van Oorschot, K.E., Akkermans, H.A., Van Wassenhove, L.N. and Wang, Y. (2022), "Organizing for permanent beta: performance measurement before vs performance monitoring after release of digital services", International Journal of Operations & Production Management, Vol. ahead-of-print No. ahead-of-print. https://doi.org/10.1108/IJOPM-03-2021-0211

Organizing for Permanent Beta: Performance Measurement *Before* versus Performance Monitoring *After* Release of Digital Services

KIM E. VAN OORSCHOT[1]
BI Norwegian Business School
Department of Accounting and Operations Management
NO-0442 Oslo, Norway
phone: +4746410631
kim.v.oorschot@bi.no

HENK A. AKKERMANS
Tilburg School of Economics and Management
Department of Management
Warandelaan 2
5000LE Tilburg, The Netherlands
ha@uvt.nl

LUK N. VAN WASSENHOVE
INSEAD
Humanitarian Research Group, Technology and Operations Management
Boulevard de Constance
77305 Fontainebleau, France
luk.van-wassenhove@insead.edu

YAN WANG
Delft University of Technology
Faculty of Technology Policy and Management
Jaffalaan 5
2628BX Delft, The Netherlands
y.wang-16@tudelft.nl

---

[1] Corresponding author

Organizing for Permanent Beta: Performance Measurement *Before* versus Performance Monitoring *After* Release of Digital Services

Abstract

Purpose: Due to the complexity of digital services, companies are increasingly forced to offer their services 'in permanent beta', requiring continuous fine-tuning and updating. Complexity makes it extremely difficult to predict when and where the next service disruption will occur. We examine what this means for performance measurement in digital service supply chains.

Methodology: We use a mixed-method research design that combines a longitudinal case study of a European digital TV service provider and a system dynamics simulation analysis of that service provider's digital service supply chain.

Findings: With increased levels of complexity, traditional performance measurement methods, focused on detection of software bugs before release, become fragile or futile. We find that monitoring the performance of the service after release, with fast mitigation when service incidents are discovered, appears to be superior. This involves organizational change when traditional methods, like quality assurance, become less important.

Originality: We draw on unique empirical data collected from a digital service provider's struggle with performance measurement of its service over a period of nine years. We use simulations to show the impact of complexity on staff allocation.

Implications: The performance of digital services needs to be monitored by combining automated data collection about the status of the service with data interpretation using human expertise. Investing in human expertise is equally important as investing in automated processes.

Keywords: digital services, performance measurement, resource allocation, system dynamics

1. Introduction

Digital services are services provided by telecom, banking, and insurance companies that serve millions of consumers through mostly automated processes. Over the past decade, these IT-enabled service supply chains have transformed themselves into digital service networks (Maull *et al.*, 2012; Pathak *et al.*, 2007; Sampson and Spring, 2012). People supported by IT mainly provide the service (Ramachandran and Voleti, 2004), with a considerable amount of do-it-yourself customer input (Akkermans and Voss, 2013; Barrett *et al.*, 2015). While a dominant organization sells and provides the services, it is backstopped by a network of dozens of independent companies that continuously deliver new innovations. Today's digital services are said to live in permanent beta, where beta testing is the term used to denote the phase during which a new product is tested by end-users just prior to wide release. Digital services require continuous fine-tuning and updating (Zomerdijk and Voss, 2011).

Service innovations can in principle affect every other entity in the network and compromise reliability. The modem provider, the provider of the broadcast service, a middleware firm, a content provider, all can and will generate problems with the service provided to customers, or "service incidents". The ability to detect such incidents before a new service deploys is decreasing. Service incidents may occur through a unique constellation of multiple constituent services and programs, each of which is harmless on its own, but together result in unforeseen breakdowns.

Using concepts from decision theory, we can say that the environment of digital services has changed from a complicated context to a more complex one (Snowden and Boone, 2007). A complicated context is structured; cause and effect can be determined (Alexander *et al*., 2018). This is the domain of quality assurance (QA) experts, who measure the performance of the developed software, detect and fix bugs before the service is released to customers. A complex context is unstructured; cause and effect are only coherent in retrospect (Alexander *et* *al.,* 2018). Here, performance measurement through QA becomes ineffective. Bugs may or may not emerge over time. In such situations it may be better to release a service, monitor its performance, and solve problems as soon as possible (Arora *et al.*, 2006; Choudhary and Zhang, 2015; Guo and Ma, 2018).

Changing environments thus require organizations to re-align their strategy and performance measures (Alexander, *et al.,* 2018; Hanson, *et al.,* 2011; Melnyk, *et al.*, 2014; Micheli and Mura, 2017). However, research about the implications for resource allocation and the roles of employees is scarce (Bowen, 2016; Ostrom *et al.*, 2015). With increasing levels of complexity, should digital service providers reduce pre-release QA activities and instead invest in post-release performance monitoring? We have therefore formulated this research question: What is the relative effect of performance measurement *before* and performance monitoring *after* release of digital services on the distribution of staffing?

To investigate, we employ a mixed methods research design (Creswall and Clark, 2007; Johnson *et al.*, 2007). We combine a longitudinal case study of a telecom company providing digital TV services with a system dynamics model of this company's innovation, QA, and operations processes, and associated staffing policies. Our findings in both our empirical analysis and our simulation experiments concur: As the possibility to accurately measure performance before release decreases, allocating staff to QA becomes increasingly futile, and monitoring performance after release becomes the superior policy. It is then more effective to quickly discover and resolve service incidents than to search for the bugs that caused the incidents. This policy is the opposite of much recent and current industry practice.

Our findings contribute to the literature in several ways. We provide a better understanding of how strategy and performance measurement approaches should be connected in response to changing environments (Alexander *et al.,* 2018; Melnyk *et al.,* 2014; Snowden and Boone, 2007). We also examine what this means for resource allocation. We show a shifting dominance of organizational processes in response to increasing levels of complexity. Organizing for permanent beta also implies design choices on resources. We provide more knowledge about how performance measurement impacts lower levels of the organization, i.e., at the function or group level (Bourne *et al.,* 2018). Finally, we offer insights into what could happen when environments evolve further, from complex to chaotic.

2. Theoretical Background

2.1 Increasing complexity of digital services

Telecom, banking, and insurance companies serve millions of customers through automated service processes, with human resources used for direct customer contact, fixing problems, and developing new functionality. Such a system can be called a digital service supply chain, which is defined as a network of interactive service processes (Maull *et al.*, 2012; Sampson and Spring; 2012). Another term used is a service system, which is “an interactive configuration of various resources and their mutual exchange to facilitate value cocreation that is institutionalized and regulated through institutional logics and standards” (Eaton *et al.*, 2015, p. 218).

Digital service supply chains have transformed themselves into digital service networks (Maull *et al.*, 2012; Pathak *et al.*, 2007; Sampson and Spring, 2012), with challenging consequences. Behind the organization that sells and provides the service are dozens of independent companies that deliver innovations. All can contain software bugs that lead to service incidents. Service incidents are unplanned interruptions or reductions in quality of IT services (Fanning, 2008). An incident initiates a request for maintenance. Software maintenance is the modification of a software system after delivery to correct faults, improve performance, or adapt to a changed environment (Banker *et al.*, 1998). While sometimes one software component fails, more often it is the interaction of (software and hardware) components that causes trouble and leads to unforeseen breakdowns. Incidents can arise anywhere in the network, and causes become difficult to detect.

The interaction of components that contribute to a service is also called structural complexity, one of the five dimensions of complexity (Damasiotis *et al.*, 2018; Geraldi *et al.*, 2011). Behind these services is a network of organizations interacting with each other. This is referred to as socio-political complexity. Activities are carried out by human actors in different organizations or different units within the same organization, with potentially conflicting interests (Geraldi *et al.*, 2011; Pathak *et al.*, 2007). Interactions between components and companies create uncertainty, a third dimension of complexity. Dynamics, a fourth dimension, refers to changes in requirements, software code, innovations, human resources, or the environmental context (Wang *et al.*, 2018). Beyond rework, organizational strategies, goals, performance measurement, and resource allocation may be obliged to adapt (Hanson *et al.*, 2011; Melnyk *et al.*, 2014). The last dimension of complexity relates to pace: the urgency and criticality of time goals (Geraldi *et al.*, 2011; Mendelson and Pillai, 1999). Customers of digital services often expect high innovativeness and reliability. These two goals conflict with each other when new functionalities embark software bugs that reduce reliability. Meanwhile, service incidents are known to negatively effect customer loyalty, and customers expect swift recoveries (Sousa and Voss, 2009). That adds more urgency and complexity for service organizations. Complexity obscures the perception and understanding of information cues and the functionality of the software underlying a service (Banker *et al.*, 1998). Consequently, increasing levels of complexity challenge performance measurement of digital services.

2.2 Performance measurement in changing environments

Researchers have argued that most knowledge about performance measurement has been captured from organizations operating in stable environments, and that similar knowledge is required for dynamic and complex environments (Bititci *et al.,* 2018; Micheli and Muctor, 2021). A performance measure is defined here as the qualitative or quantitative assessment of the efficiency and/or effectiveness of action (Bititci *et al.,* 2018). Not all organizations recognize the need to change performance measures when environments change. Measures that were successful in stable and structured environments could be counterproductive in complex and unstructured environments (Hanson *et al.*, 2011). This is seen as the “alignment problem”. Performance measurement is most effective when it aligns with elements such as business strategy, organizational culture and the external environment (Melnyk *et al.*, 2014). To explain the relationship between strategy and performance measurement, Melnyk *et al.* (2014) developed the performance alignment matrix, consisting of two dimensions. *Outcomes* represent organizational goals or visions. *Solutions* represent organizational approaches to delivering those outcomes. Both outcomes and solutions range from general (broad understanding) to specific (fairly good idea). It is assumed that general outcomes are more appropriate in fast changing environments as generic goals provide managers with some strategic flexibility (Melnyk *et al.,* 2014). Alexander *et al.* (2018) made the relationship between the performance alignment matrix, the external environment, and managerial decision-making more explicit by connecting it to the Cynefin framework (Snowden and Boone, 2007). The Cynefin framework characterizes environments by the nature of the relationship between cause and effect: simple, complicated, complex, chaotic and disordered. Each environment requires decision makers to diagnose and act in appropriate ways (Snowden and Boone, 2007). Failing to do so entails what Melnyk *et al.* (2014) call misalignment (Alexander *et al.*, 2018).

Table I summarizes these previous findings in a matrix, in which each cell describes: (1) the environment according to the Cynefin framework, (2) the strategy and performance measurement aligned to this environment, and (3) examples of performance measurement approaches deployed by digital service providers (these approaches will be explained in the next subsection).

----- INSERT TABLE I AROUND HERE -----

2.3 Environmental context and performance measurement in digital services

Table I suggests that with increasing levels of complexity in digital services, organizational strategies and approaches for performance measurement should be adapted to maintain a fit with the changing environment. In this section we present a brief overview of different performance measurement approaches used in software development, an important process in the digital service supply chain.

Simple environment with measurement-driven management. Measurement-driven management is suitable in stable environments where the method is fixed, and the outcome can be specified. It is measurement after the fact (Melnyk *et al.,* 2014). Software development can be organized like a waterfall: the different phases of innovation, QA, operations are executed sequentially and by separate teams of specialists. These plan-based methods emphasize anticipating changing conditions (Austin and Devin, 2009). This requires standardization of component interfaces and other structures, and reduction of interdependence between and within components. Performance measurement by QA experts can be fine-tuned and optimized. It is a “quality first” approach (Aby-Shararah and Rich, 2018). Cause and effect are familiar and obvious (Alexander *et al.*, 2018).

Complicated environment with outcome-driven solutions. In this context, the outcome is clearly specified (e.g., no bugs remaining in the software before release), but the solution is only outlined in general terms (e.g., QA is required to detect bugs). The waterfall approach, fit for stable and simple environments, is seen as inadequate for large software systems (Royce, 1987). Larger systems take a longer time to develop, which extends the time between the different phases of innovation, QA, and operations. Mistakes made in the innovation phase may not be detected before the end of the QA phase.

Incremental or agile methods can overcome the challenges caused by long and sequentially executed innovation and QA phases. In agile methods, software is developed, tested (QA), and released in incremental parts, as opposed to the “big bang” releases of waterfall methods (Holweg and Maylor, 2018; Maruping *et al.*, 2009). Similarly, the DevOps concept – continuous integration between software development (Dev) and operational deployment (Ops) (Holweg and Maylor, 2018) – addresses an increasing disconnect between the development and operations functions within large software companies. Combinations of agile and DevOps methods are known as “continuous * (star)” methods (Fitzgerald and Stol, 2017). The star can stand for many activities in development and operations, and continuous delivery is among the most prominent. Continuous delivery maintains close collaboration between development and operations, but squeezes development cycles to their limits (Fitzgerald and Stol, 2017). Agile and DevOps methods proceed from the same mindset as waterfall methods: Although they may not be obvious, cause and effect are knowable (Alexander *et al.*, 2018), and bugs can be detected by QA and eliminated pre-release.

Complex environment with solution-driven outcomes. In complex environments, the outcome may not be clearly specified (e.g., the reliability of the digital service should be high), but the methods that will be used can be specified (e.g., QA or performance measurement before release needs to be supplemented with performance monitoring after the release of the service). Here, it is more important to specify what to do instead of what the result should be (Hanson *et al.*, 2011).

There are many industries in which the performance and uptime of physical technical assets is crucial, such as in chemical and energy industries, infrastructure, aerospace, and shipbuilding. None of these sectors shares the belief that it is possible to design and make products that will not fail once they are deployed, because of the many interactions in the tightly coupled systems formed by these complex technical assets. Enabled by greater availability of data through IoT (Schwab, 2016) these industries have been moving toward maintenance based on continuous monitoring of the equipment's condition – its "health" (Moubray, 1997). That enables maintenance of equipment before breakdowns occur (Jardine *et al.*, 2006).

Digital service providers are currently looking at condition monitoring concepts. Organizing for condition monitoring assumes not only that products will fail, but also that the cause will be very difficult to detect before release. Conversely, cause and effect are retrospectively knowable (Alexander *et al.,* 2018). Despite emphasis on preventing service failure through QA, achieving 100 percent reliability can be impossible or cost prohibitive in these complex settings (Sousa and Voss, 2009). Thus the efficacity of performance measurement before release is reduced. There is also a risk of wasteful over-checking (Holweg and Maylor, 2018). It may therefore make sense to release a product and then fix any problems that occur (Arora *et al.,* 2006; Cavusoglu *et al.*, 2008; Choudhary and Zhang, 2015; Guo and Ma, 2018).

Chaotic environments with assessment-driven management. In these environments outcomes are only broadly described and any solution is possible as long as it is consistent with the broad goal (Melnyk *et al.,* 2014). It is not possible to measure performance compared to a specified goal; only progress assessment is possible. No cause-and-effect relationships are perceivable (Alexander *et al.,* 2018). In digital services, this situation may imply that organizations eliminate the entire QA function and rely on monitoring service performance after release.

2.4 Implications for organizing the digital service supply chain: Where does the employee fit?

The Cynefin framework combined with the performance alignment matrix can support decision makers in understanding their environments, to make better decisions and avoid problems arising from misalignment. However, previous research is less clear about how this alignment impacts human resources or resource allocation in organizations. Following a strategic change, organizations also need to reconfigure their resources and capabilities (Santos and Spring, 2013; Sklyar *et al.*, 2019). Ostrom *et al.* (2015) mentioned that research is needed on fitting together service strategies with internal organization to drive positive customer experiences.

Where does the employee fit in increasingly complex service systems (Bowen, 2016)? When organizing digital service supply chains, employees may be assigned new roles in front-end and back-end units (Sklyar, *et al.,* 2019), where the back-end is related to innovation and QA (before release of the service) and the front-end to operations (after release). It remains unclear what this means for the traditional functions of software development (innovation, QA and operations). When increasing levels of complexity require more performance monitoring *after* releasing the service, should digital service providers reduce QA activities and, if possible, reallocate QA staff to condition monitoring? Lack of knowledge about organizing digital service supply chains in complex environments motivates our research question: What is the relative effect of performance measurement *before,* and performance monitoring *after* release, on the distribution of staffing? In the next section we describe the mixed method we used to analyze it.

3. Method

3.1 Mixed methods: combining empirical and model-driven research

Our research question calls for a combination of a qualitative study and a quantitative model-driven analysis. Combining these approaches can provide a better understanding of research problems and complex phenomena than either approach alone, incorporating the strengths of both methodologies and reducing some of the problems associated with singular methods (Creswell and Clark, 2007). The empirical component is needed to ground this research in reality. A longitudinal case study is preferable (Eisenhardt, 1989; Yin, 1984; Åhlström and Karlsson, 2016) since it allows us to closely study a supply chain that experiences increasing levels of complexity.

However, empirical research cannot look beyond what actually happened. Here simulation modeling is beneficial, since the models can perform both history-replicating and history-divergent simulations (Malerba *et al.*, 1999). In the latter case, key parameters of the model are modified to explore conditions that have not (yet) occurred. Simulation models also allow for more precise description of tipping points in behavior. In short, simulation models can generate new theory (Davis *et al.*, 2007).

3.2 Empirical research component

The empirical component of this research is based on a longitudinal study of a telecom services company (referred to as “TeleSP”), particularly its digital TV services. TeleSP invited one of the authors to study and address potential problems associated with its growth. This author subsequently introduced another, to study innovation and operations processes of digital TV services.

We followed one of TeleSP’s biggest TV innovation projects during 2014–2016. The data collection activities included 42 semi-structured interviews with engineers, consultants and managers from innovation, QA, and operations in three rounds, with an average of six months between rounds; three model-building workshops (Vennix, 1996); project meetings twice a week for around one year; and company documents containing relevant historical data. More information about the interviews and workshops is provided in the online Supporting

Information[2]. All interviews, workshops, and meetings were recorded and partly transcribed. Two years later, in 2018, one of the authors interviewed senior management of TeleSP in two separate meetings to reflect on changes in the ensuing period and the outlook for coming years.

3.3 Model-driven research component

System dynamics (SD) models capture information feedback and time delays to allow the simulation of complex and dynamic behavior (Forrester, 1961; Senge, 1990; Sterman, 2000). They can model complex business decisions with real-world characteristics (Davis *et al.*, 2007; Größler *et al.*, 2008). SD is an excellent method for our research, for three reasons. First, our work involves multiple interacting feedback processes between innovation, QA, and operations. Second, time delays and nonlinear effects between decisions and the consequences of these decisions need to be considered in a digital service supply chain. Finally, a fundamental trade-off exists between investing in innovation, QA, or operations, assuming the organization lacks unlimited resources.

The input required for developing the model (feedback loops, conceptual structure, values of exogenous variables) was based on three group model-building sessions with management of the focal service in 2014 and 2015 and on our interviews during that same period. Previous research used an earlier and limited version of the model (Oorschot *et al.*, 2018). The model was updated after 2018 to accommodate the effects of increasing levels of complexity.

---

[2] The online supporting information can be accessed at: www.kimvanoorschot.info/publications

4. The Case of Digital TV Services

4.1 Introduction

TeleSP is a medium-sized telecom and ICT service provider in Northern Europe. The company offers fixed-line and wireless telephony, internet, and TV to individual customers, and end-to-end telecommunications and ICT services to business customers. To deliver high-quality telecom services, TeleSP runs a complex digital service supply chain (including all five dimensions of complexity: structural, uncertainty, dynamics, pace and socio-political (Geraldi, *et al.*, 2011). Constant IT innovations, and a great deal of IT-enabled service operations are involved.

The business unit we focus on, Digital TV Services, launched in 2007. Over the years, TeleSP grew the business steadily, despite various challenges. First, providing a TV service was outside the traditional realm of telecom providers, and telecom incumbent TeleSP was merely a startup amidst the established cable companies in this segment. Second, the load requirements on transmitting video signals were much heavier than telephony or normal PC use. Third, the performance requirements for TV are much higher than for ordinary broadband use: Customers watching a TV program will notice just a few seconds of service interruption, while a browser may lose its connection repeatedly without a customer noticing. TeleSP's market performance could drop because of lagging functionality and insufficient innovativeness, but its market performance could also go down because of service incidents, often caused by recent innovations in the service.

4.2 Increasing levels of complexity

After around 2010, a new challenge emerged in the digital TV market: the rise of video on demand service providers, with Netflix emerging as the clear winner. These "single play" competitors did not have the burden of making their services consistent with an installed base of legacy systems. Nevertheless, TeleSP managed to sustain a prolonged period of significant growth in the years after 2010.

At this point, our direct involvement as researchers with the Digital TV unit began. The third author and the first author analyzed archival records of 11 major service incidents from mid-2011 to late 2012. These were so-called Code Orange (fairly severe) incidents, based on several criteria (number of customers affected, duration, risk of escalating). During this time, no Code Red (major nationwide impact) incidents. Table II summarizes these 11 service incidents.

----- INSERT TABLE II AROUND HERE -----

Some observations regarding these incidents underline the complexity of this digital service supply chain. First, the location where the incident started varied greatly. It could be in a third-party device that customers keep in their homes to communicate between the service and the devices in the house. It could be a third-party database, or simply human error. Second, the biggest source of incidents was innovation. Seven out of 11 incidents were a result of an update or release. Third, in nine out of 11 incidents the root cause was not known beforehand (cause and effect were retrospectively knowable). Staff were taken by surprise. In 10 of the 11 cases, neither the innovation nor the operations team heard of the incident before the customer care center inquired about a problem reported by users. In other words, customers discovered the performance issue before TeleSP did.

The operations team was not well prepared for these incidents because it was unfamiliar with their causes. Their technical knowledge was inadequate, because for most incidents, root causes were undiscovered bugs in recent innovations. During the diagnostic phase of incident management, it was often up to one or two in-house senior experts to oversee the entire digital service supply chain and to identify the likely root cause. Root cause analysis was often not conducted, because operations staff endured time pressure. Removing direct customer impact is their top priority: Service must be restored as soon as possible. As multiple major and minor incidents occurred, there was often no time to conduct a root cause analysis with the innovation team.

4.3 Changing performance measurement (QA) in response to increasing complexity

Digital TV's management recognized from the incident analysis that incidents peaked following innovative new releases. As a countermeasure, the management team of Digital TV moved staff temporarily from innovation to operations to help clear up issues resulting from innovations, and resolve existing problems that had not yet escalated into incidents. However, innovation staff eventually had to return to their posts; otherwise the rate of service innovations would fall below target, and a key driver of market performance would be harmed.

To structurally solve coordination issues between innovation and operations, Digital TV management made two organizational design changes. Firstly, in 2014 an agile way of working was promoted. Secondly, from 2015 onwards, the DevOps method was implemented, where innovation and operations staff were grouped into a single unit. As the senior manager responsible for service quality at TeleSP reflected in 2018: *"DevOps at TeleSP is really what you would always do during a crisis, standard. So, assemble a multidisciplinary team, both operations and innovation. No split responsibilities. Then you see that this shared responsibility also helps marketing and innovation to better understand operational stability. [...] The customer still notices at the front end that the rate of innovation is high, but the back end remains stable."*

The agile and DevOps methods seemed to work, as the number of Code Orange incidents dropped during these years, even though the customer base multiplied, and the functionality of Digital TV increased greatly. However, Code Red incidents started to occur, like a hacking effort, and a blackout of the TV service on Friday night, during *The Voice of [Country X]*, the most popular live show at the time, attracting millions of viewers. The hack led to a complete overhaul of existing infrastructure, but also to the development of new features, as it became clear that security had to be built into every new functionality.

In recent years (from 2018 onward), TeleSP is increasingly "taking the human out of the loop", as several of the other telecom service companies are doing (Baroudy *et al.*, 2019). The increased occurrence of Code Red incidents suggested that less human expertise was necessary in QA, because these incidents cannot be foreseen. The performance of the digital service is increasingly being monitored *after* release. TeleSP surveils anomalies internally and corrects them before significant customer impact occurs and customers inform TeleSP. As one manager commented: *"The network of the future is one where all the time all sorts of indicators are monitored, with fully automated control processes, a self-healing network."* However, human expertise is required to monitor and make sense of the collected data.

What does this mean for Digital TV's staff requirements now and in the future? Are more staff required in operations to monitor the service after a new release? If so, what happens to QA staff when performance measurement before release is increasingly difficult? TeleSP could not answer these questions, which formed the starting point for the model-based component of our research.

5. Causal Loop Diagram and Simulation Model

5.1 Feedback loops in the digital service supply chain

The semi-structured interviews outlined in Section 3.2 provided us with the necessary information to develop a causal loop diagram and a stocks and flows diagram (simulation model) of the most important processes in the digital service supply chain, and the relationships between these processes. The causal loop diagram that explains the trade-offs and consequences of resource allocation is depicted in Figure 1. We have assumed that there is a constraint to the extra staff that can be hired following a decision to change the target service innovativeness. So, when more staff is hired in innovation, the possibility to hire staff in QA and operations is reduced, and vice versa. This trade-off is depicted by the three balancing loops (all three are called "resource allocation") at the top of Figure 1. The other three balancing loops describe the effects of more staff in a particular process. In innovation, more staff can develop more functionalities, which increases service innovativeness of the organization and brings the organization closer to its target. As new functionalities are developed, bugs are generated. This increases the need for QA staff. When more staff is allocated to QA, more bugs can be discovered and resolved. The variable "bug discovery potential" operationalizes the complexity of digital services (explained in Section 5.3). The higher the potential, the more bugs can be discovered by QA staff before release. This increases the need for QA staff and reduces the likelihood that bugs will cause service incidents. Service incidents require operations staff, so the higher the number of incidents (low service reliability) the more operations staff is required. An increase in staff leads to an increase of incidents resolved or managed, which increases service reliability again. Finally, both service innovativeness and service reliability influence the overall market performance as experienced by customers.

----- INSERT FIGURE 1 AROUND HERE -----

5.2 Software bugs in a digital service supply chain

Following convention, we represent our SD model using a stocks and flows diagram (shown in Figure 2) to map relevant variables, associated interrelationships, and delays. The modeled processes in the digital service supply chain are innovation, QA, and operations. Figure 2 represents the activities within these processes as a collection of stocks (rectangles) connected with flows (double arrows with valves). Feedback between these processes is shown with causal links (single arrows). (Note that we have not included all variables in Figure 2.) The causal loop diagram, the model description, and a complete list of all variables, equations, and values used for constants can be found in the online Supporting Information. Note that because the structure of the model and its equations are grounded in literature and parameter values are based on empirical data, the evidence level of our model is at least medium, according to the classification scheme for system dynamics models suggested by Homer (2014). Further validation steps are also described in the online Supporting Information.

Our model is an extended version of one used in previous research (Oorschot *et al*., 2018). The earlier model assumed a very low level of complexity and fixed allocation of resources to activities within the three processes of the digital service supply chain. The model was updated after 2018 to accommodate increasing levels of complexity, in line with new insights and challenges raised by TeleSP. Furthermore, the updated model allows us to experiment with resource allocation on a more detailed level: Beyond resource allocation to innovation, QA, and operations, we can examine the different activities *within* these three processes. As such, we can analyze more precisely which activities require more or fewer resources as complexity increases. To model increasing levels of complexity, we introduced a new stock of undiscoverable undiscovered bugs (see Figure 2), which is influenced by a new variable called the *bug discovery potential*, as we discuss in the next subsection.

The central construct in our model is software bugs. These are mistakes made by innovation staff or failures that arise from the interconnectedness of different software components. The development of new software products will lead to the generation of bugs (Arora, *et al.*, 2006), but when these bugs remain undetected, they can multiply and regenerate into even more bugs (Abdel-Hamid and Madnick, 1991; Westland, 2004). This cycle can cause longer development times, higher costs, and lower quality (Akkermans and Van Oorschot, 2016). Bugs that remain undetected after release reduce service reliability and may put the organization's survival at risk, especially if reliability is of high importance to users (Lee *et al.*, 2018; Sousa and Voss, 2009).

The three main processes in Figure 2 are all related to bugs. Bugs are generated during innovation, and hopefully fixed during QA; if not, operations will face service incidents. Below we describe specific processes and activities, followed by independent and dependent variables.

----- INSERT FIGURE 2 AROUND HERE -----

5.3 Digital service supply chain processes

Innovation is responsible for the development of new services, adding new functionality and improving existing features. Management sets the target for the innovativeness of the service, which drives the number of innovation projects. New services typically flow through three stages of innovation. First, they are in the pipeline (work in process); when finished, they are introduced to the market (recent innovations); then, after a maturity delay, they become part of the innovation infrastructure.

Quality assurance consists of three activities. First, undiscovered bugs made during innovation must be discovered through performance measurement. The bugs then must be fixed rapidly, to prevent them from becoming incidents that are noticed by customers. Bugs with a quick fix flow into the stock "patched bugs". Finally, a structural solution for the bug is developed and implemented that permanently removes it from the digital service. The higher the complexity, the lower the likelihood that all bugs can be discovered by the QA staff. We call this likelihood the *bug discovery potential*. For example, if the potential is 95 percent, then five percent of all undetected bugs cannot be discovered, regardless of QA efforts. These bugs, together with any bugs that do not yet receive a structural resolution, may activate service incidents after release to customers.

Operations is responsible for the reliability of the service through condition monitoring and incident management. Monitoring helps to discover potential incidents before the customer does. Service incidents that occur require highest priority from the operations staff. The longer it takes to resolve incidents, the lower the service reliability and the lower the resulting market performance of the service.

5.4 Allocation of staff

Each of the six activities in this digital service supply chain (innovation, bug discovery, bug fixing, structural resolution, condition monitoring, and incident management) has dedicated staff (note that staff are not shown in Figure 2). The capacity that is available for each activity is determined by the number of staff allocated to the process, multiplied by the average productivity of staff in this process. The required capacity is determined by the work that needs to be done (level of the stock), divided by the average productivity.

5.5 Independent and dependent variables

The independent variables in our model are those we use to define the different scenarios.

- Target service innovativeness: All simulation scenarios will start in Week 0 in equilibrium: The performance of the digital service supply chain is completely stable. This equilibrium arises with a target service innovativeness of 0.3 (on a scale of 0 to 1). As TeleSP needed to keep up with innovative competitors, we evaluate the effect of an increased target. We therefore simulate a step increase of this target service innovativeness at Week 50, from 0.3 to 0.4 (33 percent increase), and analyze its effects on staffing and market performance.

- Bug discovery potential: This is a proxy for service complexity, as higher levels of complexity will make it more difficult to discover software bugs. We will run simulations with bug discovery potentials between 100 percent and 5 percent.

The dependent variables in our model – the variables that we will use to compare different scenarios – are listed below.

- Market performance: This variable measures how the market (customer) evaluates the overall performance of the service over the simulation period. Market performance is determined by service innovation and service reliability. Deploying new services rapidly has become increasingly important for competitiveness (Wang *et al.*, 2018). Furthermore, quick service recovery is vital to maintain customer loyalty; therefore, it is also important to guarantee a continuous and reliable service (Miller *et al.*, 2000; Sousa and Voss, 2009).
- Hiring of new staff members: Increasing target innovativeness means that more new functionalities of the service must be developed, quality-checked, and released. As such, we assume that a 33 percent increase in target innovativeness is combined with a 33 percent increase in staff. However, it is not necessary to add staff in a balanced way. That is, some of the six activities in the digital service supply chain may need more than 33 percent and some may need less. We will optimize market performance under the condition that no more than 33 percent of total staff is hired. As such, the simulation model will find out which activity requires the most staff.

6. Simulation Results

6.1 Increasing target service innovativeness

Each of the scenarios we simulated starts with the same initial and stable situation, in which the entire system is in equilibrium. This means that all stocks that were shown in Figure 2 show stable behavior over time (no change), and the market performance is 0.50 from Week 0 until

Week 50. Then, in Week 50 we introduce a change in the target service innovativeness. The target is increased with 33 percent, implying a 33 percent increase of work, accompanied by a similar increase in staffing levels. The initial values for staff working on each activity are given in the second column of Table III. Initially, QA is by far the biggest business process and requires the largest proportion of the total staff (26 out of 38.2 people).

First, we simulated a scenario in which the bug discovery potential is 1 (100 percent). An efficient hill-climbing algorithm was used to search the parameter space for the best allocation of 33 percent extra staff during the remaining 250 weeks (from Week 50 to 300) to maximize the average market performance (Kauffman, 1993). The hill climbing algorithm is suitable because the goal is to simulate search behavior in managerial decision making (Sommer and Loch, 2004). The optimization function is as follows:

$$\max \sum_{t=50}^{300} Market\ Performance\ (MP)$$

The results are shown in the third column of Table III. The number of innovation staff needs to increase by 31 percent (from eight to 10.5 people). The number of QA staff also needs to increase by 28 percent (from 26 to 33.3). Although operations is still the smallest process in terms of people, its staff needs to increase by almost 70 percent (from 4.2 to 7.1) to ensure that service reliability is maintained. To protect service reliability, condition monitoring becomes increasingly important. Staff needs to increase by 343 percent (from 1.4 to 4.8) to monitor the condition of the service and eliminate potential incidents before they occur. As a result, the number of staff allocated to incident management can be reduced by 17 percent (from 2.8 to 2.3). This allocation will lead to an average market performance of 0.62 (23 percent increase).

----- INSERT TABLE III AROUND HERE -----

6.2 Increasing complexity by decreasing the bug discovery potential

The scenario discussed in the previous subsection assumed that all undetected bugs can be detected before an innovation is released. However, as our case study showed, with increasing levels of complexity more Code Orange and Code Red incidents will occur. Some bugs simply cannot be detected, regardless of staff allocated to discovery. Therefore, the likelihood that the *bdp* will remain 1 in the future is low. We therefore repeated our simulations with decreasing values of the *bdp*. Table III shows the results in terms of staff that should be allocated to each activity to maximize market performance. Figure 3 depicts the relative staffing size (compared to the total number of staff) for each activity.

----- INSERT FIGURE 3 AROUND HERE -----

6.3 Discussion of simulation results

The simulation results shown in Table III and Figure 3 reveal the following:

1. Average market performance decreases with decreasing levels of the *bdp*. The more difficult it is to discover undetected bugs (lower *bdp*), the higher the likelihood that incidents will occur that reduce service reliability, which impacts market performance. When the *bdp* is lower than 0.20, it is not possible to increase market performance beyond its value when target innovativeness was still low. In this situation, a higher performance on innovation is counteracted by a lower performance on reliability.
2. The staffing size of innovation is fairly stable. Innovation staff increases by 31% when *bdp* is 1, to 36% when *bdp* is 0.05. This is not surprising because service innovativeness is an important part of market performance, so a continuous investment in innovation is required.
3. However, the staffing of QA is far from constant. The number of extra staff added to the three different QA activities decreases gradually for lower values of *bdp*. In fact, for

extreme low levels of *bdp,* QA almost disappears completely. In short, when it becomes harder to discover bugs, it does not make sense to allocate more staff to this activity.

4. What does make sense when *bdp* is low is making large investments in operations, and condition monitoring in particular. As Figure 3 shows, this activity becomes the largest activity in terms of staff, relative to the other activities.
5. The numbers in Table III also show that the relative importance of processes executed *before* service release (innovation and QA), versus processes executed *after* release (operations), may shift. This is reflected in team sizes with decreasing levels of *bdp.* With a *bdp* of 1, QA has the largest team, with 65 percent of all staff, followed by innovation (21 percent) and operations (14 percent). When *bdp* reaches about 0.8, operations (25 percent) has outgrown innovation (21 percent). When the *bdp* reaches 0.5, operations becomes the largest team with 45 percent of total staff, but QA (34%) is still larger than innovation (21 percent). Finally, when *bdp* decreases even further, to levels below 0.3, QA becomes the smallest team of the organization. These numbers show that for increasing levels of complexity, i.e., decreasing levels of *bdp*, resource (re)allocations are required to such an extent that the QA team can practically disappear while the operations team swells.

7. Implications

7.1 Research Implications

Our study leads to four main contributions to theory. First, our findings provide a better understanding of how strategy and performance measurement should be connected to respond to changing environments (Alexander *et al.,* 2018; Melnyk *et al.,* 2014; Snowden and Boone, 2007). Traditional approaches, appropriate for complicated environments, assume that service incidents can be prevented by measuring the performance of the system before release (the traditional function of QA). Our case study showed that when the organization was suffering from a growing number of service incidents, the typical response was to measure more and earlier during the innovation process. Our simulation results showed that QA is indeed the process requiring the largest number of staff and that this number increases further when more functionalities are introduced to the service. Performance measurement before release is characterized by control: as causes and effects of bugs can be determined before release, the service that is offered to customers will be predictable and reliable (Choi *et al.*, 2001). This is what Melnyk *et al.* (2014) referred to as an outcome-driven solution. However, as complexity increases further, strengthening QA even more only creates an illusion of control, as it becomes difficult or even impossible to detect bugs before release. Service incidents emerge, and their causes and effects can only be understood in retrospect. Our case study indicated that in these settings, instead of aiming for more control and more QA, the organization started to monitor the performance of the service *after* its release. This is referred to as a solution-driven outcome (Melnyk *et al.*, 2014). The outcome cannot be completely predicted or controlled; instead it emerges from the solution, that is, performance monitoring *after* release. Our simulation results corroborated this: With increasing levels of complexity (a reduced potential to discover bugs), the QA process becomes less dominant while operations becomes more dominant. Performance monitoring after release implies that operations staff need to apprehend early warning indicators that an unlikely event with high impact (“code red”) may be about to happen (Akkermans and Van Wassenhove, 2018). As organizations move into more complex or even chaotic environments, the importance of early warning indicators is expected to increase. Bellisario *et al.* (2021) point out that the development of effective local performance indicators depends on the interpretive work of people and requires a substantial cognitive effort. How such indicators should be developed is hardly understood and requires further research. Perhaps the field of digital services can learn from existing work in physical asset management, where it is clear that asset failures cannot be prevented solely by error-free designs. There, the move toward condition monitoring has already taken place (Jardine *et al.*, 2006; Moubray, 1997). In addition, redundancy of components is used in physical asset management to improve system reliability (Pate-Cornell *et al.,* 2004). The digital world is currently also experimenting with software redundancy to improve service reliability. Though they come from very different directions, physical asset management and digital service management appear to be converging toward a similar policy to keep performance reliable. These similarities require further research.

The shifting dominance of organizational processes in response to changes in performance measurement is a second new finding. The performance alignment matrix (Melnyk *et al.,* 2014), combined with the Cynefin framework (Alexander *et al.,* 2018) explains the relationship between environmental changes, organizational strategy and performance measurement, but it does not explain what this relationship means for resource allocation. The question "where does the employee fit" (Bowen, 2016; Ostrom *et al.*, 2015) remains unanswered. Our simulation results indicate that for increasing levels of complexity, employees need to be reallocated from performance measurement (QA) to performance monitoring (operations). Our case study confirms this as it showed that operations had insufficient capacity to fix all incidents. This entails a change in organizational structure because front-end units (facing customers) require more resources than back-end units, and new roles may be assigned to employees (Sklyar *et al.,* 2019). Our findings show that it is not sufficient to align performance measurement to changing environments; the allocation of resources needs to be aligned as well. This finding answers to the call for research about how organizations should respond when strategy and performance measurement change (Alexander *et al.*, 2018) and how these changes should be deployed (Melnyk *et al.*, 2014). Our findings suggest that as environments become more complex, organizations require both organizational and operational flexibility. The organizational flexibility relates to strategic agility (Doz and Kosonen, 2008) in which staff can be dynamically allocated to and perform different processes (e.g., moving from QA to operations). The operational flexibility relates to skills: people must be able to perform different tasks. Future research should examine how strategic agility is deployed, for example by analyzing how much staff to allocate to different processes and when to make changes in allocation.

Third, our research builds on earlier work that was based on the same digital service provider (TeleSP) and its struggle with organizing its supply chain (Oorschot *et al.*, 2018). The model we used in our previous research assumed a complicated environment, meaning that all software bugs could be detected by QA, and a fixed distribution of resources to activities *within* QA and operations. Our previous findings indicated that when innovation levels increase, all processes (innovation, QA, operations) require more staff to deal with the increased workload, but relatively more staff are required in both QA and in operations compared to innovation. We did not analyze which activities within QA and operations required more staff. The present work, however, allows for the simulation of *increasing levels of complexity* and *different staffing levels to activities within QA and operations.* Consequently, this work allows us to analyze more precisely which activity in the digital service supply chain becomes more important as complexity increases. Our simulation results show the importance of the operations process in general and condition monitoring within this process in particular. This new finding provides more knowledge about how performance measurement impacts lower levels of an organization, i.e., at the function or group level (Bourne *et al.*, 2018). It suggests that lower organizational levels should be involved in the design of performance measurement systems (Bellisario *et al.*, 2021). An important question here is how to support the cognitive readiness of staff for monitoring. Here, artificial intelligence may be a promising avenue for further research, to examine how to speed up diagnostic processes in close collaboration with human experts.

Fourth, our simulation findings provide insight into what could happen when environments change from complex to chaotic. Although QA becomes more futile for higher levels of complexity, a combination of both QA (before release) and condition monitoring (after release), seems to be best for moderate levels of complexity. This implies a balance between control and emergence (Choi *et al.*, 2001). Our simulation model also allowed us to simulate situations of extreme complexity in which only 20 percent (or less) of the generated bugs can be discovered before release. Under these circumstances, the function of QA seems to disappear completely. This resembles a chaotic environment in which no cause-and-effect relationships can be perceived, and performance can only be assessed (Alexander *et al.,* 2018; Melnyk, *et al.,* 2014). This is an interesting avenue for further research as it could be a future scenario for TeleSP and other digital service providers. The question of how to design organizations faced by environmental uncertainty goes back to the work of Galbraith (1974). It may be relevant to reexamine this work in the context of digital service supply chains. For instance, one way to deal with uncertainty or complexity in Galbraith's conceptual framework is the creation of lateral relations. In the present decade, we may view this in the context of ecosystem strategies (Micheli and Muctor, 2021), where operational data from multiple independent companies that together "run" a digital service supply chain can be shared real-time, and problem-solving across companies can happen real-time. Telecom companies have long had their network operations centers, but in Galbraith's terminology, those would have to be extended to connect with the centers of the other providers in the ecosystem.

While the fact that we studied only one case limits our findings, our simulation model is quite generic and the modeled business processes of innovation, QA, and operations will resemble the processes of other digital service providers. It remains to be seen whether the trend towards condition monitoring will expand further in the future (Baroudy *et al.,* 2019). Clearly, our work is exploratory research that may yield useful grounds for subsequent research designs.

7.2 Managerial Implications

As services become increasingly complex, creating bugs which may turn into service incidents, the relative importance of innovation and QA (processes executed *before* release) versus operations (processes executed *after* release) changes. Initially, our findings show that operations was the business process that required the fewest number of total staff. However, when complexity increases, operations will become the largest business process. Eventually, the function of QA may disappear. This will be a huge change for organizations. Assuming that not all staff working in QA can easily shift to operations, this requires a long-term hiring policy to prepare the digital service supply chain for changing staff skills.

Managers of digital services need to accept that service incidents will happen (Sousa and Voss, 2009). Agile and DevOps methods that focus on detecting failures before software is released are no longer sufficient; monitoring performance after release becomes a better policy. Digital services need to be monitored continuously and this needs to be automated; the human is taken out of that loop. However, human expertise is vital for interpreting data, making sense of them, analyzing and managing their impact (Aby-Shararah and Rich, 2018). This cannot be automated or routinized, because though we may know a lot about every software component, we do not know how all components interact with each other.

Acknowledgements

We acknowledge the support of the people who made our field research possible: Peter Claerhoudt, Peggy Corstens and Theo Wakkermans.

References

Abdel-Hamid, T.K. and Madnick, S. (1991), *Software Project Dynamics. An Integrated Approach*, Prentice Hall Software Series, Englewood Cliffs, NJ.

Afy-Shararah, M. and Rich, A. (2018), “Operations flow effectiveness: A systems approach to measuring flow performance”, *International Journal of Operations and Production Management*, Vol. 38, No. 11, pp. 2096-2123.

Akkermans, H.A. and Oorschot, K.E. van (2016), “Pilot error? Managerial decision biases as explanation for disruptions in aircraft development”, *Project Management Journal*, Vol. 47, No. 2, pp. 79–102.

Akkermans, H.A. and Van Wassenhove, L.N. (2018), “A dynamic model of managerial response to grey swan events in supply networks”, *International Journal of Production Research*, Vol. 56, No. 1-2, pp. 10-21.

Akkermans, H.A. and Voss, C.A. (2013), “The service bullwhip effect”, *International Journal of Production and Operations Management*, Vol. 33, No. 6, pp. 765–788.

Alexander, A., Kumar, M., and Walker, H. (2018), “A decision theory perspective on complexity in performance measurement and management”, *International Journal of Management Reviews*, Vol. 38, No, 11, pp. 2214-2244.

Arora, A., Caulkins, P.J. and Telang, R. (2006), “Research note – sell first, fix later: impact of patching on software quality”, *Management Science*, Vol. 52, No. 3, pp. 465–471.

Austin, R.D. and Devin, L. (2009), “Weighing the benefits and costs of flexibility in making software: toward a contingency theory of the determinants of development process design”, *Information Systems Research*, Vol. 20, No. 3, pp. 462–477.

Banker, R.D., Davis, G.B. and Slaughter, S.A. (1998), “Software development practices, software complexity, and software maintenance performance: a field study”, *Management Science*, Vol. 44, No. 4, pp. 433–450.

Baroudy, K., Kishore, S., Mahajan, N., Nair, S., Sigurdsson, H. and Sukumar, K. (2019), *Reinventing telco networks: five elements of a successful transformation*, available at: www.mckinsey.com/industries/technology-media-and-telecommunications/our-insights/reinventing-telco-networks-five-elements-of-a-successful-transformation (accessed 30 March 2021).

Barrett, M., Davison, E., Prabhu, J. and Vargo, S.L. (2015), "Service innovation in the digital age: key contributions and future directions", *MIS Quarterly*, Vol. 39, No. 1, pp. 135–154.

Bellisario, A., Pavlov, A., and Steen, M.P. van der. (2021), "The role of performance measurement in aligning operations with strategy: Sustaining cognitive processes of internal alignment", *International Journal of Operations and Production Management*, Vol. 41, No. 12, pp. 1879-1907.

Bititci, U.S., Bourne, M., Cross, J.A., Nudurupati, S.S. and Sang, K. (2018), "Editorial: Towards a theoretical foundation for performance measurement and management", *International Journal of Management Reviews*, Vol. 20, pp. 653-660.

Bourne, M., Franco-Santos M., Micheli, P. and Pavlov, A. (2018), "Performance measurement and management: a system of systems perspective", *International Journal of Production Research*, Vol. 56, No. 8, pp. 2788–2799.

Bowen, D.E. (2016), ‘‘The Changing Role of Employees in Service Theory and Practice: An Interdisciplinary View,’’ *Human Resource Management Review*, Vol. 26, pp. 4-13.

Cavusoglu H., Cavusoglu, H. and Zhang, J. (2008), "Security patch management: share the burden or share the damage?" *Management Science*, Vol. 54, No. 4, pp. 657–670.

Choi, T.Y., Dooley, K.J. and Rungtusanatham, M. (2001), "Supply networks and complex adaptive systems: Control versus emergence", *Journal of Operations Management*, Vol. 19, 351-366.

Choudhary, V. and Zhang, Z. (2015), "Patching the cloud: the impact of SaaS on patching strategy and the timing of software release", *Information Systems Research*, Vol. 26, No. 4, pp. 845–858.

Creswell, J.W. and Clark, V.L.P. (2007), *Designing and conducting mixed methods research*, Sage Publications, Thousand Oaks, CA.

Damasiotis, V., Fitsilis, P. and Okane, J.F. (2018), "Modeling software development process complexity", *International Journal of Information Technology Project Management*, Vol. 9, No. 4, pp. 17–40.

Davis, J.P., Eisenhardt, K.M. and Bingham, C.B. (2007), "Developing theory through simulation methods", *Academy of Management Review*, Vol. 32, No. 2, pp. 480–499.

Doz, Y. and Kosonen, M. (2008), "The dynamics of strategic agility", *California Management Review*, Vol. 50, No. 3, pp. 95-118.

Eaton, B., Elaluf-Calderwood, S. and Sørensen, C. (2015), "Distributed tuning of boundary resources: the case of Apple's IOS service system", *MIS Quarterly* Vol. 39, No. 1, pp. 217–243.

Eisenhardt, K.M. (1989), "Building theories from case study research", *Academy of Management Review*, Vol. 14, No. 4, pp. 532–550.

Fanning, P. (2008). *ITIL Version 3 Service Operation*, Office of Government Commerce, Buckinghamshire.

Fitzgerald, B.K and Stol, K.-J. (2017), "Continuous software engineering: a roadmap and agenda", *The Journal of Systems and Software*, Vol. 123, pp. 176–189.

Forrester, J.W. (1961), *Industrial Dynamics*, MIT Press, Cambridge, MA.

Galbraith, J.R. (1974), "Organization design: An information processing view", *Interfaces*, Vol. 4, No. 3, pp. 28-36.

Geraldi, J., Maylor, H. and Williams, T. (2011), "Now, let's make it really complex (complicated): A systematic review of the complexities of projects", *International Journal* *of Operations and Production Management,* Vol. 31, No. 9, pp. 966-990.

Größler, A., Thun, J.-H. and Milling, P.M. (2008), “System dynamics as a structural theory in operations management”, *Production and Operations Management,* Vol. 17, No. 3, pp. 373–384.

Guo, Z. and Ma, D. (2018), “A model of competition between perpetual software and software as a service”, *MIS Quarterly*, Vol. 42, No. 1, pp. 101–120.

Hanson, J.D., Melnyk, S.A. and Calantone, R. A. (2011), “Defining and measuring alignment in performance management”, *International Journal of Operations and Production Management*, Vol. 31, No. 10, pp. 1089-1114.

Holweg, M. and Maylor, H. (2018), “Lean leadership in major projects: from “predict and provide” to “predict and prevent”, *International Journal of Operations and Production Management,* Vol. 38, No. 6, pp. 1368-1386.

Homer, J. (2014). “Levels of evidence in system dynamics modeling”, *System Dynamics Review,* Vol. 30, No. 1–2, pp. 75–80.

Jardine, A.K., Lin, D. and Banjevic, D. (2006), “A review on machinery diagnostics and prognostics implementing condition-based maintenance,” *Mechanical Systems and Signal Processing,* Vol. 20, No. 7, pp. 1483–1510.

Johnson, R.B., Onwuegbuzie, A.J. and Turner, L.A. (2007), “Toward a definition of mixed methods research”, *Journal of Mixed Methods Research,* Vol. 1, pp. 112–133.

Kauffman, S.A., (1993), *The origins of order: Self-organization and selection in evolution,* Oxford University Press, Oxford.

Lee, H.K., Lee, J.S. and Keil, M. (2018), “Under perspective-taking to de-escalate launch date commitment for products with known software defects”, *Journal of Management Information Systems,* Vol. 35, No. 4, pp. 1251–1276.

Malerba, F., Nelson, R.R., Orsenigo, L. and Winter, S. G. (1999). “‘History-friendly’ models of industry evolution: the computer industry", *Industrial and Corporate Change*, Vol. 8, pp. 3–40.

Maruping, L.M., Venkatesh, V. and Agarwal, R. (2009), "A control theory perspective on agile methodology use and changing user requirements", *Information Systems Research*, Vol. 20, No. 3, pp. 377–399.

Maull, R., Geraldi, J. and Johnston, R. (2012), "Service supply chains: a customer perspective", *Journal of Supply Chain Management*, Vol. 48, No. 4, pp. 72–86.

Melnyk, S.A., Bititci, U.S., Platts, K., Tobias, J. and Andersen, B. (2014), "Is performance measurement and management fit for the future?", *Management Accounting Research*, Vol. 25, No. 2, pp. 173–186.

Mendelson, H. and Pillai, R.R. (1999), "Industry clockspeed: measurement and operational implications", *Manufacturing & Service Operations Management*, Vol. 1, No. 1, pp.1–20.

Micheli, P. and Mura, M. (2017), "Executing strategy through comprehensive performance measurement systems", *International Journal of Operations and Production Management*, Vol. 37, No. 4, pp. 423-443.

Micheli, P. and Muctor, G. (2021), "The roles of performance measurement and management in the development and implementation of business ecosystem strategies", *International Journal of Operations and Production Management*, Vol. 41, No. 11, pp. 1761-1784.

Miller, J.L., Craighead, C.W. and Karwan, K.R. (2000), "Service recovery: a framework and empirical investigation", *Journal of Operations Management*, Vol. 18, No. 4, pp. 387–400.

Moubray, J. (1997), *Reliability-Centered Maintenance*, Industrial Press, New York.

Oorschot, K.E. van, Wang, Y. and Akkermans, H. (2018). "Throbbing between two lives: Resource pooling in service supply chains." *Proceedings of the 51st Hawaii International Conference on System Sciences* (HICSS 2018).

Ostrom, A.L., Parasuraman, A., Bowen, D.E., Patricio, L. and Voss, C.A. (2015), "Service research priorities in a rapidly changing context", *Journal of Service Research*, Vol. 18, No. 2, pp. 127-159.

Pate-Cornell, M.E., Dillon, R.L. and Guikema, S.D. (2004), "On the limitations of redundancies in the improvement of system reliability", *Risk Analysis*, Vol. 24, No. 6, pp. 1423–1436.

Pathak, S.D., Day, J.M., Nair, A., Sawaya, W.J. and Kristal, M.M. (2007), "Complexity and adaptivity in supply networks: building supply network theory using a complex adaptive systems perspective", *Decision Sciences*, Vol. 38, No. 4, pp. 547–580.

Ramachandran, K. and Voleti, S. (2004), "Business process outsourcing (BPO): emerging scenario and strategic options for IT-enabled services", *Vikalpa*, Vol. 29, No. 1, pp. 49–62.

Royce, W.W., (1987), "Managing the development of large software system", in: *Proceedings of the 9th international conference on Software Engineering*, pp. 328–338.

Sampson, S.E. and Spring, M. (2012), "Service supply chains: introducing the special topic forum", *Journal of Supply Chain Management*, Vol. 48, No. 4, pp. 23–27.

Santos, J.B. and Spring, M. (2013), "New service development: managing the dynamic between services and operations resources", *International Journal of Production Research*, Vol. 33, No. 7, pp. 800-827.

Schwab, K. (2016), *The Fourth Industrial Revolution*, World Economic Forum, Crown Publishing, New York.

Senge, P.M., (1990), *The Fifth Discipline: The Art and Practice of the Learning Organization*, Doubleday Currency, New York.

Sklyar, A., Kowalkowski, C., Tronvoll, B. and Sörhammar, D. (2019), "Organizing for digital servitization: A service eco-system perspective", *Journal of Business Research*, Vol. 104, pp. 450-460.

Snowden, D.J. and Boone, M.E. (2007). "A leader's framework for decision making", *Harvard*

Business Review, 85(11): 68-+.

Sommer, S.C. and Loch, C.H. (2004), "Selectionism and learning in projects with complexity and unforeseeable uncertainty", *Management Science,* Vol. 50, No. 10, pp. 1334–1347.

Sousa, R. and Voss, C.A. (2009), "The effects of service failures and recovery on customer loyalty in e-services: an empirical investigation," *International Journal of Operations and Production Management,* Vol. 29, No. 8, pp. 834–864.

Sterman, J.D. (2000), *Business Dynamics: Systems Thinking and Modeling for a Complex World,* McGraw-Hill, New York.

Vennix J.A.M. (1996), *Group Model-Building. Facilitating Team Learning Using System Dynamics*. Wiley: Chichester.

Wang Q., Voss, C.A. and Zhao, X. (2018), "Deployment strategies for service innovation", *IEEE Transactions on Engineering Management,* Vol. 66, No. 4, pp. 514-528.

Westland, J.C. (2004), "The cost behavior of software defects", *Decision Support Systems*, Vol. 37, No. 2, pp. 229–238.

Yin, R. (1984), *Case Study Research,* Sage Publications, Beverly Hills.

Zomerdijk, L.G. and Voss, C.A. (2011). "NSD processes and practices in experiential services", *Journal of Product Innovation Management*, Vol. 28, pp. 63–80.

Åhlström, P. and Karlsson, C. (2016), "Longitudinal field studies", in Karlsson, C. (Ed.), *Research Methods for Operations Management,* Routledge, London, UK, pp. 198–232.

Table I. Performance alignment matrix positioned in four environmental contexts

		Outcomes: Organizational goals, strategies	
		General	Specific
Solutions: Approaches for delivering goals, strategies	General	1. Chaotic environment: no cause-and-effect relationship is perceivable 2. Assessment-driven management 3. Only PM after release of service?	1. Complicated environment: cause and effect can be determined but not obvious 2. Outcome-driven solutions 3. Agile/DevOps and continuous delivery: PM/QA during innovation but before release of service
	Specific	1. Complex environment: cause and effect are only coherent in retrospect 2. Solution-driven outcomes 3. Condition monitoring and continuous delivery: PM/QA during innovation and after release of service	1. Simple environment: cause and effect are familiar and obvious 2. Measurement-driven management 3. Standardization and sequential delivery: PM/QA after innovation but before release of service

1. Description of the organizational environment according to the Cynefin framework (Alexander *et al.*, 2018; Snowden and Boone, 2007)
2. Description of the organizational strategy and performance measurement according to the performance alignment matrix (Melnyk *et al.*, 2014)
3. Description of approaches used in software development, consisting of innovation, performance measurement (PM)/quality assurance (QA) and operations

Table II. Serious (“Code Orange”) service incidents reported at Digital TV

Date	**Incident Description**	**Root cause**	**Known/ Unknown issue**	**Situation**	**Origin of issue**
May 19, 2011	Failure in all HD channels	Reset database	Unknown	The person who chose the license was not aware of using the incorrect license.	Human
June 22, 2011	Startup SetTop Boxes (STB)	STB software set up	Unknown	No alarm received at operations despite many complaints received at incident control team.	Innovation
July 23, 2011	ITV router	DHCP server problem	Unknown	An incorrect operation was performed during an upgrade and not known due to night shift hand-over.	Human
March 15, 2012	Defect using menus by STB software bug	STB software bug	Unknown	A rare software bug was triggered.	Innovation
May 2, 2012	Problem by longer maintenance	Maintenance upgrade ran out of service window	Unknown	Complications were found in the final phase of the planned upgrade. Decision: complete upgrade with several impacts ongoing.	Maintenance
July 4, 2012	Problem HBO on Demand	Unauthorized and untested changes from supplier	Known	Problem was already reported end of June 2012, and investigation started 2 days before the incident.	Innovation
September 4, 2012	Cloud-storage service problems	STB Firmware bug	Unknown		Innovation
September 12, 2012	Streaming platform disrupted	Driver software bug	Unknown		Innovation
October 24, 2012	Crashed STBs	STB Firmware bug	Known	The bug was known to problem management who have worked on it for about one month. No major impact on customer side during the month.	Innovation
October 26, 2012	Inactive customer accounts	Wrong list of customers deleted	Unknown	An employee deleted the wrong customer list.	Human
November 3, 2012	Glitches & freezes Digital TV	Bug in Oracle database	Unknown	The database bug drained the archive space within a very short time.	Innovation

Table III. Simulation results

Allocation of staff to six different activities to maximize average market performance											
	initial values	**Bug discovery potential**									
		1.00	**0.95**	**0.90**	**0.80**	**0.70**	**0.50**	**0.30**	**0.20**	**0.10**	**0.05**
Innovation	8.00	10.513	10.610	10.681	10.659	10.714	10.584	10.796	10.876	10.892	10.898
Bug discovery	2.00	2.604	2.495	2.370	2.131	1.944	1.323	0.790	0.508	0.240	0.066
Bug fixing	4.00	5.210	4.992	4.740	4.262	3.748	2.646	1.580	1.016	0.482	0.133
Structural resolution	20.00	25.486	24.453	23.352	21.229	18.676	13.416	7.901	5.390	2.473	0.935
Condition monitoring	1.40	4.808	6.098	7.728	11.072	13.821	20.062	26.008	28.955	32.657	34.711
Incident management	2.80	2.313	2.286	2.062	1.581	2.031	2.903	3.858	4.188	4.189	4.190
Total staff	38.20	50.933	50.933	50.933	50.933	50.933	50.933	50.933	50.933	50.933	50.933
Sum of all innovation staff	8.00	10.513	10.610	10.681	10.659	10.714	10.584	10.796	10.876	10.892	10.898
Sum of all QA staff	26.00	33.299	31.940	30.463	27.622	24.368	17.385	10.271	6.914	3.195	1.134
Sum of all operations staff	4.20	7.121	8.385	9.789	12.653	15.852	22.965	29.866	33.143	36.847	38.901
Average market performance	0.50	0.617	0.615	0.611	0.602	0.588	0.557	0.518	0.506	0.499	0.497

Figure 1. Causal loop diagram of the digital service supply chain

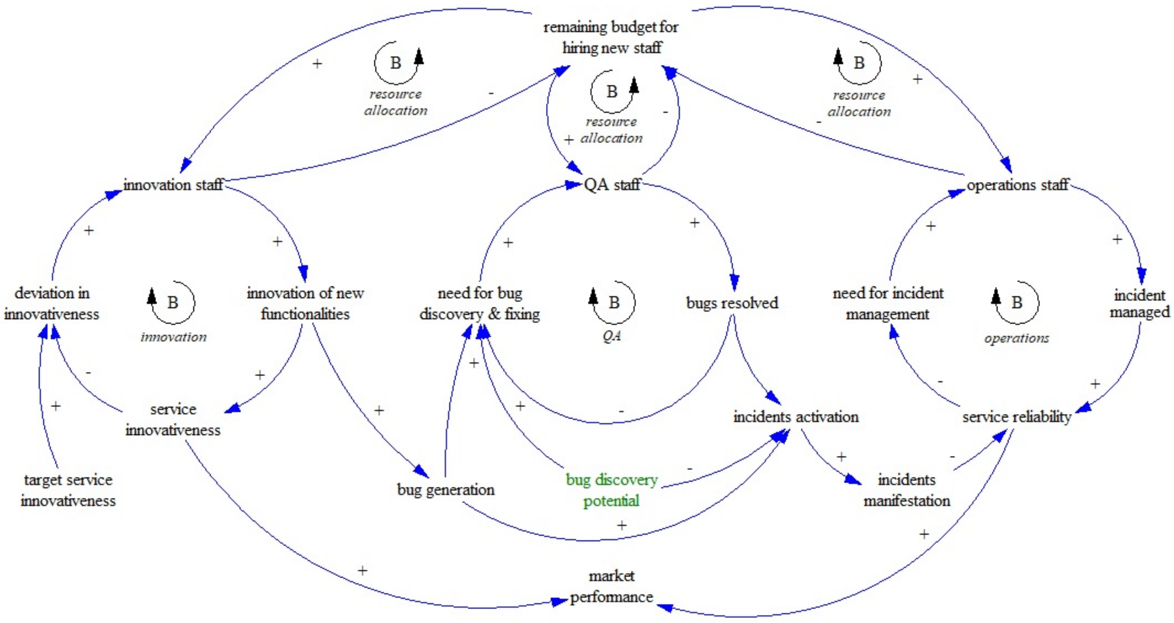

Figure 2. Stocks and flows diagram of the digital service supply chain

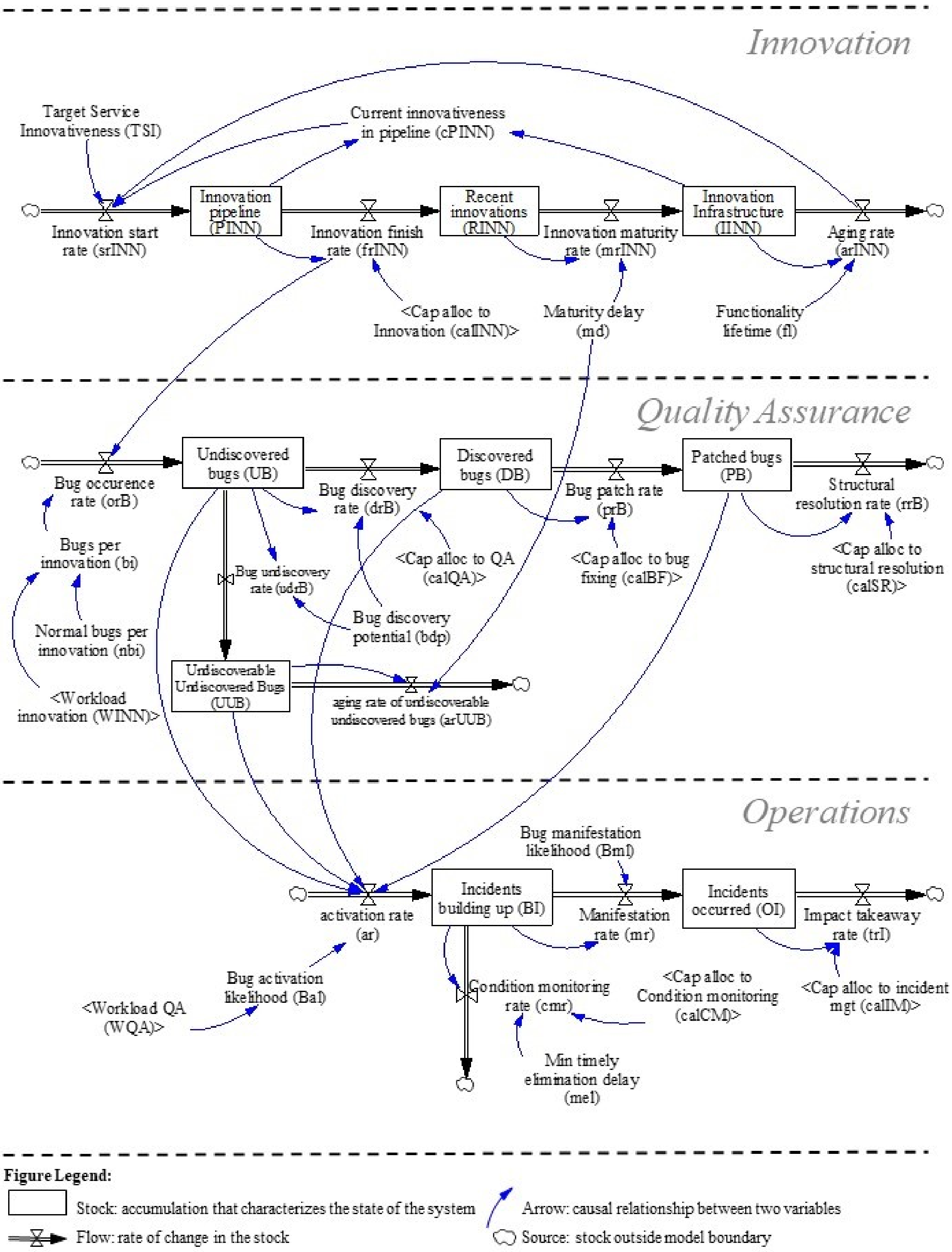

Figure 3. Relative optimal staff allocation to maximize market performance