Modular Service Design of Information Technology-enabled Services

Tuure Tuunanen[a]
Markus Salo[b]
Ms. Feng Li[c]

[a] University of Jyväskylä, Faculty of Information Technology, Mattilanniemi 2, P.O. Box 35, FI-40014, Jyväskylä, Finland, p. +358 40 805 4628, tuure@tuunanen.fi

[b] University of Jyväskylä, Faculty of Information Technology, Mattilanniemi 2, P.O. Box 35, FI-40014, Jyväskylä, Finland, p. +358 40 8054295, markus.t.salo@jyu.fi

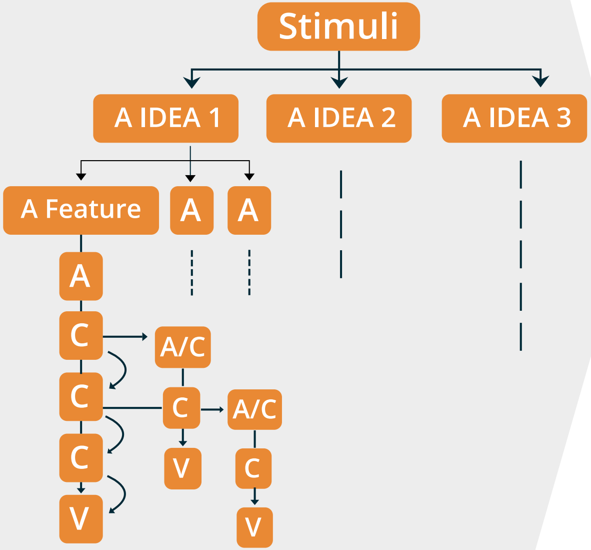
[c] University of Auckland Business School, 12 Grafton Road Sir Owen G Glenn Building, 1010, New Zealand, angelafengli@gmail.com

Keywords: service modularization, information-technology-enabled services, design principles, modular service design method.

Design Principle (DP) based Modular Service Design: A Qualitative Approach

Step 1

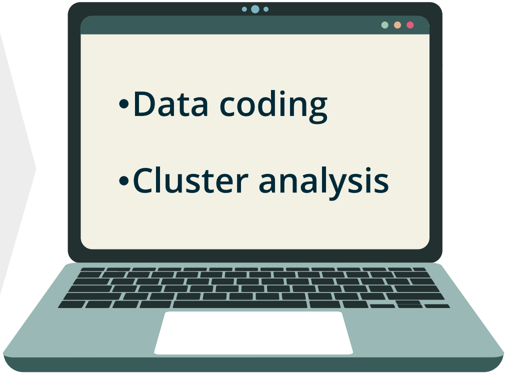
Laddering interviews



Data collected from 6 New Zealand companies
(25 interviews, 227 recorded laddering chains)

Step 1

Data analysis



- Data coding
- Cluster analysis

Development of DPs

Prioritized DPs

Computational

Flexible information integration

Adaptive

Customer-centricity and usability

Collaborative

Knowledge sharing and accessibility

Network

Improving quality of service

MODULAR SERVICE DESIGN OF INFORMATION-TECHNOLOGY-ENABLED SERVICES

Executive Summary

Recent research on service design has studied how to design information-technology enabled services (ITeS) and the related challenges. However, researchers have found designing ITeS increasingly complicated and challenging. ITeS are not only technological solutions but moreover a socio-technical phenomenon in which services are designed and delivered using all available means to realize value for both providers and service users. Meanwhile, the service modularization research has more specifically attempted to address ITeS design challenges by proposing different ways to reuse, substitute, or vary service elements. However, researchers have not yet been able to operationalize these concepts to support practical use by service designers or managers. Thus, the current service modularization approaches have not been widely adopted in the industry despite the promise of enabling companies to offer more service variety, improved flexibility, simplified complex systems, enhanced quality, and cost savings.

Accordingly, we argue that new approaches are needed to support modular service designs and, specifically, the design of ITeS. Specifically, we propose that the concept of design principles (DPs) is a way to design modular services efficiently. DPs have already been widely used in design-focused research in information systems. DPs are generalizable design guidelines and abstractions that can be applied to develop service-specific solutions. However, the prior research has yet to offer service design methods to support the development of DPs. With our qualitative research study with industry practitioners (25 in-depth interviews with six ITeS developer firms) in New Zealand, we develop exemplar DPs and a set of prioritized DPs for ITeS. We demonstrate how complex service systems, specifically ITeS, can be modularly designed. Our DPs show how different ITeS design elements or service attribute combinations impact the outcome-driven design of service experience.

As the key findings, we present a Modular Service Design (MSD) method that adopts DPs to create effective modular ITeS designs based on the findings. The MSD method is ready for use by service designers. The firms can use the MSD method to collect data and derive DPs that offer a concise overall view of the most central design aspects for their ITeS. We also suggest to first recognize the standard features for service modularization. These features can then be later customized and combined with others. DPs can be used to identify standard, customized, or combined service features.

Our MSD method also provides rich information about the actual or perceived use of different types of ITeS. These details can be utilized in tandem with the service modularization approach; developers and providers can look at each service module and evaluate how well it supports the IT-enabled service in question. In this way, service designers and managers can more accurately identify developmental areas and weaknesses in their service concept and delivery systems at a feature or feature set level to improve the customers' service experience and design better services. Our method can also be modularized itself, and it can be adapted to fit the company's work practices and specific projects.

MODULAR SERVICE DESIGN OF INFORMATION-TECHNOLOGY-ENABLED SERVICES

Abstract: The literature has proposed ways to modularize information-technology-enabled services (ITeS) with limited success. We argue that applying design principles (DPs) can address this gap and revitalize the service modularization literature. With a qualitative research study, we develop exemplar DPs and a set of prioritized DPs for ITeS. We contribute to the literature by demonstrating how complex service systems, specifically ITeS, can be modularly designed. Our DPs show how different ITeS design elements or service attribute combinations impact the outcome-driven design of service experience. Based on the findings, we present a modular service design framework and a service design method that adopts DPs to create effective modular ITeS designs. We also offer ways to conceptualize and apply service modularization to improve the adoption of the modular service design by service designers and managers.

Keywords: service modularization, information-technology-enabled services, design principles, modular service design method

1. INTRODUCTION

The increasing deployment of technology is altering how individuals interact with organizations; the traditional goods-centric paradigm is now being challenged by the acceleratory growth of service-dominant economic activities that use information-technology-enabled services (ITeS) (Lusch & Nambisan, 2015; Peters et al., 2016; Vargo & Lusch, 2004). This recent trend has amplified technology-driven enthusiasm, as demonstrated by the explosive growth in customization, integration, intelligence, and globalization. However, this kind of enthusiasm makes designing ITeS increasingly complicated and challenging. ITeS are essentially a socio-technical phenomenon in which services are designed and delivered using all available means to realize value for both providers and service users (Grönroos, 2006, 2007; Grönroos & Voima, 2013).

We define ITeS as *any type of service-based activity that utilizes IT to satisfy users' needs and requirements during the consumption process.* Examples of ITeS include self-service solutions, such as online banking or shopping services; software as a service, such as online accounting software; and personal mobile services, such as mobile applications. It is worth mentioning that "IT service" is a sub-concept of ITeS. An IT service is generally defined as complex, knowledge-based work that aims to provide business clients with quality service (Ronnie et al., 2008). Often, this simply involves technical solutions developed by IT industries. In contrast, ITeS include a broader range of services mediated by IT but that are not necessarily only provided digitally.

The recent service design literature has studied how to design ITeS (see, e.g., Maglio et al., 2009; Patrício et al., 2011; Teixeira et al., 2017). For example, Teixeira et al. (2017) examined the challenges related to designing ITeS. Meanwhile, the service modularization literature (see, e.g., Bask et al., 2010; Dörbecker & Böhmann, 2013; Tuunanen & Cassab, 2011; Voss & Hsuan, 2009) has more specifically attempted to address ITeS design challenges by proposing different ways to reuse, substitute, or vary service elements. However, the service modularization literature has not yet been able to operationalize these concepts for wide adoption by service design researchers or practitioners. For example, Bask et al. (2014) defined a modular service offering as consisting of a standardized base service(s), customized service(s), or combinations thereof. Tuunanen and Cassab (2011), in turn, proposed a way to modularize the service process using concepts derived from software development, namely reuse and variation of service processes. Neither approach has been adopted in practice, and it seems that the service modularization literature has stagnated in recent years. This is a concern as the perceived view is that service modularization allows companies to offer more product variety, improved flexibility, simplified complex systems, enhanced quality, and cost savings (Jose & Tollenaere, 2005; van Liere et al., 2004).

Accordingly, we argue that new approaches are needed to support modular service designs and, specifically, the design of ITeS. Specifically, we propose that the concept of design principles (DPs) is a way to efficiently design modular services. DPs have been widely used in design-focused research on information systems (see, e.g., Hevner et al., 2004; Markus et al., 2002; Walls et al., 1992). In this study, we define DPs as generalizable design guidelines and abstractions that can be applied to develop service-specific solutions. However, the prior literature has yet to offer service design methods to support the development of DPs. We believe that ITeS provide an excellent setting to study this. ITeS are based on digital solutions, which are today designed modularly.[1]

In the information systems literature (Gregor & Jones, 2007; Hevner et al., 2004; Markus et al., 2002; Walls et al., 1992), DPs have been used to theorize about and generalize the design of IT solutions. Walls et al. (1992) proposed that meta-level user requirements can be used to depict the meta-designs of IT solutions and that such meta-designs can be used to theorize further and generalize efforts. Later, Gregor (2002) described DPs as design decisions and design knowledge intended for manifestation or encapsulation in an IT solution. Following

[1] Modern programming languages are based on object-oriented programming that enables modular system design where objects of a system interact with each other. A more recent adaptation of the same concept is micro services, where a collection of small solutions forms a working service system. Netflix, for example, has adopted this approach to deliver its streaming services.

Sein et al. (2011), in our study, we apply Mathiassen and Sørensen's (2008) framework for defining the different types of DPs: adaptive, collaborative, computational, and network.

Accordingly, our research objective is to examine how DPs can be identified and formulated and whether this approach can be formalized as a modular service design (MSD) method for ITeS. Based on this, we developed our research question: *How can DPs be identified and formulated to support MSD and, more specifically, for ITeS?* We apply a qualitative research approach to answer this. The data collection for this study was done in collaboration with six New Zealand-based companies that develop ITeS, and we interviewed 25 persons who participated in ITeS development. We used cluster analysis with the data to develop exemplar DPs for modular ITeS design and present how to prioritize these for MSD.

Our study contributes to existing knowledge by proposing a new definition for modular service offerings to renew the service modularization literature and an MSD framework, which can be used to develop new MSD methods. We operationalize the framework to develop the primary output of the study, an MSD method, that can be applied to the design of ITeS. The developed MSD method is ready for use by practitioners. We demonstrate how complex service systems, specifically ITeS, can be modularly designed. We also suggest that it is important to consider why specific service design methods work, which the extant literature typically does not discuss. Our study offers an exemplar of how to develop a theory-based and driven service design method. This opens new ways to advance the development of service design methods and the practice of service design.

The following section reviews the literature on ITeS, service design and modularization, and the four general types of DP for ITeS. After that, our research methodology and analysis are presented, followed by the findings. Finally, we conclude by discussing the findings and limitations of the current work and topics for future research.

2. RATIONALE FOR MSD OF ITeS

Services are becoming increasingly important for organizations. Service, as a concept, refers to applying specialized knowledge, skills, and experience to co-create value for both the service user and the provider (Grönroos, 2008; Grönroos & Voima, 2013; Vargo & Lusch, 2004). Recently, there has been a shift toward "a world in which value is the result of an implicit negotiation between the individuals and the firm" (Prahalad & Ramaswamy, 2004, p. 7). Enabled by ITeS, the interactions between customers and organizations strongly affect the way companies operate and compete in the market (Vargo & Lusch, 2008).

However, the extant service research literature does not sufficiently accommodate the needs related to designing ITeS. Berry and Lampo (2000) highlighted the need to consider how service (re)design should be performed to make services appealing and satisfying. For this purpose, they proposed a framework for different service (re)design approaches that may help to create innovative new services or rejuvenate existing ones. This is a typical approach to service design in the literature. Authors often offer different methods for designing services or service systems (Maglio et al., 2009). However, they do not explain why this method would provide better utility or value to service designers than other methods.

Yu and Sangiorgi (2014, 2018) proposed an approach that can provide a foundation for comparing service design methods. Yu and Sangiorgi (2018, p. 41) defined service design as "an integrative approach to collaborative and cross-disciplinary service innovation." This definition is based on the design-centered approach to service design (cf. Mager, 2008). Yu and Sangiorgi conceptualized the role of service design in value co-creation and new service development. Yu and Sangiorgi (2014, 2018) divided the approach in terms of focus to how services are designed (service design and analysis) and how services are implemented (service development and implementation). This argument is summarized in Table 1.

The framework below depicts objects of service design and links these to different phases of the process. The left side focuses on the value, form, and function of the service and the service experiences and outcomes, while the right side looks at the structure, infrastructure, and process of the service delivery system. Yu and Sangiorgi (2014) also described some facilitators for service design, such as different methods and tools, staff and customer involvement in the service design process, and how the organization's characteristics may impact the service design work.

[Insert Table 1. The service design framework, adapted from Yu and Sangiorgi (2014, 2018).]

Our review of service design method literature,[2] which addressed how service design methods manage the integration of "design and analysis" and "development and implementation," found two methods that, at some level, achieve this goal: the multilevel service design method (Patrício et al., 2011) and the management and interaction design for service (MINDS) (Teixeira et al., 2017). The authors looked at different levels of service design and proposed how to apply well-known service design methods (or facilitators), such as blueprinting (Bitner et al., 2008; Shostack, 1984) or value constellation mapping (Michel et al., 2008), to understand the potential (or perceived) value for a developed service. The authors also highlighted the importance of the technological aspects of the design (e.g., the service delivery system). The data collection was done with semi-structured interviews with customers and other actors related to the developed services. In a more recent study to prototype a service, Teixeira et al. (2017) integrated more traditional service design methods with those commonly used for IT design, such as affinity diagrams, system navigation mapping, and user experience blueprinting. In Table 2, we have summarized the MINDS service design method using the Yu

[2] We reviewed the volumes of *Journal of Service Research* published between 2000 and 2020.

and Sangiorgi (2014) framework. We can see that MINDS supports developing a service concept and encounter design by focusing on the form and functions of the service.

MINDS pays attention to the service experience design with a specific version of blueprinting and combines that with user interface/wireframe sketching. Similarly, MINDS looks at the service delivery system from the user's navigation process and structure, but the method does not directly support infrastructure design. In addition, the method currently does not provide support contextualizing the use of the method to different organizational settings and work cultures.

[Insert Table 2. The illustration of the MINDS service design method.]

However, while the multilevel service design method (Patrício et al., 2011) and MINDS (Teixeira et al., 2017) are a step toward the integration of design and analysis (of a service) and development and implementation (of a service delivery system), we argue that the authors still only investigated instantiations of service systems and, more specifically, how the selected service design methods enabled the design of these systems. What is lacking, and more general in the extant service design literature, is a way to integrate the service concept and encounter better service delivery system design, which becomes more and more important when designing ITeS. Here, an MSD can provide an answer to resolve this difficult problem.

The service modularization literature (see, e.g., Bask et al., 2010; Dörbecker & Böhmann, 2013; Tuunanen & Cassab, 2011; Voss & Hsuan, 2009) has examined how different ITeS design elements (Sheng et al., 2017) or service combinations (Ordanini et al., 2014) impact the design of service experiences. The extant service research defines service experience as phenomenological or process- or outcome-based (Helkkula, 2011), and it typically takes the customer's point of view to study it (Teixeira et al., 2012). The service modularization literature, in turn, looks at the service experience as an objective for the service design. Patrício et al. (2011) have been proponents of developing service design methods that support the design of service experiences. Zomerdjk and Voss (2010) described this outcome-driven approach as experience design. In turn, Jaakkola et al. (2015, p. 190) characterized this as a way to orchestrate service elements to design service experiences. We suggest that service modularization enables the reuse of service elements, such as process steps, which can be combined in service implementation (Bask et al., 2010) to improve service experience outcomes.

Tuunanen and Cassab (2011) aimed to understand how different service modularization choices impact the likelihood of customers using the service again and the impact of these choices on the utility perceived by the customers. However, the use of IT can dramatically increase or decrease the number of encounters between a person and a firm (e.g., Bitner et al., 2000). It has been found that users' perceptions of service encounters are strongly linked to their satisfaction with overall service quality (Parasuraman et al., 1985; Parasuraman et al., 1991). This can apply to ITeS-related services as well (e.g., Jiang et al., 2002). Users form perceptions of service encounters by evaluating the tangible aids in these encounters, such as the IT interfaces. Service users are increasingly creating their own experiences dynamically and autonomously (Ostrom et al., 2015). For instance, in an interactive episode of Netflix's television show *Black Mirror*, viewers can choose different story paths and achieve different service experience outcomes. This highlights the need to examine how ITeS should be designed and how service designers could be assisted in this process.

Although service modularization attracted some interest in the 2010s, its impact on the extant service research literature has been marginal so far. Brax et al. (2017) summarized this situation and proposed research topics to advance the field, including service-specific modularity theories and principles, architectural innovation in services, and modularity in hybrid offerings combining service(s) and tangible product(s). Furthermore, Ordanini et al.'s (2014) call to compare combinations of service attributes with individual service attributes can be echoed for the service modularization literature in general: by understanding how different combinations of service attributes impact the service experience, we can (re)design services to better fulfill the needs and wants of customers, including combinations of the attributes of the architecture, scale, style, shape, and layout of the service as well as functional or aesthetic service features (Sheng et al., 2017). How this can be accomplished remains an open question.

We argue that, by shifting the focus from attempts to modularize different aspects of a service to the DPs of services, we can identify possible ways to generalize the modular design of ITeS and respond to the need for better integration of the service concept and encounter, as well as service delivery system design. For this purpose, we apply Mathiassen and Sørensen's (2008) framework to develop DPs for modular ITeS design. According to Mathiassen and Sørensen (2008), there are four general types of DPs for ITeS. *Computational* DPs rely on encountered services, and they support repeatable patterns of information processing. *Adaptive* DPs interpret and transform available and emergent information by adapting information processing patterns to specific contexts. They rely on relationships and allow involved actors to explore and debate interpretations while executing tasks. *Networking* DPs help actors produce their information about phenomena by following standardized and repeatable patterns of information processing. They connect actors to relevant information sources through IT solutions, such as email systems, search engines, electronic libraries, mobile phones, and SMS messaging. *Collaborative* DPs support actors in producing information about phenomena within an organization and its environment by interpreting the specific work context (Mathiassen & Sørensen, 2008).

McKenna et al. (2013) utilized Mathiassen and Sørensen's (2008) framework to study how adaptive and computational service components are linked to self-efficacy, collaborative service components are linked to social influences, and networking service components are linked to facilitating conditions. A follow-up study by McKenna et al. (2018) demonstrated that different service components have more diverse ties than were found in the earlier study. Although these studies provide valuable insights into the design of ITeS, they do not provide a means for developing an MSD method.

3. RESEARCH METHODOLOGY

This section presents how we resolved the problem of developing an MSD method using a qualitative research approach. The methodological choices depicted here also provide the foundation for the MSD method summarized and presented later in the findings section. We first applied laddering interviews to enable rich qualitative data collection. With laddering interviews researchers are able to collect in-depth data about individuals' perceptions of products or services and their reasoning for these perceptions (Grunert et al., 2001; Reynolds & Gutman, 1988). More specifically, we applied a version of laddering interviewing, which a group of information systems researchers has developed for system design applications in the past two decades (Tuunanen & Peffers, 2018). We then performed a cluster analysis to identify clusters in the collected data within each different service type (Peffers et al., 2003). This procedure allowed us to formulate DPs. The data collection and analysis processes are described in more detail below.

3.1 Data Collection

We collected data from New Zealand companies developing different ITeS for the firms' service offering(s). We adopted Mathiassen and Sørensen's (2008) four general types of

DPs for ITeS as the basis for our theoretical sampling and selection of the participating companies (Patton, 1990) to cover all four different types in our data collection. Four of the six companies were selected based on each of the four general types. This was to ensure that the participating companies were specialized in at least one of the general types.

The six participating companies were all small- to medium-sized and operated within New Zealand. Companies 1 and 3 specialize in innovative, online-enabled, web-based service solutions for individual consumers or small businesses, and both had been operating for less than four years. Company 2 has 12 years of experience in offering adaptive online marketing consultancy services to the marketplace. It specializes in search engine optimization and tactical implementation of online marketing. Company 4 is an experienced franchise retailer of one of the biggest mobile network services in New Zealand. It specializes in selling network services to individual customers and businesses. Company 5 has 32 years of experience developing systems or online service platforms for business users, such as online work management systems or e-commerce solutions. Company 6 is a travel service wholesaler that uses a combination of online services and computer-based software to operate its business. This company is one of the few travel service wholesalers that has been operating for almost 25 years in New Zealand. The data collection was conducted in 2009.

We followed the examples of Reynolds and Gutman (1988), Peffers et al. (2003), and Tuunanen and Peffers (2018) to conduct our laddering interviews. The laddering interview technique is based on the personal construct theory developed by Kelly (1955). Kelly's aim was to model individuals' belief structures based on personal constructs, which result from individuals' observations and interpretations of events (Pervin, 1993), to determine how their perceptions of certain situations impact their experiences in those situations. In other words, he argued that a person has individual multi-dimensional models (i.e., constructs) that describe the attributes and behavior in relation to objects and events, their consequences, and their relationship to a person’s values.

Later, Gutman (1982) proposed that product attributes are relevant to consumers because of the consequences derived from consumption behavior. These consequences are relevant to the personal values they help to satisfy for the consumer. A complete sequence of attribute-consequence-value association is referred to as a means-end chain. To study consumers’ means-end chains for a given product, Reynolds and Gutman (1988) developed an interview approach called “laddering.” In such interviews, participants are typically given a choice or decision task related to a service or product category and then asked to describe which service or product attributes informed their decisions (Veludo-de-Oliveira et al., 2006). Then, participants are questioned to identify the relevant consequences they experienced from using the service or product (Reynolds & Gutman, 1988). Probing questions are asked until the participants describe the final personal values they satisfied by consuming the service or product.

We began the interviews by presenting a list of stimuli (see Appendix A) intended to suggest ideas about possible service applications to enable brainstorming by the participants (Peffers et al., 2003). Following Peffers et al. (2003), we then asked the participants to rank the stimuli in terms of importance. Then, one at a time (for the two highest-ranked stimuli), the interviewer asked each participant to describe the important applications and the desirable attributes (i.e., features) of said applications. The interviewer proceeded to ask the participant to explain why each particular feature was important to elicit the consequences that the participant expected from the feature. The interviewing process continued with a series of “Why?” and “Why would that be important?” questions to determine what end result the subject expected (Reynolds & Gutman, 1988). To elicit more concrete system attributes, we asked the participants a series of questions, such as “What about the system makes you think that it would do that?” The data were recorded as attribute-consequence-value chains, as Tuunanen and Peffers (2018) described. An example of the laddering interview process is included in Figure 1. In this study, we selected six participating companies and conducted 25 individual laddering interviews with the companies’ employees. The demographic information for the 25 interviewees is presented in Table 3.

We recorded all the data we obtained from the individual laddering interviews in a series of laddering chains. From the 25 interviews we conducted, we recorded 556 chains that we used for data analysis. For each chain, the individuals mentioned approximately 8 to 14 consequences. In addition, we recorded at least one attribute for each chain, and when a chain branched out, we recorded several attributes. Regarding values, our experiences were similar to those of other studies (e.g., Peffers et al., 2003), and we recorded values for a total of 227 chains.

[Insert Figure 1. Laddering interview example.]

[Insert Table 3. Demographic profiles of participants.]

3.2 Data Analysis

Data analysis for the study was done in two phases: data coding and cluster analysis. First, we needed to adapt the rich textual data in the laddering chains to what we could use for the cluster analysis. For this purpose, we performed two iterations of data coding. The two iterations are the interpretation process for analyzing our laddering data. This is considered the most critical step for analyzing laddering data as it will directly influence the content quality and the results (Gengler & Reynolds, 1995). To avoid bias, the laddering data were coded by two researchers at the same time. The codes were revised and checked several times by the two researchers. The second phase of the data analysis used cluster analysis to conceptualize the

DPs. For this purpose, we conducted hierarchical cluster analysis using Ward's method (Peffers et al., 2003).

The first iteration of data coding involved assigning descriptive codes for service attributes, performance consequences, and values. Three new columns for attribute, consequence, and value codes were inserted into the worksheet containing the laddering chains. If more than one code could be directed from the laddering chain, the ladders were copied into a separate line to form a sub-chain. The ladders that derived the relevant codes are colored accordingly. After the first iteration, we obtained 344 unique attribute codes, 505 unique consequence codes, and 91 unique value codes, which were summarized based on the original words used by the participants. The second data coding iteration concerned the classification of similar attributes, consequences, and values. We aggregated similar codes into smaller sets based on the three new columns for attributes, consequences, and values. First, we sorted the chains in alphabetical order based on the 505 unique consequence codes. These consequence codes were examined based on similarity and were classified into smaller groups. Then, an identical code was developed for each small group of consequence codes. Thereafter, identical codes were aggregated and cross-checked by the two researchers. The same process was performed for attribute and value codes.

Next, the data needed to be converted from text to binary format to enable the application of the hierarchical cluster analysis using Ward's method, in line with Peffers et al. (2003). Therefore, we converted the unique codes of 221 attributes (A), 96 consequences (C), and 21 values (V) into columns of binary numbers (0 and 1). The service attributes were extracted from the features of ITeS tools mentioned by the participants, and, as such, they are specific to certain services. This resulted in a large aggregation of service attribute codes that emerged from the various ITeS being used or developed by the participants. The columns

Interview ID, Reference Code, Service Type, and Chain Number were selected from the original laddering chains. With the binary representations of A, C, and V codes, the selected columns formed a binary matrix table for use in the cluster analysis. The binary matrix table contains 424 columns and 556 rows of data. The columns, codes, and data types are described in Appendix B. For laddering chains that did not have a relevant A, C, or V code, we created three extra columns: A0, C0, and V0. The binary columns have a value of 0 or 1, where 0 indicates no such code exists in the laddering chain, while 1 indicates the reverse.

Next, we wanted to develop DPs from the data set. For this purpose, the laddering chains were first clustered based on the following variables: Attribute Codes (A), Consequence Codes (C), Value Codes (V), and Service Type (ST). The ST code was derived from the stimuli selections made by the interview participants, which we recorded during the interviews for each ladder chain. After that, we first generated cluster solutions from two to eight clusters for the initial hierarchical analysis within each ITeS type (i.e., subsets of the data). For the clustering analysis, the laddering chains formed the unit of analysis. To measure distance, we squared the Euclidean distance to measure similarities between laddering chains, as our data were in binary metric format for a hierarchical clustering procedure (Hair et al., 2006). In hierarchical clustering, clusters are nested rather than mutually exclusive as they are formed only by joining existing clusters. Any member of a cluster can trace its membership in an unbroken path to its beginning as a single observation. Finally, we adopted Ward's method as our clustering algorithm.[3]

Finally, we used correlation tests between each consequence and value code[4] within each sub-cluster to determine aggregate connections between the constructs using the Kendall

[3] Ward's minimum-distance hierarchical method calculates the sum of squared Euclidean distances from each case in a cluster to the mean of all the variables.

[4] Online Appendix 2 reports the correlated consequence and value constructs, and the key notes derived from these, at the .05 level, and an example of a correction test for DP10.

tau rank correlation method (Abdi, 2007), which measures the strength of association between the consequences values within each sub-cluster. The correlation test helps elucidate the dependence relationship between the design elements and what impacts them.[4] If we did not find a connection between the consequence code and a value code, we added a not applicable (N/A) note to the table. We began to interpret the codes by developing an analytical tool, keynotes, based on the wording used in the consequence and value codes. These keynotes were then used to develop the DPs. Furthermore, the content within the chains was used to interpret each DP. The consequences reflect the design elements valued by individuals based on their experience with each type of ITeS.

4. FINDINGS: PRIORITIZED DPs

This section depicts how the applied research methods enabled us to define DPs and prioritize them. Figure 2 presents a summary of how specific DPs fall into each of the four general types. We use the number of laddering chains to indicate priority between the general DP types and order the DPs within each type. In Figure 2, the most important DPs are on the left both within the four general types and within each DP type. Table 4 describes all the DP definitions and the descriptive statistics for each DP.[5] Below, we also present four exemplars of the dataset based on the computational, adaptive, collaborative, and network DPs. We selected two of the highest-ranking examples for adaptive and computational DPs and two of the second highest-ranking collaborative and network DPs to give some variety.

[Insert Figure 2. Prioritized DPs.]

[Insert Table 4. Summarized DPs.]

[5] Online Appendix 1 presents all prioritized DP descriptions.

4.1 Computational: Flexible Information Integration

For this DP, the key issues are integration, usability, flexibility, performance, and responsiveness of the service delivery system. This requires standardized information obtained from different units within or across organizations. According to the participants, successful integration is the backbone of the other key aspects because it organizes information from several sources and thus reduces, for example, the risk of inputting the same information several times. It is also recommended to deliver information through one integrated system and update information in real-time within one system or in sync within several systems. Based on these results, we propose the following DP:

DP9: *Computational designs should focus on flexible information integration with other services.*

4.2 Adaptive: Usability and Customer-Centricity

For this DP, three important issues emerged from the data: functionality, customer-centricity, and usability. The participants emphasized the role of practical functions in adapting to tasks performed by knowledge workers, such as merging information from different databases. The participants also highlighted usability since they required easy-to-use functions, hassle-free recovery options, and timely responses in their service encounters. Positively perceived usability led participants to think that they could rely on the ITeS they used. Based on these results, we propose the following DP:

DP1: *Adaptive designs should focus on the usability and customer-centricity of the functionalities.*

4.3 Collaborative: Accessibility and Knowledge Sharing

For this DP, we observed two service concept issues: accessibility and sharing. The participants indicated that they seek easily accessible ITeS to support distributed work and collaboration. However, end-users often experience transaction overload when interacting with an ITeS. The participants also believed that if an ITeS promotes open knowledge sharing across users (e.g., in organizations), it can create a culture where everyone is encouraged and potentially willing to contribute to a common good. Based on these results, we propose the following DP:

DP5: *Collaborative designs should focus on accessibility and information sharing to support collaboration.*

4.4 Network: Constantly Improving Service Efficiency

For this DP, efficiency, mainly cost and resource savings, were the most important aspects. Participants also highlighted that they aim to minimize technical support by constantly improving the quality of services. For example, one participant stated that the goal was to eliminate customer support and, in so doing, decrease costs and increase the efficiency of the service delivery system. Based on these results, we propose the following DP:

DP13: *Network designs should focus on constant improvement of service efficiency.*

5. GENERAL DISCUSSION AND IMPLICATIONS FOR RESEARCH AND PRACTICE

Our study contributes to the literature by showing how complex service systems, specifically ITeS, can be modularly designed (Maglio et al., 2009). Our DPs show how different ITeS design elements (Sheng et al., 2017) or service attribute combinations (Ordanini et al., 2014) impact the outcome-driven design of service experiences (Zomerdijk & Voss, 2010). Our study also responds to the recent call for the advancement of service modularization research (Brax et al., 2017). We offer a new approach to designing services (specifically ITeS), applying DPs as the conceptual framework foundation for service modularization. The implications of our research for future research are summarized in Table 5 and discussed below.

[Insert Table 5. Implications for research and practice and topics for future research.]

Tuunanen and Cassab (2011) argued that it is necessary to understand how ITeS should be designed and proposed service process modularization to achieve this. While some modularization research has been published in recent years (see, e.g., Brax et al., 2017), the overall impact of this research stream on the service research community has remained modest, and it has not yet significantly impacted service design research or practice. Recent papers published by the *Journal of Service Research* (Patrício et al., 2011; Teixeira et al., 2017) still focus on combinational solutions that apply sets of well-known service design methods, such as blueprinting (Bitner et al., 2008; Shostack, 1984) or value constellation mapping (Michel et al., 2008), to address the complexity of designing ITeS. While they offer valuable insights, these studies do not adequately explain how to design combinations of service attributes.

Therefore, we argue that we should move beyond traditional ways of conceptualizing service design methods to facilitate the design of service attribute/feature combinations. The application of DPs provides a meaningful foundation for this. However, this solution requires re-thinking the expected outputs from service design activities and how DP-driven modular service design should be performed. Earlier, Yu and Sangiorgi (2014, 2018) proposed an approach to compare service design methods (cf. Table 1) based on activities (design and analysis/development and implementation), objects of the activities (service concept/service delivery system), and facilitators for these (methods and tools/staff and customer involvement/organizational dimensions). However, the extant literature does not yet offer ways to integrate DPs into service design methods.

We argue that DPs can be used to modularize service modules (Brax et al., 2010), and that the four general DPs for ITeS provide a theoretically sound way to accomplish this. We further suggest that these general types can be further divided. Namely, for the service concept object, we propose that adaptive and collaborative DPs should be emphasized. Correspondingly, for the service delivery system object, computational and network DPs should be considered first. Our findings support this argument (cf. Table 4 and Online Appendix 1), which is summarized in Table 6. Still, we also recognize that this issue is likely not straightforward. Moreover, the application of the DPs is contextual, and the emphasis of DPs for each IT-enabled service will vary. Thus, the presented DPs should not be considered general guidance for all ITeS designs but instead instantiation for our MSD method's applicability to support modular service design for ITeS.

[Insert Table 6. The MSD framework, based on Bask et al. (2010), Mathiassen and Sørensen (2008), and Yu and Sangiorgi (2014, 2018).]

While our conceptualization of the MSD methods (Table 6) offers new possibilities for theorizing about service solutions (Markus et al., 2002) and developing generalizable DPs for ITeS, its more immediate impact is on the further development of service design methods that accommodate the discovery and recognition of DPs. These methods should also enable modularization at the service attribute combination (i.e., feature) level, thus resolving the complex problem of operationalizing service modularization in service design and, more importantly, how to show its benefits in terms of efficacy and efficiency. So far, these have remained unsolved by the service modularization research community (Brax et al., 2017).

However, our study provides the foundation for this work. As a first step forward, we argue that the definition of a modular service offering (Bask et al., 2014) should be reformulated. We suggest that modular service offerings should be defined as *standardized sets of base service features and DPs, sets of customized service features and DPs, or combinations thereof*. Our study offers an example of how DPs can be formed based on rich laddering interview data. Tuunanen and Peffers (2018) have demonstrated how attribute-level information for service features can be derived from similar datasets. We foresee that the proposed definition of modular service offerings can revitalize the current stagnant state of modularization research. It opens new ways, for example, to study architectural innovation in services and how modularity in hybrid offerings combining service(s) and tangible product(s) can be accomplished (cf. Brax et al., 2017).

5.1. Implications for Practice—Presenting an MSD Method

The findings provide general guidance for applying the MSD framework in practice, specifically for the design of ITeS. Table 7 depicts how the MSD framework can be operationalized and how service modularization can be used effectively for service design, especially for ITeS design. While the literature has argued for the benefits of service modularization (see, e.g., Bask et al., 2017), the industry has been slow to adopt the concept in practice despite the argued benefits of efficiency gains for the design shown in product and software development (Tuunanen & Cassab, 2011). Our argument here is that the problem may be in the conceptualization of service modularization by academic researchers.

The presented MSD method is ready for use by service designers. It can be extended further, for example, to develop the analysis approaches to target specific service modules in an IT-enabled service. The firms can use the MSD method to collect data and derive DPs that offer a concise overall view of the most central design aspects for their ITeS. Such a comprehensive view can be beneficial for service designers and managers and decision-makers, e.g., by orchestrating the integration of service elements into the design of service experiences (Jaakkola et al., 2015). Our advice for applying this method for service modularization, and the resulting DPs, to a particular study, is first to look at the general composition of an ITeS and then use specific DPs to enhance its design. Of course, this should be done with the help of various methods and techniques to identify the user needs that will inform the features of the ITeS. Our findings provide a starting point for understanding the underlying principles of different types of ITeS and how they should be designed. In this way, our DPs start to bridge the potential "language gap" between the more tech-savvy individuals and others.

We also offer a more straightforward definition that focuses on service features and more generalizable DPs. Furthermore, we suggest that practitioners should initially recognize the standard features instead of using several modularization types (like reuse, substitution, or variation). These features are customized and, lastly, combinations of these modularization types. In our view, the more important matter is to understand how to design new services versus discussing the finer details of philosophical differences between, for example, reused and variated service features. Thus, we simply propose that DPs should be the basis for service modularization efforts. We suggest that practitioners use DPs, for example, to recognize standard, customized, or combined service features. These can be later considered for reuse, variation, or substitution to improve the MSD efficiency further.

Our MSD method also provides rich information about the actual or perceived use of different types of ITeS. These details can be utilized in tandem with the service modularization approach; developers and providers can look at each service module and evaluate how well the module supports the DPs relevant to the IT-enabled service type in question. In this way, managers can more accurately identify developmental areas and weaknesses in their service concept and delivery systems at a feature or feature set level (cf. Table 7) to improve the customers' service experience. In other words, by recognizing how different service features impact the service experience, we can design better services (Ordanini et al., 2014). Furthermore, our MSD method can provide service designers with ways to achieve a fine-grained view of each service module. By analyzing the laddering data and the attribute-consequence linkages, service designers and managers can gain detailed information about how each attribute related to a particular module matches (or does not match) the desired outcome of the module. In addition, Table 7 also illustrates how the MSD method facilitates different methods and tools to support staff and customer involvement. Furthermore, we argue that the method also enables taking account of organizational needs and customer requirements for the developed ITeS via prioritizing the recognized DPs.

[Insert Table 7. Operationalizing the MSD framework to depict our method.]

Our MSD method can also be modularized itself, and service designers can substitute or variate parts of the method to fit their organizations or projects. Our approach builds on the work done in applying personal construct theory to understand customers' mental models (Kelly, 1955). However, other suitable theories may similarly help guide the development of DPs and MSD methods. We ask the service designers to consider that they pause to think why specific service design methods work and the (theoretical) reasoning for this. This may lead to the design of better service design methods. Our method provides an exemplar of this by purposefully developing a method to support MSD for ITeS. This is a different approach to developing service methods than we usually see in the literature. For example, blueprinting (Bitner et al., 2008) or the more recent MINDS (Teixeira et al., 2019) do not make an argument as to why these methods work. Moreover, these are combinations of earlier work that have been customized to offer a solution to a set of service design problems.

We recognize that these well-known service methods, such as blueprinting, clearly have value as the industry has widely adopted them. Still, what could we achieve if we purposefully, using the literature and research methods, developed theory-ingrained methods for service design, and how would this change service design practice? Kurt Lewin argued in the 1940s that "there is nothing more practical than a good theory" (Hunt, 1987). We believe that this is also true for service design methods and the practice of service design. The answer is not likely to be just "efficiency gains for service design." Still, we would also learn, for example, why certain service design methods are easier to adopt by service designers and managers and whether the applied service design methods impact the service itself.

6. LIMITATIONS

Our study develops an MSD method that particularly fits the design of ITeS. Furthermore, the developed method applies DPs to conceptualize how modularization can be accomplished in service design in general. We applied a qualitative research approach for development of the method. The data consist of 25 individual interviews with representatives of six organizations in New Zealand. We recognize that our participant sample is limited since it was obtained from a single country. However, New Zealand's multicultural society brings potential richness to the dataset. Another limitation is that the organizations chosen for the case study were selected from the current pool of companies in the Auckland region using theoretical sampling. While the number of interviews (25) conducted can be considered low, it is within the range of the number of interviews conducted in similar studies (Peffers et al., 2003; Peffers & Tuunanen, 2005; Tuunanen & Govindji, 2016; Tuunanen & Kuo, 2015; Tuunanen & Peffers, 2018). Thus, we see that our study meets the expectations of similar studies in the literature, and it opens many new avenues for further research. For example, we should study how the context of specific ITeS impacts the application of the MSD method or how to define new, generalizable DPs that can be applied outside of specific ITeS contexts.

References

Abdi, H. (2007). Kendall rank correlation. In N. J. Salkind (Ed.), *Encyclopedia of measurement and statistics* (pp. 508-510). Sage.

Bask, A., Merisalo-Rantanen, H., & Tuunanen, T. (2014). Developing a modular service architecture for e-store supply chains: The small-and-medium-sized enterprise perspective. *Service Science*, *6*(4), 251–273.

Berry, L. L., & Lampo, S. K. (2000). Teaching an old service new tricks: The promise of service redesign. *Journal of Service Research*, *2*(3), 265–275.

Bitner, M. J., Brown, S. W., & Meuter, M. L. (2000). Technology infusion in service encounters. *Journal of the Academy of Marketing Science*, *28*(1), 138–149.

Bitner, M. J., Ostrom, A. L, & Morgan, F. N. (2008). Service blueprinting: A practical technique for service innovation. *California Management Review*, *50*(3), 66–94.

Brax, S. A., Bask, A., Hsuan, J., & Voss, C. (2017). Service modularity and architecture—An overview and research agenda. *International Journal of Operations & Production Management*, *37*(6), 686–702.

Cenfetelli, R. T., Benbasat, I., & Al-Natour, S. (2008). Addressing the what and how of online services: Positioning supporting-services functionality and service quality for business-to-consumer success. *Information Systems Research*, *19*(2), 161–181.

Gengler, C. E., & Reynolds, T. J. (1995). Consumer understanding and advertising strategy: Analysis and strategic translation of laddering data. *Journal of Advertising Research*, *35*(4), 19–33.

Goldstein, S. M., Johnston, R., Duffy, J., & Rao, J. (2002). The service concept: The missing link in service design research? *Journal of Operations Management*, *20*(2), 121–134.

Gregor, S. (2002). Design theory in information systems. *Australasian Journal of Information Systems*, *10*(1), 14–22.

Gregor, S., & Hevner, A. R. (2013). Positioning and presenting design science research for maximum impact. *Management Information Systems Quarterly*, *37*(2), 337–355.

Gregor, S., & Jones, D. (2007). The anatomy of a design theory. *Journal of the Association for Information Systems*, *8*(5), 312–335.

Grönroos, C. (2006). Adopting a service logic for marketing. *Marketing Theory*, *6*(3), 317–333.

Grönroos, C. (2007). *In search of a new logic for marketing: Foundations of contemporary theory*. Wiley & Sons, Ltd.

Grönroos, C. (2008). Service logic revisited: Who creates value? And who co-creates? *European Business Review*, *20*(4), 298–314.

Grönroos, C., & Voima, P. (2013). Critical service logic: Making sense of value creation and co-creation. *Journal of the Academy of Marketing Science*, *41*(2), 133–150.

Grunert, K. G., Beckmann, S. C., & Sorensen, E. (2001). Means-end chains and laddering: An inventory of problems and an agenda for research. In T. J. Reynolds & J. C. Olson (Eds.), *Understanding consumer decision making: The means-end approach to marketing and advertising strategy* (pp. 63–90). Lawrence Erlbaum Associates.

Gutman, J. (1982). A means-end chain model based on consumer categorization processes. *The Journal of Marketing*, *46*(2), 60–72.

Hair, J., Black, W., Babin, B., Anderson, R., & Tatham, R. (2006). Cluster analysis. In *Multivariate data analysis* (6th ed.). Pearson Prentice Hall.

Helkkula, A. (2011). Characterizing the concept of service experience. *Journal of Service Management*, *22*(3), 367–389.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *Management Information Systems Quarterly*, *28*(1), 75–105.

Huang, M.-H., & Rust, R. T. (2018). Artificial intelligence in service. *Journal of Service Research*, *21*(2), 155–172.

Hunt, D. E. (1987). *Beginning with ourselves: In practice, theory and human affairs*. Brookline Books.

Iversen, J. H., Mathiassen, L., & Nielsen, P. A. (2004). Managing risk in software process improvement: An action research approach. *Management Information Systems Quarterly*, *28*(3), 395–433.

Jaakkola, E., Helkkula, A., & Aarikka-Stenroos, L. Service experience co-creation: Conceptualization, implications, and future research directions. *Journal of Service Management*, *26*(2), 182-205.

Jia, R., Reich, B. H., & Pearson, J. M. (2008). IT service climate: An extension to IT service quality research. *Journal of the Association for Information Systems*, *9*(5), 294-320.

Jiang, J. J., Klein, G., & Carr, C. L. (2002). Measuring information system service quality: SERVQUAL from the other side. *MIS Quarterly*, *26*(2), 145–166.

Jose, A., & Tollenaere, M. (2005). Modular and platform methods for product family design: Literature analysis. *Journal of Intelligent Manufacturing*, *16*(3), 371-390.

Karmarkar, U. (2004). Will you survive the services revolution? *Harvard Business Review*, *82*(6), 100–107.

Karwan, K. R., & Markland, R. E. (2006). Integrating service design principles and information technology to improve delivery and productivity in public sector operations: The case of the South Carolina DMV. *Journal of Operations Management*, *24*(4), 347–362.

Kelly, G. A. (1955). *The psychology of personal constructs.* W. W. Norton & Company.

Ladhari, R. (2009). A review of twenty years of SERVQUAL research. *International Journal of Quality and Service Sciences*, *1*(2), 172–198.

van Liere, D. W., Hagdorn, L., Hoogeweegen, M. R., & Vervest, P. H. (2004). Embedded coordination in a business network. *Journal of Information Technology*, *19*(4), 261-269.

Lusch, R. F., & Nambisan, S. (2015). Service innovation: A service-dominant logic perspective. *Management Information Systems Quarterly*, *39*(1), 155–175.

Mager, B. (2008). Service design. In M. Erlhoff, T. Marshall & L. Bruce (Eds.), *Design dictionary: Perspectives on design terminology* (pp. 354-357). Birkhäuser Verlag.

Maglio, P., Vargo, S. L., Caswell, N., & Spohrer, J. (2009). The service system is the basic abstraction of service science. *Information Systems and E-business Management*, *7*(4), 395–406.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *Management Information Systems Quarterly*, *26*(3), 179–212.

Mathiassen, L. L. M., & Sørensen, C. C. (2008). Towards a theory of organizational information services. *Journal of Information Technology*, *23*(4), 313–329.

Mathiassen, L., Saarinen, T., Tuunanen, T., & Rossi, M. (2007). A contingency model for requirements development. *Journal of the Association for Information Systems*, *8*(11), 569.

McKenna, B., Cai, W., & Tuunanen, T. (2018). Technology enabled information services use in tourism: An ethnographic study of Chinese backpackers. *Pacific Asia Journal of the Association for Information Systems*, *10*(4), 37–64.

McKenna, B., Tuunanen, T., & Gardner, L. (2013). Consumers' adoption of information services. *Information & Management*, *50*(5), 248–257.

Meuter, M. L., Ostrom, A. L., Roundtree, R. I., & Bitner, M. J. (2000). Self-service technologies: Understanding customer satisfaction with technology-based service encounters. *Journal of Marketing*, *64*(3), 50–64.

Michel, S., Brown, S. W., & Gallan, A. S. (2008). Service-logic innovations: How to innovate customers, not products. *California Management Review*, *50*(3), 49–65.

Modesto Veludo-de-Oliveira, T., Akemi Ikeda, A., & Cortez Campomar, M. (2006). Laddering in the practice of marketing research: Barriers and solutions. *Qualitative Market Research: An International Journal*, *9*(3), 297–306.

Möller, K., Rajala, R., & Westerlund, M. (2008). Service innovation myopia? A new recipe for client-provider value creation. *California Management Review*, *50*(3), 31–48.

Ordanini, A., Parasuraman, A., & Rubera, G. (2014). When the recipe is more important than the ingredients: A qualitative comparative analysis (QCA) of service innovation configurations. *Journal of Service Research*, *17*(2), 134–149.

Ostrom, A. L., Parasuraman, A., Bowen, D. E., Patrício, L., & Voss, C. A. (2015). Service research priorities in a rapidly changing context. *Journal of Service Research*, *18*(2), 127–159.

Parasuraman, A., Zeithaml, V. A., & Berry, L. L. (1985). A conceptual model of service quality and its implications for future research. *The Journal of Marketing*, *49*(4), 41–50.

Parasuraman, A., Berry, L., & Zeithaml, V. (1991). Refinement and reassessment of the SERVQUAL scale. *Journal of Retailing*, *67*(4), 114–139.

Patrício, L., Fisk, R. P., Falcão e Cunha, J., & Constantine, L. (2011). Multilevel service design: From customer value constellation to service experience blueprinting. *Journal of Service Research*, *14*(2), 180–200.

Patton, M. Q. (1990). *Qualitative evaluation and research methods*. Sage Publications.

Peffers, K., Gengler, C. E., & Tuunanen, T. (2003). Extending critical success factors methodology to facilitate broadly participative information systems planning. *Journal of Management Information Systems*, *20*(1), 51–85.

Peffers, K., & Tuunanen, T. (2005). Planning for IS applications: A practical, information theoretical method and case study in mobile financial services. *Information & Management*, *42*(3), 483–501.

Pervin, L. A. (1993). *Personality theory and research.* John Wiley & Sons.

Peters, C., Maglio, P., Badinelli, R., Harmon, R. R., Maull, R., Spohrer, J. C., Tuunanen, T., Vargo, S. L., Welser, J. J., Demirkan, H., & Griffith, T. L. (2016). Emerging digital frontiers for service innovation. *Communications of the Association for Information Systems*, *39*(1), 136–149.

Pfeffer, J., & Sutton, R. I. (2006). Evidence-based management. *Harvard Business Review*, *84*(1), 62–74.

Prahalad, C. K., & Ramaswamy, V. (2004). Co-creation experiences: The next practice in value creation. *Journal of Interactive Marketing*, *18*(3), 5–14.

Reynolds, T. J., & Gutman, J. (1988). Laddering theory, method, analysis, and interpretation. *Journal of Advertising Research*, *28*(1), 11–31.

Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *Management Information Systems Quarterly*, *35*(1), 37–56.

Sheng, X., Simpson, P. M., & Siguaw, J. A. (2017). Communities as nested servicescapes. *Journal of Service Research*, *20*(2), 171–187.

Shostack, G. L. (1977). Breaking free from product marketing. *Journal of Marketing*, *41*(April), 73–80.

Shostack, G. L. (1984), Designing services that deliver. *Harvard Business Review*, *62*(1), 133–139.

Silvestro, R., Fitzgerald, L., & Johnston, R. (1992). Towards a classification of service process. *International Journal of Service Industry Management*, *3*(3), 62–75.

Teixeira, J., Patrıcio, L., Nunes, N. J., Nóbrega, L., Fisk, R. P., & Constantine, L. (2012). Customer experience modeling: from customer experience to service design. *Journal of Service Management*, *23*(3), 362-376.

Teixeira, J., Patrício, L., Huang, K. H., Fisk, R. P., Nóbrega, L., & Constantine, L. (2017). The MINDS method: Integrating management and interaction design perspectives for service design. *Journal of Service Research*, *20*(3), 240–258.

Teixeira, J. G., Patrício, L., & Tuunanen, T. (2019). Advancing service design research with design science research. *Journal of Service Management*, *30*(5), 577–592.

Tuunanen, T., & Cassab, H. (2011). Service process modularization: Reuse versus variation in service extensions. *Journal of Service Research*, *14*(3), 340–354.

Tuunanen, T., & Govindji, H. (2016). Understanding flow experience from users' requirements. *Behaviour & Information Technology*, *35*(2), 134–150.

Tuunanen, T., Kazan, E., Salo, M., Leskelä, R. L., & Gupta, S. (2019). From digitalization to cybernization. *Scandinavian Journal of Information Systems*, *31*(2), 83–96.

Tuunanen, T., & Kuo, I. T. (2015). The effect of culture on requirements: A value-based view of prioritization. *European Journal of Information Systems*, *24*(3), 295–313.

Tuunanen, T., Myers, M., & Cassab, H. (2010). A conceptual framework for consumer information systems development. *Pacific Asia Journal of the Association for Information Systems*, *2*(1), 47–66.

Tuunanen, T., & Peffers, K. (2018). Population targeted requirements acquisition. *European Journal of Information Systems*, *27*(6), 686–711.

Vargo, S. L., & Lusch, R. F. (2004). Evolving to a new dominant logic for marketing. *Journal of Marketing*, *68*(1), 1–17.

Vargo, S. L., & Lusch, R. F. (2008). Service-dominant logic: Continuing the evolution. *Journal of the Academy of Marketing Science*, *36*(1), 1–10.

Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, *3*(1), 36–59.

Wirtz, J., Patterson, P. G., Kunz, W. H., Gruber, T., Lu, V. N., Paluch, S., & Martins, A. (2018). Brave new world: Service robots in the frontline. *Journal of Service Management*, *29*(5), 907–931.

Yu, E., & Sangiorgi, D. (2018). Service design as an approach to implement the value co-creation perspective in new service development. *Journal of Service Research*, *21*(1), 40–58.

Yu, E., & Sangiorgi, D. (2014). Service design as an approach to new service development: Reflections and future studies. In ServDes 2014 (pp. 194–204). Linköping Electronic Conference Proceedings.

Zeithaml, V. A., Berry, L. L., & Parasuraman, A. (1993). The nature and determinants of customer expectations of service. *Journal of the Academy of Marketing Science*, *21*(1), 1–12.

Zomerdijk, L. G., & Voss, C. A. (2010). Service design for experience-centric services. *Journal of Service Research*, *13*(1), 67–82.

Appendix A. Stimuli List

At the start of each interview session, the interviewee was given a brief description of the four types of ITeS in order to stimulate the discussion. The descriptions of each type of ITeS were as follows.

Network services: Services that deal with instant communication. Such services aim to connect users to relevant information sources through software or hardware. Examples include email systems, broadband services, mobile phones, and SMS messaging. The technologies available for these types of services will provide users with immediate access to relevant information sources.

Collaborative services: Services that rely on relationships. Information is shared on a collaboration platform for distributed work. Examples of collaboration systems include Microsoft SharePoint, Google Docs, and Microsoft Exchange Server.

Adaptive services: Services that create flexible business processes or enhance product customization. Examples of an adaptive service evoked by a customer include the standard task of ordering groceries from a website, paying for the items with a credit card, and choosing a delivery date and time. Another example of an adaptive service is search engine optimization (SEO) for websites, which is the process of improving a website's ranking.

Computational services: Transactional type of information services. These deal with processing operational data, such as transactions, accounts, or customers. An account management system is an example of this type of service.

To stimulate ideas, the participants were asked to talk about and list the types of ITeS with which they have interacted. Then, they were asked to describe a recent problem they encountered with this type of service. Guiding questions for the laddering interviews were as follows.

Guiding Questions:

What kinds of ITeS you are using/developing right now?

Please describe your experience using ITeS.

What is the first thing that comes to your mind when you see the term “IT-enabled services”?

What is the main challenge you have encountered when using these services?

How does the performance of the service affect your work?

How complicated do you think the types of ITeS that you are using are?

Appendix B. Binary Matrix Table of Laddering Data

Column Number	**Column Name**	**Code**	**Data Type**
Column 1	Interview ID	1, 2, 3, … 25	Numeric
Column 2	Reference Code (Company)	Company 1, Company 2, Company 3, … Company 6	String
Column 3	Service Type	1, 2, 3, 4	Numeric
Column 4	Chain Number	Main chain = Whole number starting with 10, 20, 30, etc. Sub-chain = Sequential numbers directly following the main chain number (e.g., sub-chains for Chain 10 are 11, 12, 13, etc.)	String
Columns 5, 6, 7, 8	Service Type 1 (ST1) Service Type 2 (ST2) Service Type 3 (ST3) Service Type 4 (ST4)	Binary: 0 = No 1 = Yes	Numeric
Columns 9–300	Attribute Codes A0, A1, … A221	Binary: 0 = No 1 = Yes	Numeric

Columns 301–397	Consequence Codes C0, C1, C2, … C97	Binary: 0 = No 1 = Yes	Numeric
Columns 398–424	Value Codes V0, V1, V2, … V26	Binary: 0 = No 1 = Yes	Numeric

Figures

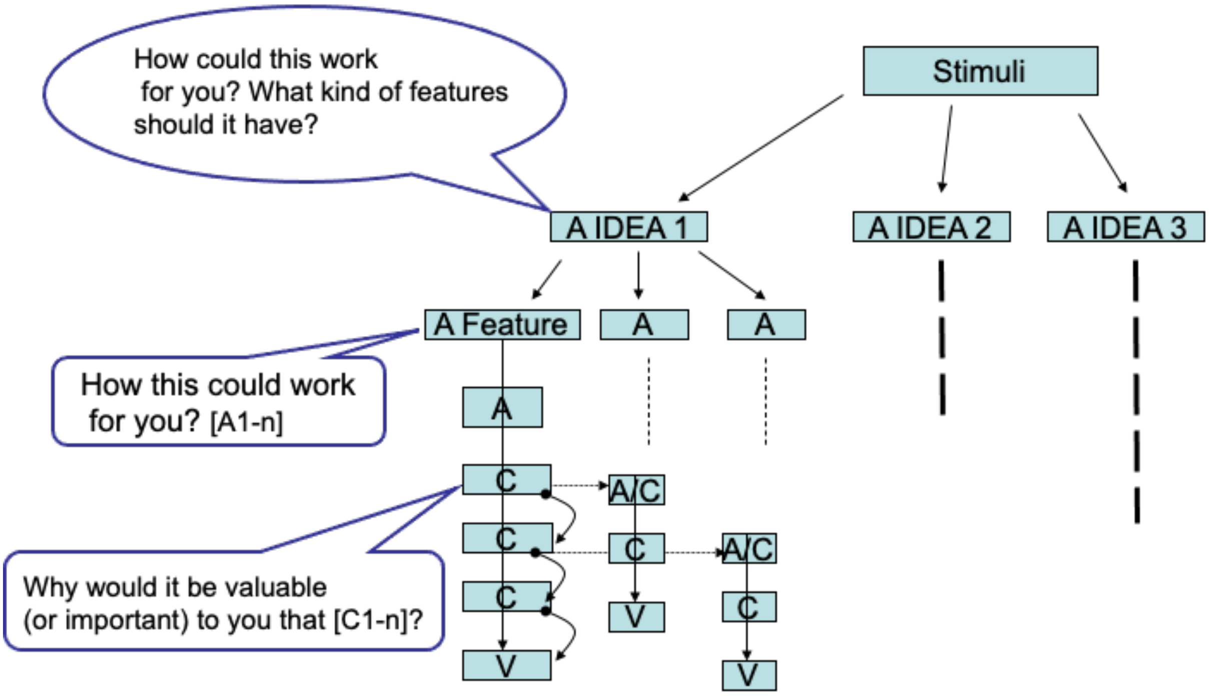


Figure 1. Laddering interview example.

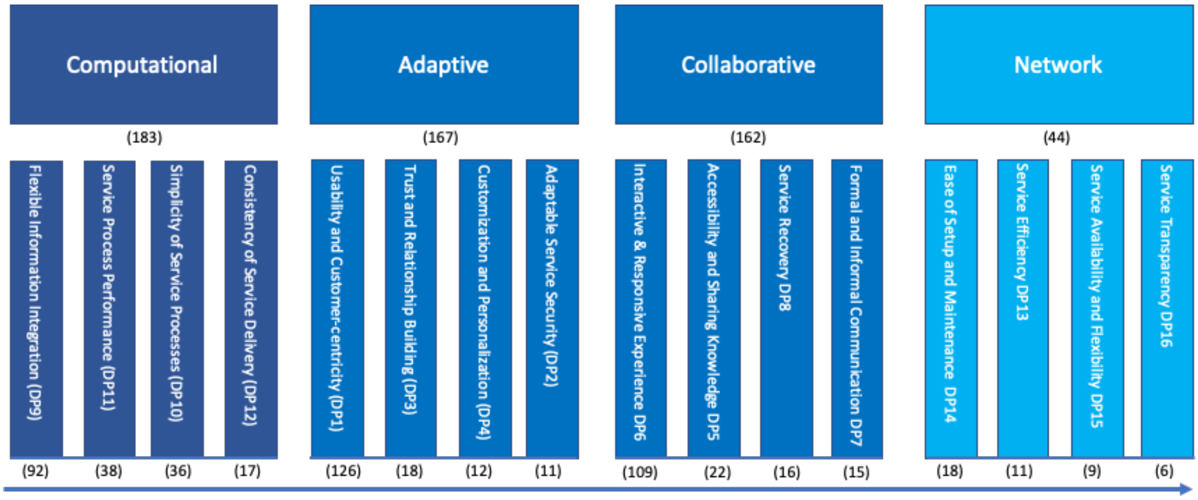

Computational
(183)
Flexible Information Integration (DP9)
(92)
Service Process Performance (DP11)
(38)
Simplicity of Service Processes (DP 10)
(36)
Consistency of Service Delivery (DP 12)
(17)
Adaptive
(167)
Usability and Customer-centricity (DP1)
(126)
Trust and Relationship Building (DP3)
(18)
Customization and Personalization (DP4)
(12)
Adaptable Service Security (DP2)
(11)
Collaborative
(162)
Interactive & Responsive Experience DP6
(109)
Accessibility and Sharing Knowledge DP5
(22)
Service Recovery DP8
(16)
Formal and Informal Communication DP7
(15)
Network
(44)
Ease of Setup and Maintenance DP14
(18)
Service Efficiency DP13
(11)
Service Availability and Flexibility DP15
(9)
Service Transparency DP16
(6)

Figure 2. Prioritized DPs.

Tables

Table 1. The service design framework, adapted from Yu and Sangiorgi (2014, 2018).

Focus	Design and analysis	Development and implementation
Objects	Service concept (value, form and function, experience, outcomes)	Service delivery system (structure, infrastructure, process)
Facilitators	Methods and tools, staff and customer involvement, organizational dimensions	

Table 2. The illustration of the MINDS service design method.

Focus	Service concept and encounter design	Service system navigation design
Objects	Form and functions, experience	Navigation process and structure
Facilitators	Use of semi-structured interviews with customers and other actors to develop: Affinity diagram—value constellation modeling User interface/wireframe sketching—blueprinting for service experience modeling Storyboarding (scenarios)—blueprinting for service system modeling	

Table 3. Demographic Profiles of Participants.

Age Distribution (Years)	**Number of Participants**	**Years of Experience (Years)**	**Number of Participants**	**Gender**	**Number of Participants**
20–29	7	1–3	9	Male	14
30–39	11	4–6	7	Female	11
40–49	5	7–9	3		
50–59	2	10+	6		

Table 4. Summarized DPs.

Type	ID	DP definition	No. of Chains	No. of Unique Consequences
Computational Design Principles	DP9	Designs should focus on flexible information integration with other services	92	43
	DP11	Designs should focus on service process performance	38	15
	DP10	Designs should focus on simple operational processes and activities	36	8
	DP12	Designs should focus on quality and consistency in service delivery	17	13
Adaptive Design Principles	DP1	Designs should focus on usability and customer-centricity of the functionalities	126	58
	DP3	Designs should focus on enabling trust and relationship building	18	10

	DP4	Designs should focus on customization and personalization of the service	12	4
	DP2	Designs should focus on offering adaptable service security	11	4
Collaborative Design Principles	DP6	Designs should focus on interactive and responsive service experiences that facilitate the development of customer relationships	109	50
	DP5	Designs should focus on accessibility and information sharing to support collaboration	22	13
	DP8	Designs should focus on robust service recovery options	16	10
	DP7	Designs should enable both formal and informal communication anytime and anywhere	15	6
	DP14	Designs should focus on offering functionalities that are easy to set up and maintain	18	15

Network Design Principles	DP13	Designs should focus on constant improvement of service efficiency	11	4
	DP15	Designs should focus on service availability and flexibility	9	5
	DP16	Designs should have a transparent service logic	6	4

Table 5. Implications for research and practice and topics for future research.

Implications for Research	**Implications for Practice**
Modularization of services should be considered at the DP level to facilitate the design of service attribute/feature combinations. The general DPs provide a theoretically sound way to reuse, substitute, or variate service modules.	The MSD method is ready for use by service designers and managers.
The MSD framework shows how DPs can be integrated into service design methods.	Service designers and managers should initially recognize the standard features, how these can be customized, and how combinations of these can be used for service modularization.
We propose that modular service offerings be redefined as standard, customized service features and DPs, or combinations thereof.	The MSD method can be modularized, and service designers can substitute or variate parts of the method.

Table 6. The MSD framework, based on Bask et al. (2010), Mathiassen and Sørensen (2008), and Yu and Sangiorgi (2014, 2018).

Focus	Design and Analysis	Development and Implementation
Objects	Service concept (potential value, form and function, experience, outcomes)	Service delivery system (structure, infrastructure, process)
Design Principles	Adaptive and collaborative service modules (standard, customized, or combinations thereof)	Computational and network service modules (standard, customized, or combinations thereof)
Facilitators	Methods and tools, staff and customer involvement, organizational dimensions	

Table 7. Operationalizing the MSD framework to depict our method.

Focus	Design and Analysis	Development and Implementation
Objects	Use interview data to describe the service concept, focusing on: • The potential value for customers and staff • Form and function of the ITeS • ITeS experience and outcomes	Use interview data to describe the service delivery system, focusing on: • ITeS structure • ITeS infrastructure • ITeS process
Design Principles	Develop prioritized adaptive and/or collaborative design principles that can be used to define ITeS modules (standard, customized, or combinations thereof)	Develop prioritized computational and/or network design principles that can be used to define ITeS modules (standard, customized, or combinations thereof)
Facilitators	Apply methods and tools for staff and customer involvement and accounting for your organizational needs and customer requirements: • Laddering interviewing for rich customer data collection • Data coding for developing user need definitions and data constructs • Cluster analysis for developing DPs and aggregated design knowledge • Prioritizing DPs to meet your organizational needs and customer requirements	

Online Appendix 1. Prioritized Design Principles (DPs)

Computational DPs

Flexible information integration

For this DP, the key issues are integration, usability, flexibility, performance, and responsiveness of the service delivery system. This requires standardized information obtained from different units within or across organizations. According to the participants, successful integration is the backbone of the other key aspects because it organizes information from several sources and thus reduces, for example, the risk of inputting the same information several times. It is also recommended to deliver information through one integrated system and update information in real-time within one system or in sync within several systems. Based on these results, we propose the following DP:

DP9: Computational designs should focus on flexible information integration with other services.

Service process performance

For this DP, the participants highlighted service performance issues, such as redundancy, and timely responses as important for the design. The participants had limited tolerance for service delays and, therefore, easily became frustrated when process performance was poor (e.g., when prompted to repeatedly enter the same information in several places). Redundant data also results in end user dissatisfaction. Based on these results, we propose the following DP:

DP11: Computational designs should focus on service process performance.

Simplicity of service processes

For this DP, three key issues emerged: 1) process and operations management (OM), 2) usability, and 3) simplicity. The participants recognized the ability to assess operational activities as critical for designs and emphasized streamlined operational activities via the use of easily memorized codes (e.g., a certain type of booking) and shortcuts (e.g., for making a booking) related to actual service processes. The design should focus on providing users with simple and easy-to-use processes. Based on these results, we propose the following DP:

DP10: Computational designs should focus on simple operational processes and activities.

Consistency of service delivery

For this DP, the participants considered consistency to be a critical element for high-quality information processing and service delivery. Inconsistency of data and information, as well as inputting of data and information in an inconsistent order, can cause difficulties related to merging different types of services together. The participants also recognized the difficulty of measuring the quality of the ITeS. Based on these results, we propose the following DP:

DP12: Computational designs should focus on quality and consistency in service delivery.

Adaptive DPs

Usability and customer-centricity

For this DP, the three important issues emerged from the data: functionality, customer-centricity, and usability. The participants emphasized the role of practical functions in adapting to tasks performed by knowledge workers, such as merging information from different databases. The participants also highlighted the usability since they required easy-to-use functions, hassle-free recovery options, and timely responses in their service encounters.

Positively perceived usability led participants to think that they could rely on the ITeS they use. Based on these results, we propose the following DP:

DP1: Adaptive designs should focus on usability and customer-centricity of the functionalities.

Adaptable service security

The participants valued the security aspects of ITeS when these are used in sensitive contexts, such as maintaining safety during data storage and properly controlling users' access rights to certain databases. For instance, the participants mentioned that they were concerned that users' own actions and newly introduced features could introduce security problems or breaches. They also desired freedom of choice in terms of secure service options. In practice, the participants preferred adaptive ITeS with options to choose from. Based on these results, we propose the following DP:

DP2: Adaptive designs should focus on offering adaptable service security.

Trust and relationship building

The participants believe that ITeS should enable trust and relationship building, especially when ITeS contain non-standardized tasks that require a high degree of customer-centricity to offer flexibility for individuals to use the solutions in different types of encounters. The participants stated that maintaining flexibility may be challenging because, on one hand, customer-centricity can provide ideas for new uses and features, but on the other hand, customers do not like constant changes. However, trust and relationship building can make customers more confident in using ITeS after such changes and, in turn, create customer lock-in. Thus, we propose the following DP:

DP3: Adaptive designs should focus on enabling trust and relationship building.

Customization and personalization

For this DP, the participants highlighted customization of service offerings and user interface design. The participants noted that a high degree of customization is required, since users can have a wide range of specific needs. For example, they may use ITeS for information searches that would benefit from personalized suggestions and search results. Detailed customization and personalization can offer enjoyable experiences that are specific to each user or user group. Thus, the customized service solutions should be carefully tailored to suit users' requirements. Based on these results, we propose the following DP:

DP4: Adaptive designs should focus on customization and personalization of the service.

Collaborative DPs

Interactive and responsive experience

For this DP, the participants highlighted responsiveness and convenience. They emphasized that discussing complicated and urgent issues requires personalized options and channels, such as interactive, real-time audio or video communication (in contrast to, for example, resolving complicated issues via email or feedback forms). Less urgent issues can benefit from asynchronous video communication (e.g., guides and tutorials). This supports the building of customer relationships with ITeS that are simple to use. Based on these results, we propose the following DP:

DP6: Collaborative designs should focus on interactive and responsive service experiences that facilitate the development of customer relationships.

Accessibility and knowledge sharing

For this DP, we observed two service concept issues: accessibility and sharing. The participants indicated that they seek easily accessible ITeS to support distributed work and collaboration. However, end-users often experience transaction overload when interacting with an IT-enabled service. The participants also believed that if an IT-enabled service promotes open knowledge sharing across users (e.g., in organizations), it can create a culture where everyone is encouraged and potentially willing to contribute to a common good. Based on these results, we propose the following DP:

DP5: Collaborative designs should focus on accessibility and information sharing to support collaboration.

Formal and informal communication anytime and anywhere

For this DP, the participants indicated that it is necessary to have a combination of formal and informal communication channels, such as structured forms and social networking, for various needs (e.g., urgent or non-urgent matters) and situations (e.g., using a desktop or mobile version) in order to lower the threshold for initiating an online conversation, as well as to prioritize or categorize the communicated information. The results indicate that participants want to stay connected anytime and anywhere. Based on these results, we propose the following DP:

DP7: Collaborative designs should enable both formal and informal communication anytime and anywhere.

Service recovery

For this DP, the participants emphasized service recovery. The participants reported frequently experiencing poor service recovery when problems occur. As such, they identified better service plans as important for dealing with complicated situations, particularly when the reliability of the IT-enabled service is unclear. The participants considered immediate service recovery to be crucial because immediacy could help to reduce the uncertainty of the situation. Based on these results, we propose the following DP:

DP8: Collaborative designs should focus on robust service recovery options.

Network DPs

Ease of setup and maintenance

For this DP, the participants wanted network ITeS that are easy to set up and maintain. Thus, the setup and maintenance functionalities and activities should be designed to match the skills that users already possess. This will enable users to avoid frustration, which is common during introductions to complex systems, and to employ ITeS fluently without having to learn new skills. The DP proposed based on this sub-cluster is as follows:

DP14: Network designs should focus on offering functionalities that are easy to set up and maintain.

Service efficiency

For this DP, efficiency, mainly cost and resource savings, were the most important aspects. Participants also highlighted that they aim to minimize technical support by constantly improving the quality of services. For example, one participant stated that the goal was to eliminate customer support and, in so doing, decrease costs and increase the efficiency of the service delivery system. Based on these results, we propose the following DP:

DP13: Network designs should focus on constant improvement of service efficiency.

Service availability and flexibility

For this DP, the participants required a high degree of flexibility and choice for OM and processes related to network ITeS. Furthermore, the participants stated that ITeS should be available for use in various situations (e.g., on the go), and they identified the availability of ITeS as a fundamental priority. Based on these results, we propose the following DP:

DP15: Network designs should focus on service availability and flexibility.

Service transparency

For this DP, the participants highlighted poor service performance as the biggest contributor to dissatisfaction. Users often do not understand (or want to understand) the hidden logic determining how ITeS work; their main priority is to secure benefits from the services. Therefore, users should not be provided with non-critical technical information (e.g., the numerical codes of error messages or the progress of system activities). Instead, the interface design should highlight only aspects that are relevant to the users and their activities. Based on these results, we propose the following DP:

DP16: Network designs should have a transparent service logic.

Online Appendix 2. Sub-clusters: Consequences, Values, and Keynotes for Design Principles

DP1. List of consequence and value code connections and resulting keynotes.

Consequence code	**Value code(s)**	**Key Note**
Change business structure to suit software	Easy to use & usability	Adaptive
Inability to adapt to changes	Effectiveness | Quality control of information and services	Adaptive
Easy to access	Flexibility	Approachability
Provide global service coverage	Availability	Approachability
Ability to collaborate	Connectivity and being collaborative	Collaborate
Ability to communicate effectively when needed	Being responsive and supportive	Communicate
Fast feedback circuit	Efficiency	Communicate
Inability to communicate effectively (externally)	Being responsive and supportive	Communicate
Cannot be standardized	Being adaptive | Flexibility	Complexity
Lack of specification	Customization and personalization	Complexity
Require aggregated or comprehensive information	Complexity	Complexity
Indistinct meaning by the avatars	Service quality evaluation	Confusion
Compensation is not paramount	Flexibility | Branding, pricing, and profitability	Customer-centric
Consumers focus strongly on end results only	Easy to use & usability	Customer-centric
Focus on what customers want	Knowledge management and training	Customer-centric

Freedom of opinions/choices to customers/choices	Being adaptive | Convenient | Security assurance and risk management	Customer-centric
Meet or exceed customer expectation	Customer-centric | Quality control of information and services	Customer-centric
Need to keep customers well informed	Being responsive and supportive	Customer-centric
Understand and meet customers’ needs	Availability | Targeting and relevance	Customer-centric
Bring in game features to service design	Delightful feeling/fun and entertaining | Quality control of information and services	Delightful
Delight and satisfy customers	Effectiveness	Delightful
Difficult to measure service success	Service quality evaluation	Evaluation
Better assistance in operations management (OM) activities	Efficiency	Functionality
Better hardware or software functionality	Easy to use & usability	Functionality
Difficult to merge and integrate legacy databases	Legacy systems or database integration | Quality control of information and services	Functionality
Legacy database creates redundancy	Availability	Functionality
Poor computer-based software or service performance	Simplicity	Functionality
Poor service delivery	Effectiveness | Branding, pricing, and profitability	Functionality
Restrict the service options	Targeting and relevance	Functionality
Time-consuming computer-based software process	Efficiency	Functionality
Employees have the right skills and knowledge	Knowledge management and training	Knowledge

Optimize learning through teamwork	Connectivity and being collaborative | Knowledge management and training	Knowledge
Require good social skills	Knowledge management and training	Knowledge
Ability to forecast and manage demand	Effectiveness	Plan
Ability to plan better	Connectivity and being collaborative	Plan
Ability to prioritize tasks	N/A	Plan
Attract more online traffic	Branding, pricing, and profitability | Targeting and relevance	Popularity
Build up service or website authority (popularity)	Being adaptive | Create additional value & business intelligence	Popularity
Enhance publicity	Targeting and relevance	Popularity
Gain competitive strength	Profitability | Create additional value & business intelligence | Knowledge management and training	Profitability
Suffer economic or reputation loss	Branding, pricing, and profitability | Trust and relationship building	Profitability
Facilitate problem solving	Being responsive and supportive | Flexibility	Recovery
Fix problem or recover failure on time	Effectiveness	Recovery
Poor service recovery	Reliability	Recovery
Build trust and loyalty	Trust and relationship building	Relationship

Build up business associations via social network	Connectivity and being collaborative	Relationship
Lack of relevant information	Targeting and relevance	Relevant
Incapable of timely response	Being responsive and supportive | Efficiency	Responsiveness
Provide timely and relevant responses	Efficiency | Simplicity	Responsiveness
Security management and awareness issues/keep customer assured	Security assurance and risk management	Security
Enhance information and knowledge sharing	Connectivity and being collaborative | Knowledge management and training	Sharing
Time savings	Efficiency | Simplicity	Time savings
Convenient to use	Being responsive and supportive | Convenient	Usability
Easy to setup and/or maintain	Convenient | Cost saving and resource planning | Easy to use & usability	Usability
Flexible to use	Easy to use & usability	Usability
Hard to use	Easy to use & usability	Usability
Simple and easy to use	Easy to use & usability	Usability

DP2. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Ability to collaborate		Collaborate
Freedom of opinions/choices to customers	Security assurance and risk management	Customer-centric/Choices

Avoid implementation risk	Security assurance and risk management	Risk
Security management and awareness issues/keep customer assured	Security assurance and risk management	Security

DP3. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Indistinct meaning by the avatars		Confusion
Consumers focus strongly on end results only		Customer-centric
Focus on what customers want		Customer-centric
Understand and meet customers' needs		Customer-centric
Customize service or UI to requirements		Customization
Customize service or user interface (UI) to requirements		Customization
Visualize the user experience	Customer experience centric	Experience
Poor computer-based software or service performance		Functionality
Gain innovative ideas from customers	Customer experience centric	Innovation
Build trust and loyalty	Customer experience centric	Relationship
Ability to target message right	Customer experience centric	Targeting

DP4. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Convenient to use		Convenience
Understand and meet customers' needs		Customer-centric
Customize service or UI to requirements	Customization and personalization	Customization
Simple and easy to use		Simplicity

DP5. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Easy to access	Connectivity and being collaborative	Accessibility
Ability to communicate effectively when needed		Communicate
Understand and meet customers' needs		Customer-centric
Freedom of opinions/choices to customers	Connectivity and being collaborative	Customer-centric/Choices
Delight and satisfy customers	Connectivity and being collaborative	Delightful
Better hardware or software functionality	Connectivity and being collaborative	Functionality
Provide global service coverage	Connectivity and being collaborative	Globalization
Need to keep customers well informed		Informed
Gain innovative ideas from customers	Connectivity and being collaborative	Innovation
Develop interactive service		Interactive
Enhance publicity		Publicity
Enhance information and knowledge sharing	Connectivity and being collaborative	Sharing
Time savings		Time savings

DP6. List of consequence and value code connections and resulting keynotes.

Consequence codes	**Value code(s)**	**Key Note**

Easy to access	Availability | Connectivity and being collaborative	Accessibility
Improve accessibility of internet services	Availability | Quality control of information and services	Accessibility
Change business structure to suit software		Adaptive/Flexibility
Optimize use of bandwidth	Cost saving and resource planning	Bandwidth
Create brand awareness and goodwill	Branding, pricing, and profitability	Branding
Ability to communicate effectively when needed	Communication	Communicate
Inability to communicate effectively (externally)	Communication | Effectiveness | Reliability	Communicate
Require aggregated or comprehensive information	Quality control of information and services	Complexity
Maintain service consistence		Consistence
Content control	Being adaptive	Content
Convenient to use	Convenience | Customization and personalization	Convenience
Inconvenient to use	Convenience	Convenience
Freedom of opinions/choices to customers	Being responsive | Connectivity and being collaborative	Customer-centric
Meet or exceed customer expectations	Being adaptive	Customer-centric
Template answers are not paramount	Quality control of information and services	Customer-centric
Understand and meet customers' needs	Customer experience centric	Customer-centric
Customize service or UI to requirements	Flexibility | Simplicity	Customization
Environmentally friendly solution	Quality control of information and services	Eco-friendly
Receive economic gains		Economics

Suffer economic or reputation loss	Legacy system or database integration	Economics
Delectate service experience through multimedia		Experience
Consumers' tolerant attitude or level toward service delay	Customization and personalization | Effectiveness	Feeling
Difficult to merge and integrate legacy databases	Legacy system or database integration	Functionality
Poor computer-based software or service performance	Easy to use & usability | Targeting and relevance	Functionality
Provide global service coverage	Connectivity and being collaborative	Globalization
Need to keep customers well informed	Being adaptive	informed
Develop interactive service	Delightful feeling/fun and entertaining | Simplicity	Interactive
Employees have the right skills and knowledge	Customization and personalization	Knowledge
Ability to plan better	Cost saving and resource planning | Efficiency	Plan
Attract more online traffic	Branding, pricing, and profitability	Popularity
Build up service or website authority (popularity)		Popularity
Expensive to use	Cost saving and resource planning	Price
Gain competitive strength	Reliability	Profitability
Enhance publicity	Flexibility | Branding, pricing, and profitability	Publicity
Build trust and loyalty	Knowledge management and training | Reliability | Trust and relationship building	Relationship

Build up business associations via social network	Knowledge management and training | Trust and relationship building	Relationship
Lack of relevant information	Quality control of information and services | Targeting and relevance	Relevant
Delay in information or service availability	Being responsive	Responsiveness
Provide timely and relevant responses	Efficiency	Responsiveness
Unclear responses lead to service delay and dissatisfaction	Communication | Flexibility	Responsiveness
Cost and resource savings	Cost saving and resource planning	Saving
Enhance information and knowledge sharing	Connectivity and being collaborative	Sharing
Simple and easy to use	Customer experience centric | Easy to use & usability | Simplicity	Simplicity
Ability to target message right	Targeting and relevance	Targeting
Time savings	Efficiency | Flexibility	Time savings
Time-consuming computer-based software process	Effectiveness	Time savings
Easy to setup and/or maintain	Cost saving and resource planning | Quality control of information and services	Usability
Hard to use	Easy to use & usability | Flexibility	Usability
Infrequent use of service features or functions	Easy to use & usability	Usability
Long waiting time		Waiting time

DP7. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Ability to communicate effectively when needed	Communicate	Communicate
Inability to communicate effectively (externally)	Communicate	Communicate
Provide global service coverage	Connectivity and being collaborative	Globalization
Ability to communicate informally	Communicate	Informal communication
Develop interactive service		Interactive
Unclear responses lead to service delay and dissatisfaction	Communicate	Responsiveness

DP8. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Ability to communicate effectively when needed		Communicate
Convenient to use		Convenience
Inconvenient to use		Convenience
Freedom of opinions/choices to customers	Being responsive and supportive	Customer-centric/Choices
Time-consuming computer-based software process		Functionality
Legacy database creates redundancy	Efficiency	Integration/ Functionality
Ability to plan better	Efficiency	Plan
Fix problem or recover failure on time	Efficiency	Recovery
Poor service recovery	Efficiency	Recovery
Poor service recovery	Efficiency | Being responsive and supportive	Recovery

Provide timely and relevant responses	Efficiency	Responsiveness

DP9. List of consequence and value code connections and resulting keynotes.

Consequence code	**Value code(s)**	**Key Note**
Create brand awareness and goodwill	Communication | Branding, pricing, and profitability	Brand
Ability to collaborate	Connectivity and being collaborative	Collaborate
Ability to communicate effectively when needed	Connectivity and being collaborative	Communicate
Inability to communicate effectively (externally)	Being responsive and supportive	Communicate
Convenient to use	Convenience	Convenience
Freedom of opinions/choices to customers	Information overload or underload	Customer-centric
Need to keep customers well informed	Flexibility	Customer-centric
Understand and meet customers' needs	Customer experience centric	Customer-centric
Delight and satisfy customers	Delightful feeling/fun and entertaining	Delightful
Suffer economic or reputation loss	Customer experience centric | Branding, pricing, and profitability	Economics
Cannot reverse transaction process	Flexibility	Flexibility/ Adaptive
Flexible to use	Flexibility	Flexibility/ Adaptive
Inability to adapt to changes	Being adaptive | Flexibility	Flexibility/ Adaptive

Functionality mismatch with requirements	Customer experience centric | Quality control of information and services	Functionality
Bring in game features to service design	Delightful feeling/fun and entertaining	Game
Difficult to merge and integrate legacy databases	Legacy system or database integration	Integration
Legacy database creates redundancy		Integration
Require aggregated or comprehensive information	Availability	Integration
Require certain degree of IT literacy	Knowledge management and training	IT literacy
Insufficient IT training	Knowledge management and training	IT literacy/training
Avoid communication or information overload	Information overload or underload	Overload
Poor computer-based software or service performance		Performance
Ability to forecast and manage demand	Availability | Quality control of information and services	Plan
Expensive to obtain support	Cost saving and resource planning | Branding, pricing, and profitability	Price
Expensive to use	Branding, pricing, and profitability	Price
High cost of switching to a new system or service	Flexibility	Price
Poor service recovery	Effectiveness	Recovery
Incapable of timely response	Being responsive and supportive | Efficiency	Responsiveness
Provide timely and relevant responses	Availability	Responsiveness
Unclear responses lead to service delay and dissatisfaction	Communication | Customer experience centric	Responsiveness
Lack of backup support	Being responsive and supportive	Security

Security management and awareness issues/keep customer assured	Connectivity and being collaborative	Security
Enhance information and knowledge sharing	Connectivity and being collaborative | Delightful feeling/fun and entertaining	Sharing
Cannot be standardized	Being adaptive	Standardization
Ability to target message correctly	Being adaptive | Targeting and relevance	Targeting
Long waiting time	Customer experience centric | Efficiency	Time savings
Time savings	Being responsive and supportive	Time savings
Time-consuming computer-based software process	Efficiency	Time savings
Difficult to use	Knowledge management and training	Usability
Inconsistent data and document format	Customization and personalization	Usability
Inconvenient to use	Simplicity	Usability
Infrequent use of service features or functions	Effectiveness	Usability

DP10. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Ability to communicate effectively when needed		Communicate
Better hardware or software functionality	Easy to use & usability	Functionality
Easy to learn	Simplicity	Learn
Poor computer-based software or service performance		Performance
Better assistance in OM activities	Create additional value & business intelligence | Service quality evaluation	Process and OM

Simple and easy to use	Easy to use & usability	Simplicity
Hard to use	Easy to use & usability	Usability
Infrequent use of service features or functions	Effectiveness	Usability

DP11. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Delay in information or service availability		Availability
Inconvenient to use		Convenience
Delight and satisfy customers		Delightful
Consumers' tolerant attitude or level towards service delay		Feeling
Need to eliminate iterative data and process	Efficiency | Reliability	Iterative process
Long waiting time	Efficiency	Long waiting time
Poor computer-based software or service performance		Performance
Time-consuming computer-based software process	Efficiency	Performance
Legacy database creates redundancy		Redundancy
Provide timely and relevant responses	Availability	Time savings
Time savings		Time savings
Incapable of timely response	Efficiency	Timely response
Provide timely and relevant responses		Timely response
Streamline the transaction process		Timely response
Difficult to use		Usability
Hard to use		Usability

DP12. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**

Provide accurate information	Quality control of information and services	Accuracy
Lack of backup support		Backup
Ability to collaborate		Collaborate
Inconsistent data and document format		Consistency
Maintain service consistence	Quality control of information and services	Consistency
Better hardware or software functionality		Functionality
Difficult to measure service success	Quality control of information and services	Measurement
Lack of specification	Quality control of information and services	Measurement
Poor computer-based software or service performance		Performance
Ability to forecast and manage demand	Quality control of information and services	Plan
Legacy database creates redundancy		Redundancy
Lack of relevant information	Quality control of information and services	Relevant
Cost and resource savings	Quality control of information and services	Savings

DP13. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Require certain degree of IT literacy		IT literacy
Benefit from open source features		Open source
Ability to plan better	Targeting and relevance	Plan
Cost and resource savings	Cost saving and resource planning	Savings

DP14. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Improve accessibility of internet services	Quality control of information and services	Accessibility
Understand and meet customers' needs		Customer-centric
Compensation is not paramount	Branding, pricing, and profitability	Feeling
Cannot reverse transaction process	Convenience	Flexibility
Require certain degree of IT literacy	Knowledge management and training	IT literacy
Need to eliminate iterative data and process	Easy to use & usability	Iterative
Expensive to use	Branding, pricing, and profitability	Price
Employees have the right skills and knowledge	Knowledge management and training	Skills and Knowledge
Streamline the transaction process	Convenience	Timely response
Build trust and loyalty	Quality control of information and services	Trust and loyalty
Convenient to use	Simplicity	Usability
Easy to access	Convenience	Usability
Easy to setup and/or maintain	Easy to use & usability | Simplicity	Usability
Lack of relevant information		Usability
Simple and easy to use	Simplicity	Usability

DP15. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Freedom of opinions/choices to customers	Flexibility	Customer-centric

Flexible to use	Flexibility	Flexibility
Better assistance in OM activities	Flexibility	Process and OM
Receive economic gains	Flexibility	Profitability
Easy to access		Usability

DP16. List of consequence and value code connections and resulting keynotes.

Consequence	**Value code(s)**	**Key Note**
Understand and meet customers' needs	Customer experience centric	Customer-centric
Consumers focus strongly on end results only	Customer experience centric	Focus on end results only
Gain innovative ideas from customers	Customer experience centric	Innovative
Poor computer-based software or service performance	Delightful feeling/fun and entertaining	Performance

Values **Consequences**		Create additional value & BI	Easy to use & usability	Effectiveness	Service quality evaluation	Simplicity
Ability to communicate effectively when needed	Sig. (2-tailed)	.881	.275	.061	.881	.710
Better assistance in OM activities	Sig. (2-tailed)	.005	.103	.301	.005	.088
Better hardware or software functionality	Sig. (2-tailed)	.897	.001	.692	.897	.748
Easy to learn	Sig. (2-tailed)	.941	.749	.820	.941	.000

Hard to use	Sig. (2-tailed)	.842	.000	.541	.842	.096
Infrequent use of service features or functions	Sig. (2-tailed)	.916	.047	.003	.916	.794
Poor computer-based software or service performance	Sig. (2-tailed)	.765	.716	.117	.722	.986
Simple and easy to use	Sig. (2-tailed)	.881	.000	.646	.881	.710