must be specified for each devolved service task if the internal customers are believed to experience a personal advantage (e.g., reduced waiting time and financial benefit) when performing the service task incorrectly
Excessive outcome production	Must be specified in each "purchase and transaction" (see Sect. 7.1.1) if IT resources provided and maintained by IT operations are required to produce the IT service outcome
Forbidden outcome production	Must be specified in each "purchase and transaction" (see Sect. 7.1.1) if the internal customers are unaware of the constraints for using the IT service outcome

2 Theoretical Background

DESERV IT was designed by applying the method engineering approach proposed by Goldkuhl et al. (1998). Therefore, this and other method engineering approaches are described and discussed in Sect. 2.1. The theoretical perspective of DESERV IT is described in Sect. 2.2.

2.1 Method Engineering

Methods are widely used as instrumental support for different engineering and development activities in the context of enterprises, such as for enterprise modeling, enterprise architecture design, and information systems development. The research area of method engineering offers a rich body of knowledge about how to systematically develop, introduce, and adapt methods (Brinkkemper 1996; Goldkuhl et al. 1998; Henderson-Sellers et al. 2014). Methods are often considered as prescriptive since they are supposed to provide guidance for problem solving or for performing complex tasks (Seigerroth 2011). This requires a method to define what activities are to be performed, how to perform them (procedure), what results (artifacts) are to be developed, and how to capture these results (notation) (Seigerroth 2011). Different conceptualizations of the term "method" and related terms have been proposed. If there is a close link between procedure, notation, and concepts, the term "method component" is used (Goldkuhl et al. 1998). The concept of "method component" is similar to the concept of "method chunk" (Ralyté et al. 2006) and the notion of "method fragment" (Brinkkemper 1996).

DESERV IT, the envisioned method for service task devolvement in IT services, was designed by applying the method engineering approach proposed by Goldkuhl et al. (1998), which advocates a component-oriented engineering of methods. Method components offer guidelines by means of defining courses of action in different situations to realize certain goals, and these are essential for DESERV IT to meet the method requirements and to support different behavioral patterns and structural settings of IT operations (see Sects. 4 and 5). The approach of Goldkuhl et al. (1998) is well accepted in the enterprise modeling discipline and it is the conceptual basis for many enterprise modeling methods (e.g., Pastor et al. 2018).

According to Goldkuhl et al. (1998), a method builds on an implicit or explicit theoretical perspective that includes and motivates the value basis of the method and the goals to be realized by using the method. The theoretical framework of DESERV IT determines how (IT) (self-)services are defined in the method context, and it implies that IT operations strive to reduce the IT personnel's workload when IT self-services are implemented. The order in which the method components must be performed is defined in a framework. DESERV IT's framework orders the method components in a strict sequence so that their performance aligns with the goal of workload reduction. Co-operation forms define the roles that participate in completing the method components as well as the interactions and kinds of co-operation between these roles. DESERV IT draws on roles defined in the IT infrastructure library (ITIL) and defines their responsibilities in performing the method components. A method component

Table 4 Suggestions of which behavioral patterns to adopt to prevent failures in IT self-services

Fail point	Suggested behavioral patterns
Self-service failure	"Standardization of IT self-service" and "training and support"
Intentional misperformance	"Standardization of IT self-service" and "authorization of service requests"
Ambiguous information	"Standardization of IT self-service" and "training and support"
Excessive outcome production	"Chargeback and limitation", "standardization of IT self-service", "authorization of service requests", "showback", and "training and support"
Forbidden outcome production	"Standardization of IT self-service", "authorization of service requests", and "training and support"

Table 5 Analysis of the IT personnel's workload for an example case

IT service	Solutions to be adopted	IT personnel's workload
IT full-service	–	00:00:10 + 00:01:00 + 00:00:03 + 00:00:05 + 00:03:00 + 00:00:03 = 00:04:21
IT self-service (no adoption of solutions)	–	00:30:00 + 01:00:00 = 01:30:00
IT self-service 1	Solution 1 and solution 3	00:05:00[a,b]
IT self-service 2	Solution 1 and solution 4	00:00:00[b]
IT self-service 3	Solution 2 and solution 3	00:05:00[a,b]
IT self-service 4	Solution 2 and solution 4	00:00:00[b]

[a]"Solution 3" specified for "fail point 2" required the adoption of the "authorization of service requests" behavioral pattern. The standard execution time of this behavioral pattern was estimated to be 00:05:00

[b]The "action 4" and "action 6" service tasks were performed by the IT system, but not by the IT personnel. Therefore, they must not be included in the calculation of the IT personnel's workload

represents a close link between procedures, concepts, and a notation. Procedures inform the method participants about the concrete actions to be performed and involve concepts that capture aspects of reality of relevance to the method. The concepts are also part of the notation's semantics that defines how the results of the procedures must be documented. The procedures of DESERV IT guide the method participants through devolving service tasks to internal customers in IT services. The visual notation included by DESERV IT allows the modeling of the procedures' results.

2.2 Service Modeling

The theoretical perspective of DESERV IT is rooted in service operations research (Sampson 2012). Accordingly, for DESERV IT, services are defined as processes that transform inputs (i.e., IT resources and capabilities) provided by the service providers and (internal) customers into outcomes (e.g., software, virtual machines (VMs), or containers) used and determined by the (internal) customers (Yalley and Sekhon 2014). In this sense and in line with ITIL, IT services are defined in the method context as processes in which IT operations take the role of the service provider (Steinberg et al. 2013). Typically, services will be performed if (internal) customers request them (Leyer et al. 2017).

DESERV IT implies that services are a continuum with two extremes as its boundaries: full-services and self-services (Globerson and Maggard 1991). In full-services, all the service tasks are performed by the service providers either together with the (internal) customers or solely in the back office. In self-services, the (internal) customers perform portions of the service tasks on their own and independently from the service personnel. The more service tasks are devolved to the (internal) customers, the higher

Table 6 Overview of the evaluation episodes

Episode	Time period	Setting	Method	Criteria	Contribution
1	Ex ante	Naturalistic	Participatory action research	Perceived effectiveness and completeness	MR5 and MR7
2	Ex ante	Artificial	Informed arguments, peer reviews	Cognitive effectiveness and perceived completeness	MR5–MR8
3	Ex post	Naturalistic	Focus group	Perceived (cognitive) effectiveness, perceived usefulness, and perceived generalizability	MR5, MR6, and MR8
4	Ex post	Naturalistic	Expert interviews	Perceived (cognitive) effectiveness, perceived usefulness, and perceived generalizability	MR5, MR6, and MR8

the self-service intensity is, that is, the percentage of time that the (internal) customers must spend performing the service tasks (Haumann et al. 2015). Similarly, DESERV IT defines the service personnel's workload in a service as the percentage of time that the service personnel must spend to perform the service tasks.

The modeling of services enables the identification of failures at the design stage – that is, before they happen – and the development of new and innovative services (Shostack 1982). It represents a basis for improvements and advancements of services, such as increased service productivity and service quality (Shostack 1982; Yalley and Sekhon 2014). A method for modeling and analyzing services that is widely adopted by scholars and practitioners is service blueprinting (Sampson 2012; Becker et al. 2013). The concepts included in the meta-model of service blueprinting are also included in the meta-model of the business process model and notation (BPMN) (Milton and Johnson 2012; Kazemzadeh et al. 2015). However, the meta-model of BPMN does not also include the concepts that are identified as relevant for decision-making regarding service task devolvement (see Sect. 6.2 for these relevant concepts). Furthermore, in light of the theory for visual notation design by Moody (2009), BPMN has a number of flaws (Moody 2011). For DESERV IT, therefore, we decided to design a new visual notation which is compliant with the principles for cognitively effective visual notations according to Moody (2009).

DESERV IT is based on service blueprinting. However, while the target variable addressed by service blueprinting is service profitability, the target variable addressed by DESERV IT is workload reduction.

3 Research Method

To address our research question, we designed a method for the devolvement of service tasks in IT services. Because we aimed to design a new artifact, we adopted a design science research approach. More precisely, we used the method framework for design science research proposed by Johannesson and Perjons (2014). This method framework is well accepted in the business information systems research discipline (e.g., Jouck and Depaire 2018). Our research activities are depicted in Fig. 1.

To explicate the problem that prevents IT operations from reducing the IT personnel's workload in IT services through service task devolvement, we conducted a multiple-case study and explored five IT self-services implemented in two German IT consulting firms and a European software company. The results of this exploratory multiple-case study are presented by Baer et al. (2018). Section 4 summarizes the results.

Based on the explicated problem, we derived a set of method requirements (see Sect. 5) that must be fulfilled by any method supporting IT operations in devolving service tasks to internal customers. In a systematic literature review, we identified and examined 47 methods for modeling and analyzing services (Baer et al. 2016). The results of this review are summarized in Sect. 9.1.

We drew upon the method engineering approach proposed by Goldkuhl et al. (1998) to design and develop DESERV IT (see Sect. 6). First, with the target variable (i.e., workload reduction) in mind (see Sect. 2.1 for DESERV IT's perspective), we defined the required method components and put them in the required sequential order to represent the framework of DESERV IT. Second, for each required method component, we defined the set of relevant concepts and added them to the meta-model of service blueprinting to form DESERV IT's meta-model (see Sect. 6.2). In addition, we developed a visual notation to visually represent the concepts. Finally, for each required method component, the relevant procedures were defined and described. Furthermore, the visual notation was adapted to properly document the results of the procedures (see Sect. 7) and a responsibility assignment (RACI) matrix describing the participation of roles in completing the procedures was defined (see "Online

Table 7 Information about the interviews conducted in evaluation episode 4

Interviewee	Date	Type	Duration (min)
Managing consultant	11-29-2018	Semi-structured interview, via telephone and TeamViewer	90
Enterprise architect	11-30-2018	Semi-structured interview, via telephone and TeamViewer	60
Managing enterprise architect	12-13-2018	Semi-structured interview, via telephone	45
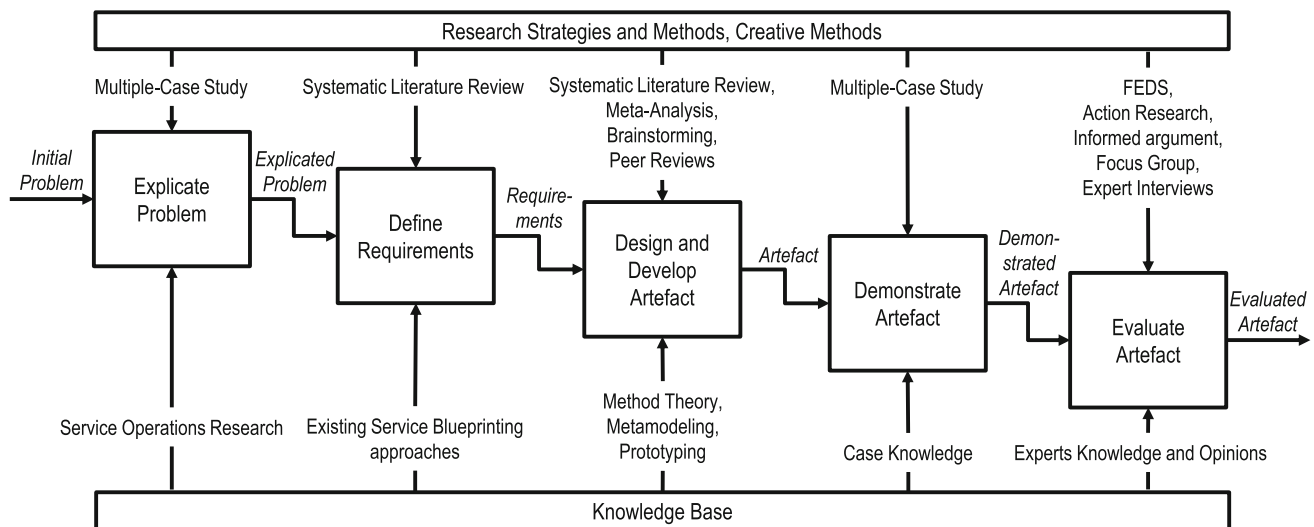
Fig. 1 The method framework for design science research with the research strategies and resources adopted from the knowledge base (Johannesson and Perjons 2014)

Appendix G"; available online via http://link.springer.com).

To demonstrate the feasibility (i.e., fulfillment of the functional method requirements) of DESERV IT, we applied DESERV IT to an example case. This case was analyzed as part of the multiple-case study, which was conducted to explicate the problem; this is summarized in Sect. 7.

To prove the fulfillment of the environmental method requirements, DESERV IT was evaluated according to the framework for evaluation in design science research (FEDS) (Venable et al. 2012). The evaluation includes four evaluation episodes (see Sect. 8). Whenever possible, we preferred to conduct an evaluation episode in a naturalistic setting to evaluate the fulfillment of the environmental method requirements. The first two evaluation episodes reflected *ex ante* evaluation strategies to best support the design and development of DESERV IT. In the first evaluation episode, an ex ante evaluation of a first draft of DESERV IT's framework was conducted in a naturalistic setting. In particular, we had the chance to take part in a participatory action research (Baskerville 1999) in an international IT consulting firm.

The evaluation strategy selected for the second evaluation episode was an ex ante evaluation in an artificial setting. DESERV IT was redesigned based on the feedback obtained from discussing it with the enterprise modeling and service operations community at the Ninth International Workshop on Enterprise Modeling and Information Systems Architectures (EMISA)[1] and the Sixth Rostock Service Management Conference.[2] Also, a first version of the visual notation was evaluated against Moody's (2009) principles for cognitively effective visual notations. As the demonstration of DESERV IT proved the fulfillment of the functional method requirements, we decided that no further redesign of DESERV IT was required, therefore we began conducting the ex post evaluations. Because we were unable to apply DESERV IT in a real-world project, the ex post evaluations in the last two episodes targeted domain experts' perceptions about DESERV IT. The third and fourth evaluation episodes reflected ex post evaluation strategies in naturalistic settings. In the third evaluation episode, we interviewed a focus group consisting of five experts in the field of IT service management (ITSM) about

[1] See http://www.emisa2018.informatik.uni-rostock.de/.
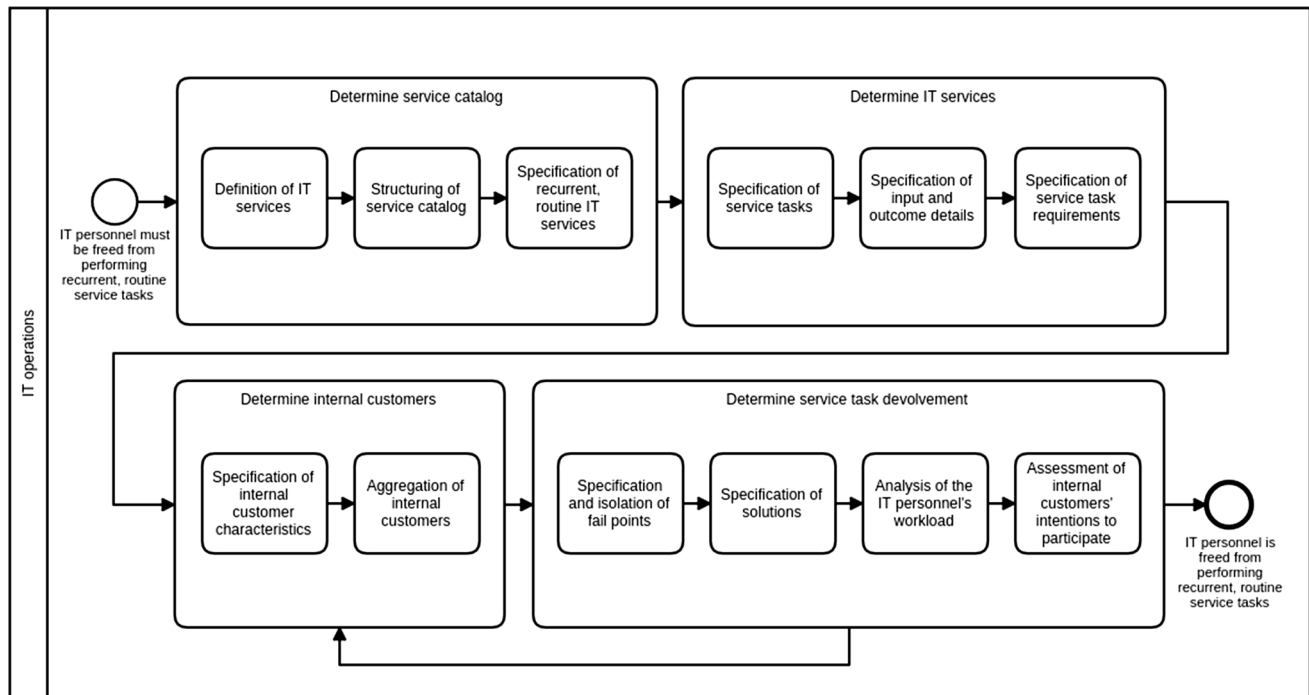[2] See https://www.wiwi.uni-rostock.de/dl-tagung/willkommen/.

Fig. 2 The framework of the latest version of DESERV IT

DESERV IT; in the fourth evaluation episode, three interviews about DESERV IT were conducted with experts in the fields of ITSM and enterprise architecture. We stopped the ex post evaluation, as we had collected enough evidence to demonstrate the perceived fulfillment of the environmental method requirements by DESERV IT.

4 Explicating the Problem

DESERV IT addresses the problem of a lack of control by IT operations in an IT self-service.

The devolvement of service tasks to internal customers in an IT service goes hand in hand with a transfer of control from IT operations to the internal customers in the IT service. The service provider's control in a self-service is conceptualized as the extent to which the service provider can ensure the correct performance of the devolved service tasks by the (internal) customers. If IT operations have too little control in an IT self-service, the IT personnel's workload will not be reduced in the IT self-service. The lack of this control by IT operations results in failures that occur in the IT self-service and must be corrected by the IT personnel, thereby increasing their workload. These failures are "self-service failures" (Hilton et al. 2013), "ambiguous information" (Kumar and Telang 2012), "intentional misperformance" (Ellway 2016), "excessive outcome production" (Baer et al. 2018), and "forbidden

outcome production" (Baer et al. 2018). The failures are conceptualized in "Online Appendix F".

A lack of control by IT operations for an IT self-service is due to capability gaps and/or a free IT self-service. Capability gaps will arise if the internal customers do not possess the capabilities required to perform the devolved service tasks correctly. A free IT self-service is an IT self-service whose outcome is produced for free for the internal customers. IT operations will face a lack of control in a free IT self-service if the IT service outcome relies on IT resources (e.g., central processing unit [CPU], memory, and storage) that are limited to IT operations and incur costs if they are increased. Therefore, with regard to service task devolvement, IT services must be classified based on their outcomes into "information seeking", "communication and interaction", and "purchase and transactions" (see "Online Appendix F"). While IT services of the "information seeking" type can be considered as simple, the service tasks required to perform IT services of the other two types are more complex and place heavy intellectual demands (Harrison and Waite 2015).

The adoptions of five behavioral patterns in various combinations form solutions to prevent the failures resulting from IT operations' lack of control in an IT self-service. These behavioral patterns are "chargeback and limitation", "standardization of IT self-service", "authorization of service requests", "showback", and "training
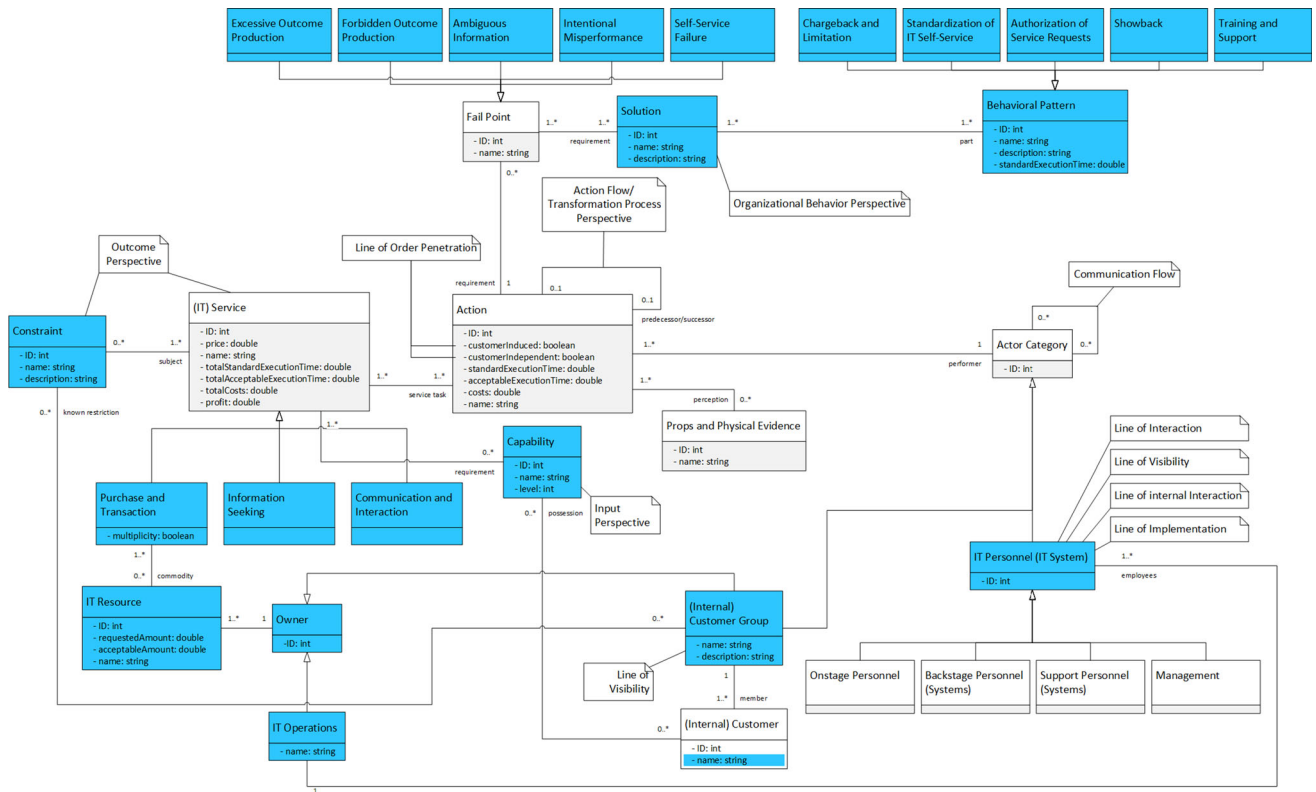
Fig. 3 Meta-model of DESERV IT. Notes: Concepts and attributes highlighted in white are included by the service blueprinting meta-model; concepts highlighted in blue are required for analysis of service task devolvement and therefore added to the service blueprinting meta-model

and support". The behavioral patterns are conceptualized in "Online Appendix F".

5 Defining Method Requirements

The problem explication and related research (Kumar and Telang 2012; Ellway 2016) reveal that IT operations often devolve service tasks to internal customers of IT services without analyzing, at the design stage, whether this service task devolvement causes a lack of control and, consequently, failures that must be corrected by the IT personnel at the execution stage. Hence, we identified the need for a method that supports IT operations in these situations to ensure that the IT personnel's workload in the resulting IT self-services is reduced at the execution stage.

Based on the problem explication, we derived four specific functional method requirements. The method must support the modeling of IT self-services in which there is no lack of control for IT operations and, therefore, a reduction of the IT personnel's workload. Hence, the method must support IT operations in analyzing IT service models regarding a potential lack of control (MR2 and MR3), failures that might occur due to this lack of control (MR1), and solutions to these potential failures (MR4).

Because we designed a socio-technical artifact, we also defined four generic environmental method requirements. Environmental method requirements, which are suggested by Johannesson and Perjons (2014), are relevant for any method. MR6 can only be evaluated by assessing the (potential) method participants' perceptions of the method's usefulness. Therefore, the fulfillment of MR6 cannot be taken for granted. A method will be complete if it includes all the method components required to realize the method's target variable. The method requirements are listed in Table 1.

Existing methods for modeling and analyzing services do not fulfill MR2–MR7 because they do not support a sufficient modeling and analysis of the inputs and service outcomes (see Sect. 9.1). In addition, many of these methods are designed for specific use cases, but not for a general usage, so that MR8 is not fulfilled. Therefore, we designed a new method that builds on service blueprinting and which, compared to the other methods, fulfills MR1 partially and MR8 completely.

Fig. 4 Graphical symbols representing the concepts "service task", "capability", "fail point", "solution", and "behavioral pattern"
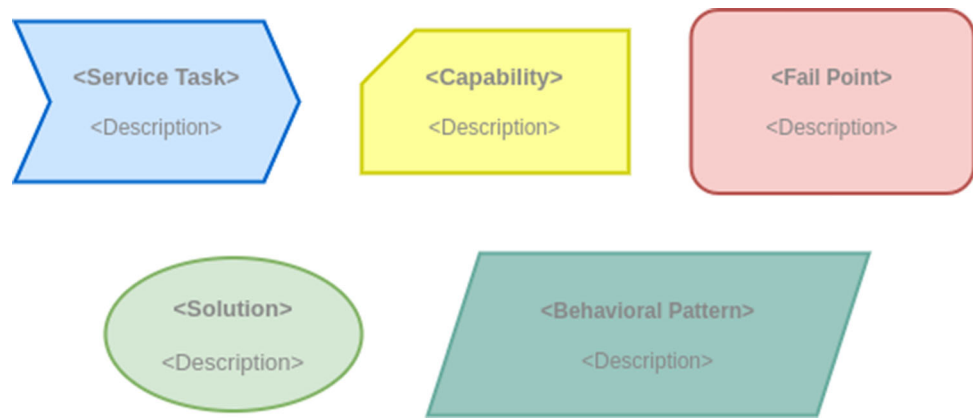
6 Design and Develop Artifact

DESERV IT is a compound of four method components that link a visual notation for modeling IT services, 12 procedures, and 39 concepts. DESERV IT's framework defining the order in which the method components must be performed is described in Sect. 6.1. The concepts included in the method components, as well as the relationships between these concepts, are defined by the DESERV IT meta-model, which is described in Sect. 6.2.

6.1 Method Framework

The framework of DESERV IT has a sequence of four method components: "determine service catalog", "determine IT services", "determine internal customers", and "determine service task devolvement". These method
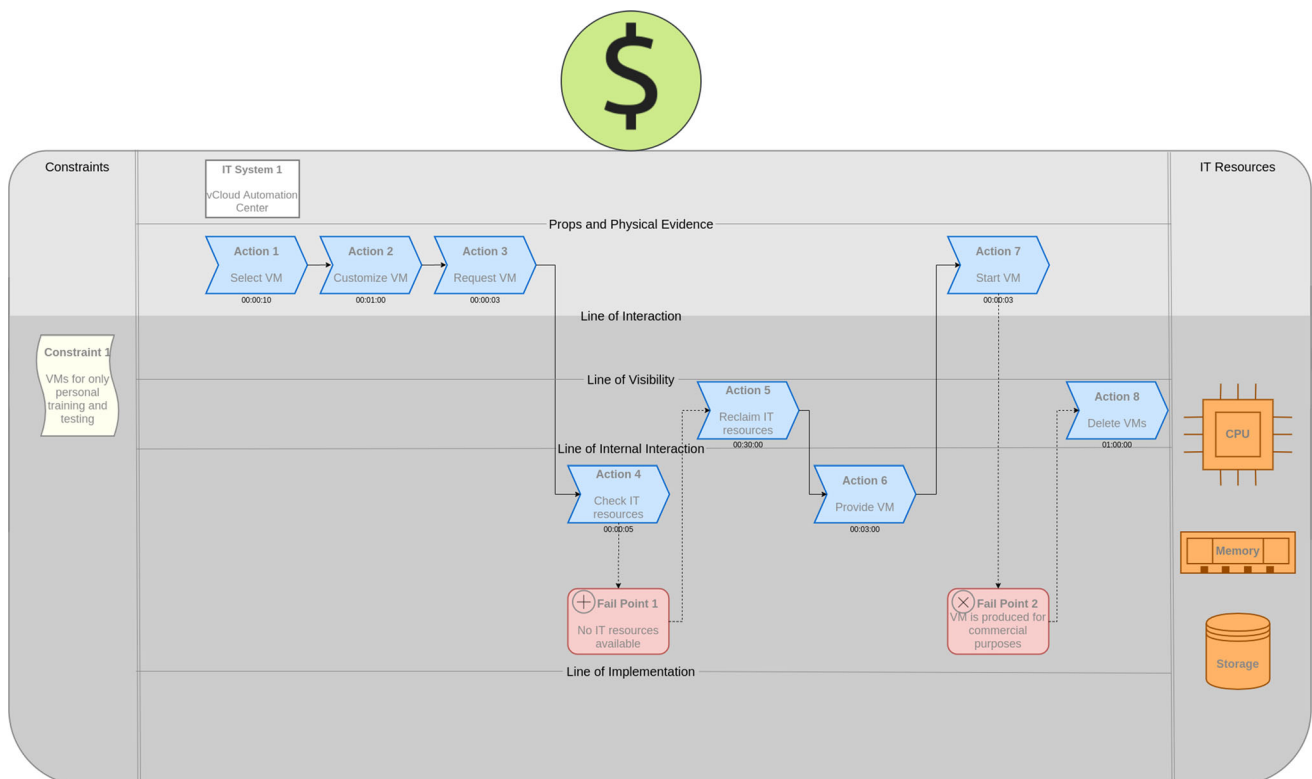

Fig. 5 An IT service model for an example case. It is a visual representation of the "IT service" concept

Fig. 6 Icons representing the "information seeking", "communication and interaction", and "purchase and transaction" concepts

components must be regarded as prerequisites, where a preceding method component is a prerequisite for the next method component, and so on. The framework is shown in Fig. 2. In Sect. 7, we describe the method components of DESERV IT in more detail by describing the procedures and the documentation of their results with the visual notation for an example case.

6.2 Meta-Model

The meta-model of DESERV IT includes all the concepts included by service blueprinting (see Fig. 3). To these concepts, the DESERV IT meta-model adds concepts required to analyze service task devolvement. These additional concepts are derived from the functional method requirements (see Sect. 5).

In matters of service task devolvement, IT services of the "purchase and transactions" type are of special interest. In "purchase and transactions", IT resources are exchanged, usually for free, between IT operations, which is the owner of these IT resources, and the internal customers

(see MR3). The IT personnel are defined as a special kind of service personnel.

The DESERV IT meta-model includes fail points for each failure occurring from a lack of control by IT operations in IT self-services (see MR1 and Sect. 4). With the exception of failures of the "intentional misperformance" type, failures are rooted in capability gaps. Therefore, for an action and group of internal customers, the capabilities required, respectively, to correctly perform the action and the capabilities possessed by the internal customers must be specified (see MR2). Internal customers possessing similar capabilities must be aggregated to internal customer groups. Failures of the "forbidden outcome production" type can also be rooted in internal customers' unawareness of constraints for using and producing the IT service outcome. To prevent the failures from occurring, IT operations must adopt solutions comprising combinations of the behavioral patterns identified according to the problem explication (see MR4 and Sect. 4).

The DESERV IT meta-model is shown in Fig. 3. The parts of the DESERV IT meta-model that are relevant for each method component are presented in "Online Appendices B, C, D, and E". "Online Appendix A" lists the relevant concepts for each method component. The meaning of the concepts included by the DESERV IT meta-model is provided by "Online Appendix F".

Fig. 7 Graphical symbols representing the "physical evidence", "IT system", "IT resource", and "constraint" concepts
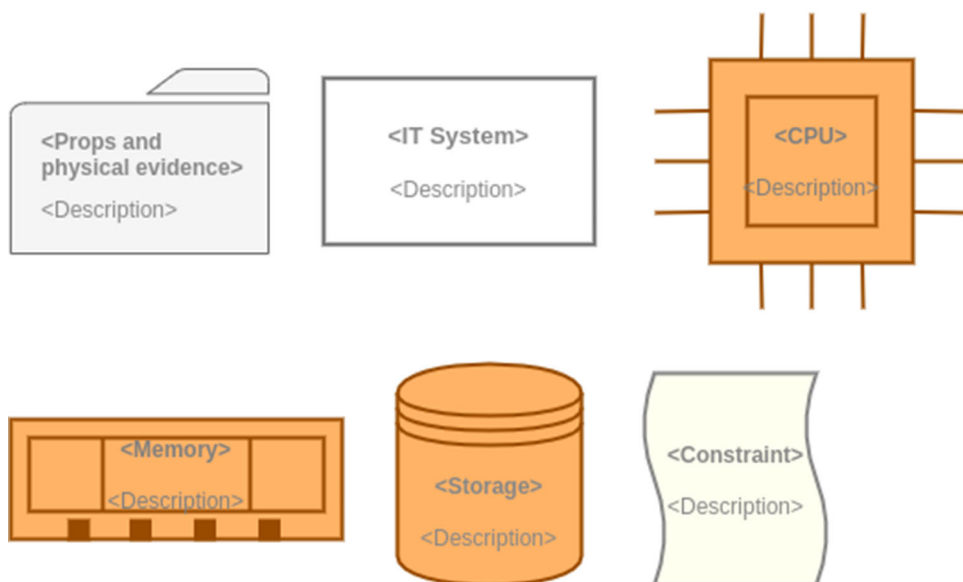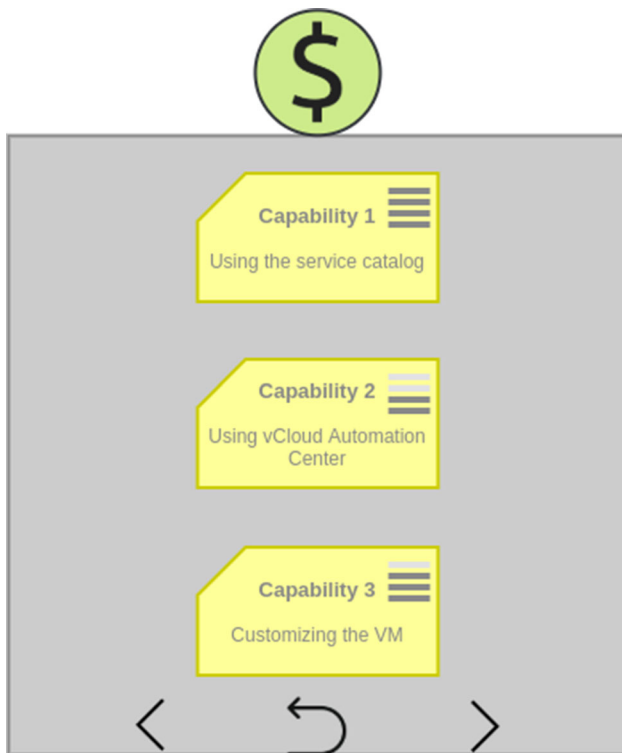
Fig. 8 A requirement model for an example case. It visually represents the "requires" relationship between the "IT service" and "capability" concepts

7 Demonstrate Artifact

In this section, we describe the procedures of each method component and demonstrate them by using an example case. The example case is analyzed as part of the problem explication (see Sect. 4) and reflects an IT self-service in which internal customers are enabled to provide VMs on their own. It is, therefore, a "purchase and transaction" and must be considered as complex (see Sect. 4). The example case is described in detail by Baer et al. (2018).

In the example case, the workload of the IT personnel was not reduced, because IT operations had only limited control of the IT self-service; thus, failures of the "excessive outcome production" and "forbidden outcome production" types occurred. These failures required the IT personnel to manually intervene in the IT self-service on a regular basis, which significantly increased the IT personnel's workload. Because of the workload non-reduction in the example case, IT operations decided to redesign the IT self-service and discussed several solutions to prevent the failures from occurring in the future.

7.1 Determine Service Catalog

This method component is about identifying those IT services that are requested frequently and on a large scale by the internal customers.

7.1.1 Definition of IT Services

The service catalog manager must define the IT services that can be requested by the internal customers. Each IT service must be classified into one of the three classes "information seeking", "communication and interaction", and "purchase and transactions" (see Sect. 4).
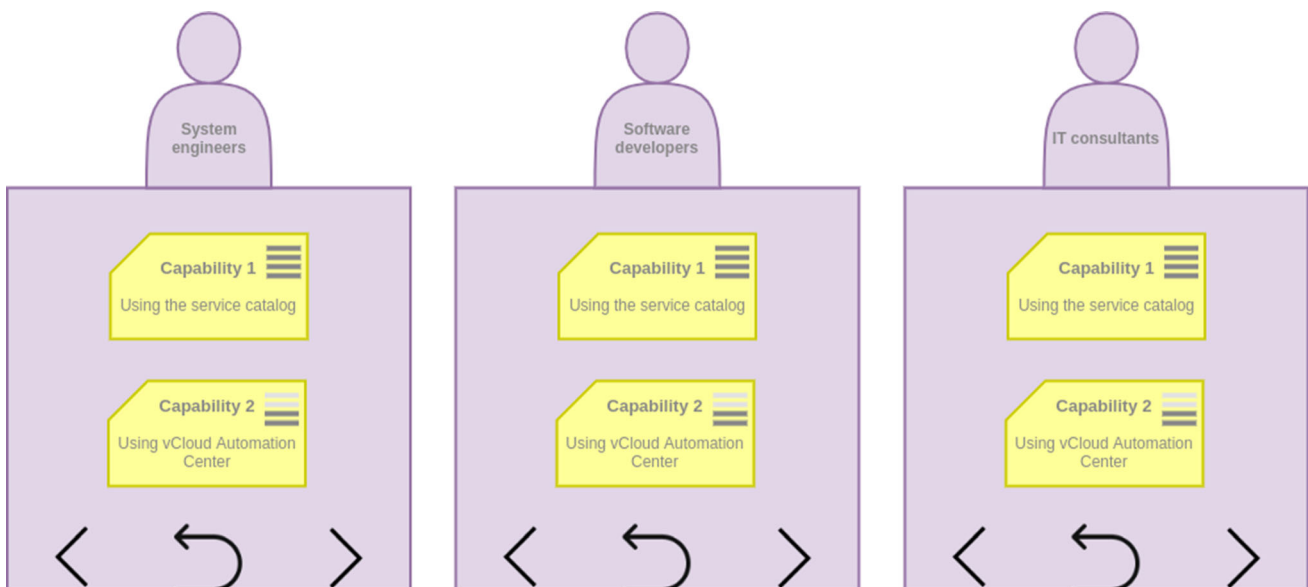

Fig. 9 Possession models for an example case. They are visual representations of the "has" relationship between the "internal customer" and "capability" concepts
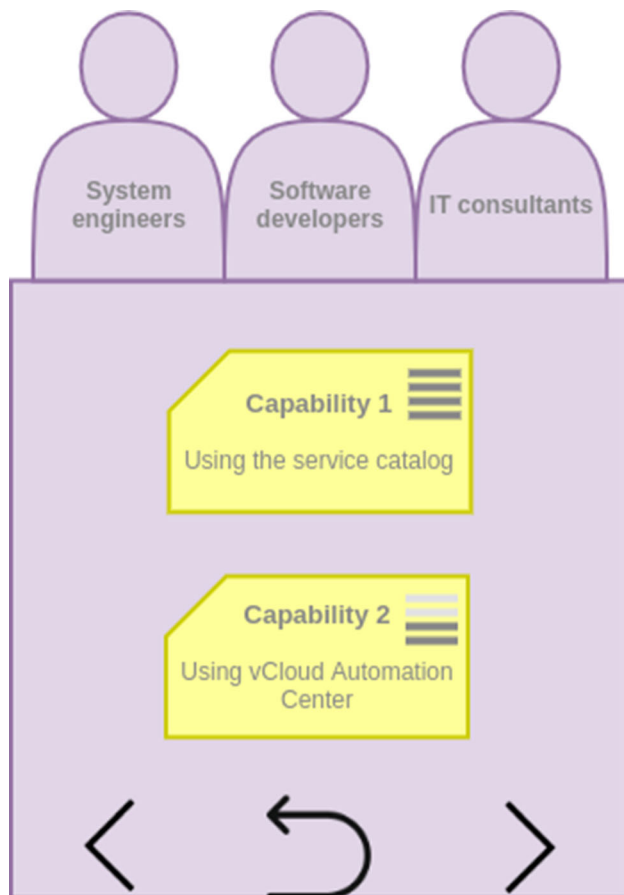
Fig. 10 A possession model for an example case. It visually represents the "internal customer group" concept

Furthermore, for each IT service, the outcome that is produced and the customization options for this outcome must be defined.

The IT service outcomes of the example case were VMs. The internal customers were able to customize the VMs, in terms of CPU, memory, and storage, at will. When providing a VM, IT resources (i.e., CPU, memory, and storage), which are owned by IT operations, are exchanged between IT operations and the internal customers. Therefore, the example case is the "purchase and transaction" type.

7.1.2 Structuring of Service Catalog

The service catalog manager must structure the initially completed service catalog. In this stage, the service catalog consists of a matrix containing the definitions of the IT services that can be requested by the internal customers. An example of a service catalog matrix (which includes the example case) is given in Table 2.

Not every IT service should be requestable by any internal customer. If IT operations have many different groups of internal customers (e.g., organizational functions and teams), there must be multiple service catalog views projected from the service portfolio (Hunnebeck 2013). A service catalog view enables one or more internal customer groups to request IT services that cannot be requested by other internal customer groups, and it defines the IT service outcome's customization options for the internal customer groups. The implementation of multiple service catalog views ensures that each internal customer can request only those IT services that are indispensable for the internal customer to do his or her daily work.

The example case could be requested by system engineers, software developers, and IT consultants.

7.1.3 Specification of Recurrent, Routine IT Services

The request fulfillment manager must specify the IT services in which a portion of the service tasks must be devolved to internal customers. The request fulfillment manager must classify the IT services defined in the service catalog (see Sect. 7.1.2) according to their scale, frequency, and whether they represent routines for the IT personnel.

The IT services that must be performed by IT operations frequently and for a large number of internal customers are responsible for the majority of the IT personnel's workload. Therefore, service task devolvement must be focused on such recurrent, routine IT services. Although other IT services can also be subject to service task devolvement, the ratio of potential benefits (i.e., workload reductions) to expenses (i.e., costs for designing and implementing the IT self-services) is especially high for recurrent, routine IT services.

Initially, the service tasks included in the example case had to be performed daily for many different internal customers by the IT personnel. Therefore, for the example case, IT operations decided to enable the internal customers to provide VMs on their own, thereby reducing the IT personnel's workload when it came to VM provisioning.

7.2 Determining IT Services

In this method component, each identified IT service must be modeled, and it must be specified which service tasks to devolve to the internal customers.

7.2.1 Specification of Service Tasks

The request fulfillment manager must model the recurrent, routine IT services (see Sect. 7.1.3). For each of these IT services, all the service tasks that allow the production of

Fig. 11 Icons representing the "self-service failure", "ambiguous information", "intentional misperformance", "excessive outcome production", and "forbidden outcome production" concepts

the IT service outcome must be specified. The graphical symbol representing a service task in an IT service model is depicted in Fig. 4.

The request fulfillment manager must specify the service tasks to be devolved to the internal customers by specifying, for each service task in the IT service, the actor category that must perform the service task. Actor categories are "internal customer", "onstage IT personnel", "backstage IT personnel/systems", "support IT personnel/systems", and "IT management". In an IT service model, these actor categories are separated by the "line of interaction", "line of visibility", "line of internal interaction", and "line of implementation" (Kingman-Brundage 1991; Fließ and Kleinaltenkamp 2004). The request fulfillment manager must also specify the physical evidence and IT systems with which the internal customers must interact in the IT service. In an IT service model, the physical evidence and IT systems are separated from the service tasks by another horizontal line. Their graphical symbols are presented in Fig. 7.

The IT service model for the example case is shown in Fig. 5. Because, in the IT service model, a portion of the service tasks is specified to be performed by the internal customers, the modeled example case is an IT self-service. In the example case, an internal customer was able to select a VM based on the installed operating system and to customize this VM – that is, to specify the amount of CPU, memory, and storage – at will. However, these IT resources were limited to IT operations and their increase came with cost. Therefore, after the customized VM had been requested, the IT system checked whether the specified amount of IT resources was still available to IT operations. If this was the case, the IT system provided the customized VM and it could be used by the internal customer.

7.2.2 Specification of Input and Outcome Details

The request fulfillment manager must model the IT service class (see Sect. 7.1.1) and specify the constraints and IT resources for using and producing the IT service outcome for each IT self-service (see Sect. 7.2.1). The IT service class must be specified at the top of an IT service model (see Fig. 5). Figure 6 depicts the icons that represent each IT service class.

The constraints for using the IT service outcome must be specified on the left side, separated by a vertical line, of an IT service model (see Fig. 5). If the internal customers are aware (unaware) of these constraints, they must be specified above (below) the "line of interaction". The IT resources required to produce the IT service outcome must be specified on the right side, separated by a vertical line, of an IT service model (see Fig. 5). IT resources provided and maintained by IT operations (internal customers) must be specified below (above) the "line of interaction". The graphical symbols for constraints and IT resources are depicted in Fig. 7.

The VMs that were produced in the example case could be used by the internal customers for training and testing purposes only, because of agreed license terms. However, the internal customers involved in the example case had only insufficient awareness of this constraint. The IT resources (i.e., CPU, memory, and storage) required for VMs in the example case were provided and maintained by IT operations. The internal customers involved in the example case were not directly charged for providing VMs. Hence, to them the example case was free.

7.2.3 Specification of Service Task Requirements

The request fulfillment manager must specify, for each IT self-service (see Sect. 7.2.1), the capabilities required for the internal customers to correctly perform the service tasks. These required capabilities must be specified in a requirement model. The requirement model for the example case is shown in Fig. 8.

In a requirement model, the request fulfillment manager must specify the level to which each capability is required for the devolved service tasks to be correctly performed by the internal customers. The level ranges from 0.25 to 1.0 in increments of 0.25. It is displayed in the upper right corner of the capability's graphical symbol (see Fig. 8). While a level of 1.0 represents required expert capability, a level of 0.25 refers to required basic capability. To identify the required capabilities and their levels, the request fulfillment manager must consult the IT personnel that has hitherto performed the service tasks.

To correctly perform the service tasks that were devolved to the internal customers in the example case, the internal customers had to possess expert knowledge and
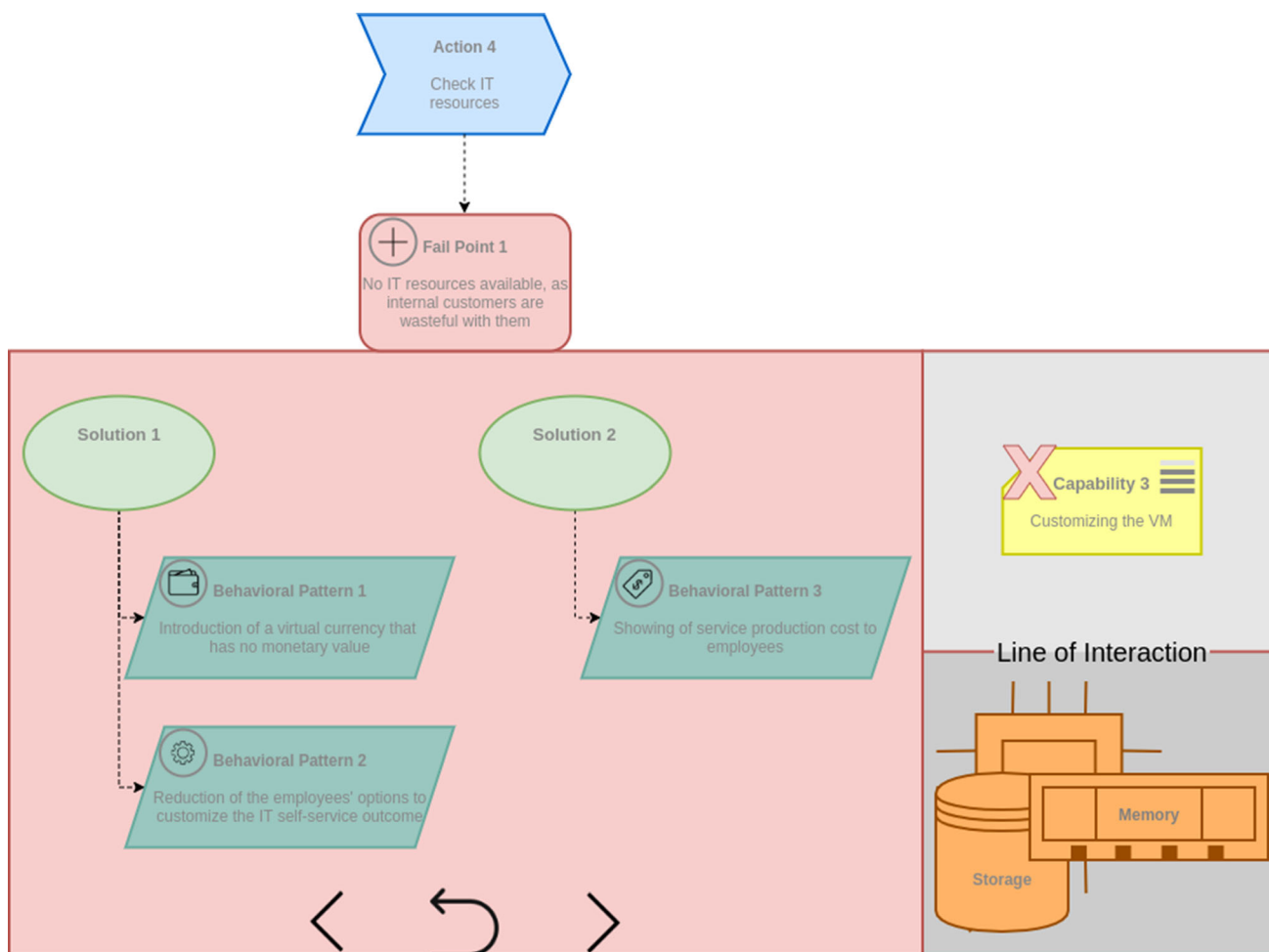
Fig. 12 A solution model for a fail point in an example case. It visually represents the "requires" relationship between the "fail point" and "solution" concepts and the "comprises" relationship between the "solution" and "behavioral pattern" concepts

basic knowledge about using the service catalog and the relevant IT system, respectively. In addition, based on the intended use case of a VM, the internal customers had to know about the appropriate customization of the VM.

7.3 Determine Internal Customers

DESERV IT holds that service task devolvement must be analyzed in light of the internal customers. Thus, in this method component, the internal customers must be aggregated into groups based on their capabilities to perform the service tasks to be devolved.

7.3.1 Specification of Internal Customers

The request fulfillment manager must specify, for each IT self-service (see Sect. 7.2.1), the capabilities possessed by the internal customer groups (see Sect. 7.1.2) that can request the IT self-service. However, for an internal

customer group, only those capabilities must be specified as being possessed that are required to correctly perform the devolved service tasks (see Sect. 7.2.3). The capabilities possessed by an internal customer group must be specified in a possession model. In Fig. 9, possession models for the example case are depicted.

In a possession model, not only the possessed capabilities must be specified, but also the level to which each capability is possessed by the internal customer group. This level ranges from 0.25 to 1.0 in increments of 0.25. It is displayed in the upper right corner of the capability's graphical symbol (see Fig. 9). A level of 1.0 refers to possessed expert capability. A level of 0.25 represents possessed basic capability. To identify the possessed capabilities and their levels, the request fulfillment manager must consult the business relationship manager and human resources manager.

The example case could be used by system engineers, software developers, and IT consultants. All these internal
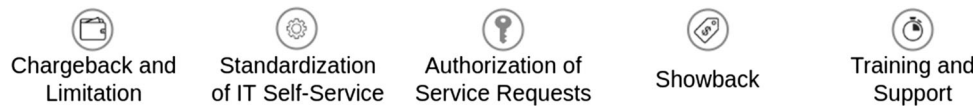
Fig. 13 Icons representing the "chargeback and limitation", "standardization of IT self-service", "authorization of service requests", "showback", and "training and support" concepts

customer groups shared an expert knowledge and basic knowledge about using the service catalog and the relevant IT system, respectively. However, none of these internal customer groups were capable of adequately customizing a VM. Hence, the required "capability 3" (see Fig. 8) is missing for all internal customer groups depicted in Fig. 9.

7.3.2 Aggregation of Internal Customers

The request fulfillment manager must aggregate, for each IT self-service (see Sect. 7.2.1), the internal customer groups that can request the IT self-service based on the possessed capabilities (see Sect. 7.3.1).

The request fulfillment manager must aggregate the internal customer groups in such a way that the internal customers within each resulting internal customer group are similar in terms of their capabilities to perform the devolved service tasks, but different across the groups. The decision about which internal customer groups to aggregate must be made by the request fulfillment manager in collaboration with the business relationship manager and IT personnel. The aggregation of internal customer groups must be documented by combining the possession models for the internal customer groups into a single possession model. Figure 10 shows a possession model that combines the possession models depicted in Fig. 9.

The internal customer groups involved in the example case shared the same capabilities at the same level. Therefore, in the example case, these internal customer groups were treated as an aggregated internal customer group.

7.4 Determining Service Task Devolvement

This method component is about finding solutions to failures that might occur in the resulting IT self-services for each internal customer group and analyzing the IT personnel's workload in the IT services.

7.4.1 Specification and Isolation of Fail Points

The request fulfillment manager must specify, for each IT self-service (see Sect. 7.2.1) and each internal customer group (see Sect. 7.3.2), where in the IT self-service failures might occur and what service tasks must be performed to

correct them. As a result of this procedure, an IT service model is created for each internal customer group (see Sect. 7.3.2).

At the design stage, failures in the execution of the IT service are modeled as fail points. The fail points relevant to service task devolvement and specification suggestions are listed in Table 3 (see Sect. 4).

While the graphical symbol representing a fail point in an IT service model is depicted in Fig. 4, the icons for distinguishing the different fail points are presented in Fig. 11. As shown in Fig. 5, these icons are displayed in the upper left corner of the fail point's graphical symbol.

In an IT service model, each fail point must be assigned to a service task whose performance might result in such a failure (see Fig. 5). In addition, to each fail point there must be assigned a set of service tasks that needs to be performed by the IT personnel to correct the failure (see Fig. 5).

In the example case, the occurrence of two failures must be assumed. First, because the example case was free for the internal customers, it must be assumed that the internal customers will excessively customize the VMs and hence that the limited IT resources may frequently be depleted. The probability of this failure's occurrence was further increased by the fact that the internal customers were not capable of adequately customizing VMs. The occurrence of this failure would require the IT personnel to reclaim IT resources regularly. Second, it must be assumed that the internal customers might provide and use VMs for purposes not allowed by the applicable license terms, because the internal customers were unaware of this constraint. To correct this failure, the IT personnel would have to regularly identify and delete VMs used for illegal purposes to prevent legal issues.

7.4.2 Specification of Solutions

The request fulfillment manager must specify, for each fail point in each IT service model (see Sect. 7.4.1), one or more solutions. Solutions specified for a specific fail point are alternatives to one another. Hence, only one of these solutions should be adopted by IT operations to prevent the failure from occurring.

A solution comprises the adoption of a set of behavioral patterns (see Sects. 6.2 and 4). In Table 4, we suggest

which behavioral patterns must be adopted to prevent a failure from occurring. Whether all, a combination, or only one of the suggested behavioral patterns should be adopted must be decided by the request fulfillment manager in consultation with the IT personnel.

The solutions for a failure must be specified in a solution model. Figure 12 presents a solution model for "fail point 1" (see Fig. 5) in the example case.

The graphical symbols representing solutions and behavioral patterns are depicted in Fig. 4. The icons for distinguishing the different behavioral patterns are depicted in Fig. 13.

In the example case, the internal customers excessively customized the VMs because they were not capable of adequately customizing VMs (see Figs. 8, 10, and the right side above the "line of interaction" in Fig. 12). Also, the example case was free (see Fig. 5 and the right side below the "line of interaction" in Fig. 12). For the example case, IT operations discussed the implementation of a charge-back system and the provisioning of only three standardized types of VMs as one possible solution. In addition, the presentation of the IT self-service costs in the service catalog was discussed as an alternative solution.

7.4.3 Analysis of the IT Personnel's Workload

The request fulfillment manager must consider, for each IT service model (see Sect. 7.4.1) and each combination of solutions (see Sect. 7.4.2), the execution of the IT self-service.

He or she has to consult the business relationship manager and IT personnel to specify the standard execution time of each service task and each behavioral pattern in the solutions specified for the IT self-service (see Sect. 7.4.2). As depicted in Fig. 5, the standard execution time of a service task must be specified below the service task's graphical symbol in an IT service model. Similarly, the standard execution time of a behavioral pattern needs to be specified below the behavioral pattern's graphical symbol in a solution model.

Because the solutions specified for a specific fail point are alternatives to each other (see Sect. 7.4.2), IT operations must choose between different combinations of solutions to be adopted for the IT self-service. In the example case, two fail points must be identified. For each fail point, two solutions were specified (see Fig. 12 for the solutions to "fail point 1"). As a result, for the example case, there were four possible combinations of solutions for IT operations to adopt. Therefore, there were four possible IT self-service designs for the example case.

Based on the standard execution times, the request fulfillment manager must calculate, for each IT self-service design, the IT personnel's workload. For an IT self-service,

the IT personnel's workload is the sum of the standard execution times of all service tasks to be performed by the IT personnel, except for the service tasks required to correct the identified failures, added to the sum of the standard execution times of the behavioral patterns forming the solutions combined by the IT self-service. Furthermore, the IT personnel's workload for a possible IT full-service and IT self-service with no adoption of any solution must be calculated as reference values. The former is the sum of the standard execution times of all service tasks, except for those service tasks required to correct the identified failures. The latter is the sum of the standard execution times of all services tasks to be performed by the IT personnel, including the service tasks required to correct the identified failures.

Table 5 provides the analysis of the IT personnel's workload for the example case at the design stage. Estimates of the standard execution times of the service tasks in the example case were based on direct observations and interviews made as part of the problem explication (see Sect. 4).

7.4.4 Assessment of Internal Customers' Intentions to Participate

The request fulfillment manager must consult the business relationship manager to decide, for each internal customer group (see Sect. 7.3.2), which of the IT services (see Sect. 7.4.3) to implement. Also, these two together must assess the internal customers' intentions to participate in the IT service, which should be implemented.

If the IT personnel's workload in an IT self-service, in which a combination of the identified solutions is adopted, is lower than that in the IT full-service, the IT personnel's workload in this IT self-service is reduced. Therefore, the request fulfillment manager must implement this IT self-service. If the IT personnel's workload is reduced in more than one of the possible IT self-services, the request fulfillment manager must choose one of these IT self-services for implementation. If the IT personnel's workload in the IT full-service is the lowest of the calculated workloads, the IT full-service and none of the IT self-services must be implemented. If the IT personnel's workload in the IT self-service, in which no solution is adopted, is the lowest of the calculated workloads, IT operations will not be required to adopt any solution. For the example case, IT operations decided to quit offering the IT self-service with no adopted solution and instead to offer it as an IT full-service until it had been decided which of the discussed IT self-services (1–4 in Table 5) should be implemented.

The internal customers' performance of devolved service tasks is the prerequisite for reducing the IT personnel's workload in IT self-services. Therefore, it is

important to assess the internal customers' intentions to participate in IT self-services at the design stage to prevent cost-intensive implementations of IT self-services in which the IT personnel's workload is not reduced.

8 Evaluate Artifact

The design process of DESERV IT was iterative: in each design cycle we designed a version of DESERV IT that was evaluated, and the results of this evaluation served as the input for the next design cycle. Table 6 provides an overview of the evaluation episodes.

Because the application of DESERV IT to the example case demonstrated the fulfillment of MR1–MR4 (see Sect. 7), in the evaluation episodes we evaluated DESERV IT's perceived effectiveness, usefulness, completeness, and generalizability (i.e., MR5–MR8).

In the following sections, we describe the evaluation episodes and the evolution of DESERV IT in more detail.

8.1 Evaluation Episode 1

8.1.1 Context

From November 2017 to January 2018, the main author worked for an international IT consulting firm that has its headquarters in Tokyo (Japan) and employs about 120,000 people worldwide. This circumstance allowed us to participate in an action research.

The IT consulting firm decided to develop its own cloud computing platform for managed services. The platform was developed and operated by a team of DevOps engineers, including the main author. The operation of the platform required the DevOps engineers to perform a set of IT services upon customer request. To free themselves from doing this themselves, from November 2017 to January 2018 the DevOps engineers had to decide which tasks of these IT services should be delegated to second-level system engineers.

8.1.2 Observations

To support the service task devolvement, the primary author presented the team with a first draft of DESERV IT's framework, which included only the three method components of "determine IT services", "determine internal customers", and "determine service task devolvement". Although the team agreed that the framework includes most of the relevant method components, it began its actions by defining the IT services (e.g., restore corrupted log data, adding new nodes to the cluster, and specification of new alerts for monitoring) that can be

requested by the customers and which are therefore offered for the cloud computing platform. From these IT services, those that had been performed on a regular basis were identified. For each of these recurrent, routine IT services (e.g., redeployment of containers, renewing outdated certificates, and creating new user accounts for the platform), the service tasks were specified. Based on these specifications, the IT services (e.g., redeployment of containers, adding new nodes to the cluster, and creating new user accounts for the platform) that could be performed by novice engineers were identified and devolved to the second-level system engineers. As a result, the DevOps engineers perceived their workload in the devolved IT services as reduced.

8.1.3 Method Improvements

To specify our learning from the participatory action research, we added the determination of a service catalog as the first method component to the method framework (see Fig. 2).

8.2 Evaluation Episode 2

8.2.1 Context

Based on the demonstration of DESERV IT in the example case (see Sect. 7) and on informed arguments, the cognitive effectiveness of a first version of DESERV IT's visual notation was evaluated.

In addition, the second version of the framework of DESERV IT (i.e., the result of evaluation episode one) was discussed with scholars and practitioners in the enterprise modeling and service operations discipline at workshops and conferences.

8.2.2 Observations

The evaluation of the cognitive effectiveness of the visual notation is summarized in "Online Appendix H". At enterprise modeling and service operations conferences and workshops, scholars and practitioners argued for a determination cycle, including the method components "determine internal customers" and "determine service task devolvement", because people's capabilities evolve over time.

8.2.3 Method Improvements

According to the scholars' and practitioners' feedback, we added the suggested determination cycle, including the "determine internal customers" and "determine service

task devolvement" method components, to the method framework (see Fig. 2).

8.3 Evaluation Episode 3

8.3.1 Context

To evaluate DESERV IT's effectiveness, usefulness, and generalizability as perceived by practitioners, a focus group was interviewed. The focus group consisted of five ITSM experts employed at a German office of the IT consulting firm in which the evaluation episode one also took place. The focus group consisted of two service managers, a senior managing consultant, the head of application management, and the head of ITSM. The focus group was selected because it included potential adopters of DESERV IT.

The interview was held in a meeting room of the IT consulting firm's German office and lasted about 60 min. Apart from two interviewees who joined the interview via Skype, all interviewees were personally present. The demonstration of DESERV IT for the example case, as discussed in Sect. 7, was presented to all the interviewees, and, based on that, the interviewees were asked whether they perceived DESERV IT as effective, useful, and generalizable.

8.3.2 Observations

Overall, the focus group perceived DESERV IT to be effective and generalizable in the ITSM context. More precisely, IT services of the "purchase and transactions" class, such as the provisioning of VMs and containers required in DevOps projects, were identified as appropriate use cases for the method. The focus group did not perceive that there were too many graphical symbols in DESERV IT's visual notation, and they perceived the symbols to be clear. However, several suggestions were made to increase DESERVE IT's perceived usefulness and the visual notation's cognitive effectiveness:

- The capabilities required to perform the devolved service tasks must be specified for the whole IT self-service, not for each service task individually.
- The visual notation of the presented version of DESERV IT required the specification of the IT service class in the upper left corner of the IT service model in the same section where the constraints are specified. The interviewees argued for a separation of IT service class and constraint specification.
- To better analyze the impact of and reasons for a potential failure, the interviewees suggested that there should also be a specification of the service task whose

performance might result in the failure and of the causes of the potential failure in the solution model. In the presented version of DESERV IT, solution models did not provide this information.

8.3.3 Method Improvements

Based on these suggestions, the procedures and visual notation of DESERVE IT were redesigned as follows:

- The capabilities required to perform the devolved service tasks are specified in a single requirements model for the whole IT self-service (see Fig. 8).
- The IT service class is specified at the top of an IT service model, and only the constraints are specified in the left section of an IT service model (see Fig. 5).
- In a solution model, the service task whose performance might result in the failure and the cause of the failure are specified (see Fig. 12).

8.4 Evaluation Episode 4

8.4.1 Context

A final redesign of DESERV IT was conducted based on feedback from interviewing a managing consultant, an enterprise architect, and a managing enterprise architect. The three experts were employed at a German office of an international IT consulting firm that has its headquarters in Paris (France) and employs about 208,800 staff worldwide. The experts were selected because they represent potential adopters of DESERV IT.

Before interviewing each expert, a summary of the demonstration of DESERV IT (see Sect. 7) was sent to the expert. During the interviews, DESERV IT was demonstrated for the example case (see Sect. 7) and the experts were asked whether they perceived DESERV IT as effective, useful, and generalizable. Information about the interviews is presented in Table 7.

8.4.2 Observations

The experts perceived DESERV IT as effective, useful, and generalizable. In particular, one expert noted that the method would have helped him in designing an IT self-service for VM provisioning in a prior project.

However, for future research, the experts suggested that DESERV IT be extended so that it could also consider the costs of modeling IT services, analyzing service task devolvement, and implementing resulting IT self-services. Although the meanings of the graphical symbols in the visual notation were clear to the experts after

demonstrating DESERVE IT for the example case, there were two suggestions to increase the cognitive effectiveness of the visual notation:

- Labels should be added to the horizontal lines in an IT service model, because in the presented version of DESERV IT such labels were not included.
- The graphical symbols used by the visual notation for IT resources in the presented version of DESERV IT reminded the experts of databases, and it was suggested that these be altered.

8.4.3 Method Improvements

Based on the experts' feedback, DESERV IT was improved in the following ways:

- The horizontal lines in IT service models are labeled (see Fig. 5).
- Three new graphical symbols for CPU, memory, and storage are used by the visual notation to represent these IT resources (see Fig. 7).

9 Discussion and Concluding Remarks

9.1 Related Work and Contributions to Theory and Practice

With this research, we have contributed to enterprise modeling. DESERV IT is the first method that supports the modeling and analysis of IT services at a level required for making well-founded decisions about service task devolvement. To support the analysis of service task devolvement (see MR1), a method must support the modeling of the required and possessed capabilities (see MR2) as well as service outcome production, including the service class and required resources (see MR3), and solutions that prevent failures from occurring in the service (see MR4). In a systematic literature review (Baer et al. 2016), we identified only seven methods supporting the modeling and analysis of the inputs of services. The shortcomings of these methods regarding MR2 are summarized in "Online Appendix I".

Because the target variables addressed by the existing methods for modeling and analyzing services are service profitability, service quality, efficiency, net benefit, goal realization, and perceived value (e.g., Gersch et al. 2011; Trkman et al. 2015), the methods do not support the modeling of any service class, of constraints on using the service outcome, and of resources required to produce the service outcome – that is, they do not fulfill MR3. Service blueprinting supports the modeling and isolation of fail points in services – that is, it partially fulfills MR1 – but it does not support the modeling and analysis of solutions to the failures (see MR4).

DESERV IT supports request fulfillment managers in modeling and analyzing IT services. It therefore adds to ITIL's request fulfillment process that suggests the modeling and analysis of IT services, but ITIL does not give any advice for this (Steinberg et al. 2013). More precisely, by supporting the analysis of service task devolvement, DESERV IT supports request fulfillment managers in devolving service tasks from IT operations to software development and vice versa. Hence, DESERV IT supports the integration of these two organizational functions and thereby the implementation of DevOps. In designing DESERV IT, we have contributed to practice, because DESERV IT supports a more agile ITSM and operations which have become increasingly relevant due to recent trends in software development and IT operations.

9.2 Limitations and Future Research

DESERV IT was designed with a focus on IT self-services – that is, it supports IT operations in devolving recurrent, routine service tasks to internal customers in IT services. However, the meta-model and visual notation of DESERV IT were designed in such a way that additional fail points, behavioral patterns, and adequate graphical symbols and icons for these new concepts can be easily added to the method. Future research should conduct exploratory multiple-case studies to explore fail points and behavioral patterns relevant to self-services in other domains (e.g., finance and retail); moreover, it should attempt to extend DESERV IT's meta-model and visual notation with concepts to make the method applicable not only to IT self-services, but also to self-services in other domains.

In this research, we have evaluated DESERV IT's effectiveness, usefulness, completeness, and generalizability, as perceived by experts in the ITSM and enterprise architecture discipline (see Sect. 8). However, an evaluation of the actual effectiveness, usefulness, completeness, and generalizability of DESERV IT based on quantified data is missing. Therefore, future research is required to apply the latest version of DESERV IT in naturalistic settings to evaluate the actual, not just perceived, fulfillment of the four generic environmental method requirements. Such future research could take the form of case studies and action research, and it must be long term, because the evaluation should begin at the design stage by applying DESERV IT to evaluate the actual usefulness and completeness, and end at the execution stage when enough data has been collected to evaluate the actual effectiveness and generalizability. In addition, such future research should enable evaluation not only of the actual

effectiveness, usefulness, completeness, and generalizability of DESERV IT, but also of the actual efficiency – that is, DESERV IT's effectiveness without wasting the time, effort, or expense of the participating roles (see "Online Appendix G").

According to the feedback obtained from the experts in evaluation episode 4 (see Sect. 8.4), future research should extend DESERV IT by a cost–benefit analysis or integrate it into an investment appraisal. Such an extended or integrated version of DESERV IT should address a cost–benefit ratio as the target variable, which reflects the costs of applying DESERV IT and the monetary rating of a reduction of the IT personnel's workload in the IT self-service are reflected. Future research should also evaluate such an extended or integrated version of DESERV IT in the light of the generic environmental method requirements (see Table 1).

References

Baer F, Sandkuhl K, Schmidt R (2016) On the maturity of service process modeling and analysis approaches. In: Abramowicz W, Alt R, Franczyk B (eds) Business information systems. Springer, Cham, pp 356–367

Baer F, Leyer M, Sandkuhl K (2018) Reducing the IT personnel's workload in IT self-services. SMR J Serv Manag Res 2(4):22–35. https://doi.org/10.15358/2511-8676-2018-4-22

Baskerville RL (1999) Investigating information systems with action research. Commun AIS 2(1):2–32

Becker J, Beverungen D, Knackstedt R, Matzner M, Muller O, Poppelbuss J (2013) Bridging the gap between manufacturing and service through IT-based boundary objects. IEEE Trans Eng Manag 60(3):468–482. https://doi.org/10.1109/TEM.2012.2214770

Bitner MJ, Ostrom AL, Meuter ML, Clancy JA (2002) Implementing successful self-service technologies. Acad Manag Exec 1993–2005 16(4):96–109

Brinkkemper S (1996) Method engineering: engineering of information systems development methods and tools. Inf Softw Technol 38(4):275–280. https://doi.org/10.1016/0950-5849(95)01059-9

Campbell CS, Maglio PP, Davis MM (2010) From self-service to super-service: a resource mapping framework for co-creating value by shifting the boundary between provider and customer. Inf Syst E Bus Manag 9(2):173–191. https://doi.org/10.1007/s10257-010-0142-4

Collier JE, Kimes SE (2012) Only if it is convenient: understanding how convenience influences self-service technology evaluation. J Serv Res 16(1):39–51. https://doi.org/10.1177/1094670512458454

Ellway B (2016) Design vs practice: how problematic call routing and rerouting restructures the call centre service system. Int J Oper Prod Manag 36(4):408–428. https://doi.org/10.1108/IJOPM-11-2013-0487

Fließ S, Kleinaltenkamp M (2004) Blueprinting the service company: managing service processes efficiently. J Bus Res 57(4):392–404. https://doi.org/10.1016/S0148-2963(02)00273-4

Gersch M, Hewing M, Schöler B (2011) Business process blueprinting: an enhanced view on process performance. Bus Process Manag J 17(5):732–747. https://doi.org/10.1108/14637151111166169

Globerson S, Maggard MJ (1991) A conceptual model of self-service. Int J Oper Prod Manag 11(4):33–43. https://doi.org/10.1108/01443579110002977

Goldkuhl G, Lind M, Seigerroth U (1998) Method integration: the need for a learning perspective. IEE Proc Softw 145(4):113–118. https://doi.org/10.1049/ip-sen:19982197

Harrison T, Waite K (2015) Impact of co-production on consumer perception of empowerment. Serv Ind J 35(10):502–520. https://doi.org/10.1080/02642069.2015.1043276

Haumann T, Güntürkün P, Schons LM, Wieseke J (2015) Engaging customers in coproduction processes: how value-enhancing and intensity-reducing communication strategies mitigate the negative effects of coproduction intensity. J Mark 79(6):17–33. https://doi.org/10.1509/jm.14.0357

Henderson-Sellers B, Ralyté J, Ågerfalk P, Rossi M (2014) Situational method engineering. Springer, Heidelberg

Hilton T, Hughes T, Little E, Marandi E (2013) Adopting self-service technology to do more with less. J Serv Mark 27(1):3–12. https://doi.org/10.1108/08876041311296338

Hunnebeck L (2013) ITIL: IT service management practices. 2: service design, 2nd edn. TSO, The Stationery Office, London

Johannesson P, Perjons E (2014) An introduction to design science, 2014th edn. Springer, New York

Jouck T, Depaire B (2018) Generating artificial data for empirical analysis of control-flow discovery algorithms: a process tree and log generator. Bus Inf Syst Eng 61(6):695–712. https://doi.org/10.1007/s12599-018-0541-5

Kazemzadeh Y, Milton SK, Johnson LW (2015) A conceptual comparison of service blueprinting and business process modeling notation (BPMN). Asian Soc Sci 11(12):307–318. https://doi.org/10.5539/ass.v11n12p307

Kingman-Brundage J (1991) Technology, design and service quality. Int J Serv Ind Manag 2(3):47–59. https://doi.org/10.1108/EUM0000000002806

Kumar A, Telang R (2012) Does the web reduce customer service cost? Empirical evidence from a call center. Inf Syst Res 23(3):721–737. https://doi.org/10.1287/isre.1110.0390

Leyer M, Tate M, Baer F, Kowalkiewicz M, Rosemann M (2017) Customer acceptance of pro-active services: a scenario-based empirical study. In: ECIS 2017 Proceedings, Guimarães, pp 1837–1852

Marrone M, Kolbe LM (2011) Impact of IT service management frameworks on the IT organization. Bus Inf Syst Eng 3(1):5–18. https://doi.org/10.1007/s12599-010-0141-5

Milton SK, Johnson LW (2012) Service blueprinting and BPMN: a comparison. Manag Serv Qual Int J 22(6):606–621. https://doi.org/10.1108/09604521211287570

Moody DL (2009) The "physics" of notations: towards a scientific basis for constructing visual notations in software engineering. IEEE Trans Softw Eng 35(5):756–778

Moody DL (2011) Why a diagram is only sometimes worth a thousand words: an analysis of the BPMN 2.0 visual notation. https://pdfs.semanticscholar.org/dc09/25bd6d879f6b867806f8badfc70d2e30b4a4.pdf. Accessed 11 May 2020

Pastor O, Ruiz M, Koç H, Valverde F (2018) Capability-based communication analysis for enterprise modelling. Enterp Model Inf Syst Archit EMISAJ 13(4):1–24. https://doi.org/10.18417/emisa.13.4

Ralyté J, Backlund P, Kühn H, Jeusfeld MA (2006) Method chunks for interoperability. In: Embley DW, Olivé A, Ram S (eds) Conceptual modeling—ER 2006. Springer, Heidelberg, pp 339–353

Roche J (2013) Adopting DevOps practices in quality assurance. Commun ACM 56(11):38–43

Sampson SE (2012) Visualizing service operations. J Serv Res 15(2):182–198. https://doi.org/10.1177/1094670511435541

Scherer A, Wünderlich NV, von Wangenheim F (2015) The value of self-service: long-term effects of technology-based self-service usage on customer retention. MIS Q 39(1):177–200

Seigerroth U (2011) Enterprise modeling and enterprise architecture: the constituents of transformation and alignment of business and IT. Int J ITBus Alignment Gov IJITBAG 2(1):16–34. https://doi.org/10.4018/jitbag.2011010102

Shostack GL (1982) How to design a service. Eur J Mark 16(1):49–63. https://doi.org/10.1108/EUM0000000004799

Steinberg R, AXELOS Limited, Stationery Office, Großbritannien (eds) (2013) ITIL: IT service management practices. 4: service operation, 2011 edn, 2nd impr. TSO, The Stationery Office, London

Trkman P, Mertens W, Viaene S, Gemmel P (2015) From business process management to customer process management. Bus Process Manag J 21(2):250–266. https://doi.org/10.1108/BPMJ-02-2014-0010

Venable J, Pries-Heje J, Baskerville R (2012) A comprehensive framework for evaluation in design science research. In: Peffers K, Rothenberger M, Kuechler B (eds) Design science research in information systems: advances in theory and practice. Springer, Heidelberg, pp 423–438

Yalley AA, Sekhon HS (2014) Service production process: implications for service productivity. Int J Product Perform Manag 63(8):1012–1030. https://doi.org/10.1108/IJPPM-10-2012-0113

Zaza I, Junglas I (2016) IT self-service engagement: a theory of trying perspective. In: ICIS 2016 Proceedings, Dublin