

Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation

Anna Wiedemann
Neu-Ulm University of Applied Sciences
Center for Research on Service Sciences (CROSS)
anna.wiedemann@hs-neu-ulm.de

Heiko Gewald
Neu-Ulm University of Applied Sciences
Center for Research on Service Sciences (CROSS)
Heiko.gewald@hs-neu-ulm.de

Manuel Wiesche
Technical University of Munich
Chair for Information Systems
wiesche@in.tum.de

Helmut Krcmar
Technical University of Munich
Chair for Information Systems
krcmar@in.tum.de

Abstract

Integrating business capabilities into software development projects is still a major challenge for organizations. New ways of working are appearing in response to react to novel market places. Hence, there are more and more business managers with good IT knowledge; thus, software developers need to understand business processes. Hence, the relationship between software development, operations, and business strategy needs to be enhanced. For collecting customer perspectives in IT projects, new approaches like DevOps and BizDevOps are being used. The customer view can be integrated within software development teams through the planning processes. Our findings show that continuous innovation mechanisms are connected with the planning of customer requirements. We present planning scalability, security, and quality as rich descriptions of continuous innovation. Furthermore, we present core categories of how the customer perspectives can be integrated within a DevOps team and insights on how planning areas influence the continuous innovation mechanisms.

1. Introduction

Business departments are working in new ways develop and explore new markets around the world. Organizations are under pressure to integrate organizational agility and respond quickly to changing customer demands. Hence, agility is a

major concern in the current business world [1]. It can be supported through Information Technology (IT) [2]. A number of trends have appeared in research and practices. For example, the concept of DevOps (Development and Operations) describes the continuous collaboration of software development and operational activities to quickly provide new software components to the customer [3, 4].

The gap between business managers and software developers is a well-known problem [5] because building business capabilities is still a great challenge for IT managers [6]. IT employees like developers are often very technical and tool-oriented; they search for the easiest technical solution to problems [7]. In the past, usually IT employees had the power to make decisions regarding the management of IT projects. But this attitude has changed because more and more business people have very good knowledge about technology. For closer cooperation, technical employees should work on their business knowledge and vice versa [8].

Existing research identifies the need for a closer connection between business managers and IT employees [9]. In practice, many software developers understand the need and are willing to collaborate with the business to make strategic decisions. It is suggested that stakeholders should be integrated into the software development process at a very early stage of the software delivery lifecycle (SDLC) [3, 8]. This integration could be achieved through planning processes. Continuous planning is an important topic in recent publications [9]. In the past, planning was often combined in annual financial cycles in traditional software development projects

with very few software releases [10]. A failure in the traditional planning cycle might have necessitated another planning cycle to resolve the problem. While annual planning cycles are not unusual in traditional project settings, continuous planning is considered a key prerequisite in the context of agile development and for delivering fast and new software [8, 11, 12].

Existing research highlights that a lack of an efficient IT architecture may hinder enterprise agility. Monolithic IT architectures are critical for firms when adjustments to processes are necessary in response to changing demands. However, high costs may be incurred when the organization wants to integrate a new strategy, for example [13]. Hence, to achieve enterprise agility, DevOps could be a suitable way of breaking down software monoliths into smaller services, where the responsibilities lie with one cross-functional team [18].

The relationship between software development and business strategy needs to be continuous. Literature labels this relationship as “BizDev” [8]. Existing research on information systems (IS), management, and software engineering calls for further investigation of this phenomenon [3, 8]. Continuous planning enables business and IT to work closely together in a “BizDevOps” environment [3, 14]. For enhancing the relations of business managers and IT employees, further research is necessary. The aim of this research is to determine *how an understanding of the relationship between customer demands and the DevOps approach can be achieved to enhance continuous innovations.*

We begin with a short introduction of the related literature and present the concepts of BizDevOps, enterprise agility, and continuous innovation. Then, we present our research method and describe our research approach. With the help of the case analysis, we present rich descriptions [15] of the results. Finally, we discuss our findings and conclude the paper.

2. Related Literature

2.1 The DevOps and BizDevOps Concept

To achieve a higher success rate of software development projects, a team should integrate skills and broad knowledge about the complete SDLC [4, 16]. For this, the DevOps approach could be a suitable solution, because project team members are responsible for the complete SDLC from planning to operations [17, 18]. DevOps is a new technological

trend that presents new challenges for organizations [19].

Business strategy and planning tasks provide challenges for the collaboration of business and IT. According to existing literature, the term BizDev implies the necessity for continuous integration and improvement between business strategy and software development. Hence, BizDev complements the DevOps concept [8, 20]. The importance of closer collaboration between business and IT arises from the short cycles of feedback from customers, which are implemented with the help of agile project management methods [21, 22]. Furthermore, more and more business employees act as proxies in the role of agile coaches or product owners (PO) in IT projects. To meet and satisfy customers demands, it is essential for the software engineering flow to have a tight connection between business, development and operations [8]. Continuous planning is a major capability for managing systems [3], as done by DevOps teams. Hence, combining BizDev and DevOps to form BizDevOps will foster the collaboration between business and IT. Integrating professional experts into DevOps teams is a key to achieving BizDevOps.

BizDevOps is defined as the integration of domain experts within DevOps teams. A major advantage is that the tighter connection between planning and execution leads to continuous planning [8]. Hence, customer demands can be satisfied faster and the team can react quickly to changing environments.

Organizations have to rapidly adapt to react to new customer demands [23] and build DevOps and BizDevOps capabilities in order to stay competitive [19]. The reasons are higher customer satisfaction with the provided software, as well as better software quality and higher project success [24]. A tighter collaboration between development and the operations part of an IT function is necessary to ensure that errors are quickly fixed and the quality and resilience of the software are enhanced. Nowadays, it is essential to develop innovative capabilities to react to digital disruptions [25].

In traditional IT functions, business managers are responsible for planning and prioritizing the processes. Furthermore, organizations centralize highly specialized IT staff in so-called silo units in order to build new software features using sequential development methods like “waterfall development.” Afterwards, there is a long time before the software features are implemented and run by the operations IT unit. The complete process has strong dependencies on the business manager [5, 26]. Through the DevOps concept, solutions are delivered

to avoid interruptions at different stages of planning, building, and running. Since the SDLC includes the steps these tasks, a tighter collaboration between planning, executing, and operating is enabled [3].

Using the DevOps concept, organizations are able to release new software features frequently and automatically [27]. Hence, risks linked to software releases can be reduced and feedback for new software features is received faster [28].

2.2 Enterprise Agility and Continuous Innovation

IS literature provides broad knowledge about enterprise and IT agility but lacks understanding of how a closer connection and flow between business and development and between business and operations can be achieved [13]. In times of uncertainty regarding planning processes in short cycle developments, agility concepts are necessary. A suitable use of IT is a key leverage factor for organizational agility [29]. Enterprise agility is defined as *“the ability of firms to sense environmental change and respond readily”* (Overby, Bharadwaj and Sambamurthy [13] p. 121). Literature highlights that a network based on trust and commitment with blurred boundaries is essential for a relationship between business and IT. A competitive advantage can be gained through better coordination, management, and structuring of relationships with stakeholders and a more agile-oriented collaboration with customers [30]. Agile software development methods could help to enhance this relationship. IT projects with short time system development enable faster delivery of innovations to the customers [29]. Existing literature states that the combining of business and technology alignment can be achieved and can support the business cycle, deliver major benefits, and provide innovations [1].

Continuous innovation is defined as a sustainable process that supports responsiveness to new requirements and changing market demands throughout the SDLC [8, 31]. In the business context, innovations are combined with new ideas, which are transformed to achieve value for business. Continuous innovation is most widely used in the area of software development through concepts like DevOps. Thereby, early customer feedback to new software deployments can be obtained [8]. Furthermore, planning is a key prerequisite for continuous innovation. Adequate planning processes

are very important for avoiding failures in the development processes. Continuous innovation helps processes to react to new market demands across the entire SDLC of planning, building, and running software [3, 8]. The BizDev approach recognizes this issue and tries to tighten the relationships between business strategies and software development. The PO is responsible for the business contact. This is emphasized by agile software development methods like scrum as the first step toward the BizDev direction.

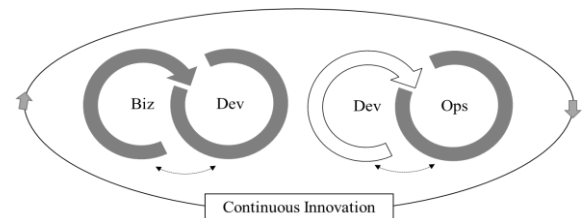


Figure 1. BizDevOps and continuous innovation

3. Research Design

We conducted a multiple-case study to analyze the flow between business stakeholder demands, software developers, and operations. Since BizDev and DevOps studies are neglected in existing literature, our aim is to provide rich descriptions with the help of grounded theory through a multiple-case study research [15, 32]. In this section, we describe our exploratory research design and approach.

The cases considered in our study have their headquarters in Germany. A case study approach is defined as *“an empirical inquiry that investigates a contemporary phenomenon in depth and within its real-life context”* (Yin [33] p. 18). The present paper is among the first studies to investigate the BizDevOps phenomenon [33]. The advantage of case study research is that it can examine real-life situations and test or develop theoretical perspectives in relation to the considered phenomena as they unfold in practice [34]. In summary, case studies are an appropriate method to improve our understanding of BizDevOps teams and to show how relationships between planning and development and between operations processes of the SDLC are implemented.

Table 1. Primary and secondary data

Primary Data		
No. of Interviews	Role of Interviewee	No. of Interviews per Role
23 (some interviews were held with more than one interviewee)	CTO	1
	IT Manager	15
	Product Owner	3
	Team Member	9
	Sum	28
Over 400 pages of transcriptions	The interviewees were mainly conducted personally through face-to-face interviews. Some interviews were held via telephone. The research team took notes regarding observations during the interviews. In total, more than 400 pages of transcriptions and memos per interview were created between the end of 2016 and February 2018.	
Secondary Data		
Webpages, blog articles, and white papers	We searched through the internet and collected information about the companies. Often, the companies have blogs where they publish information about collaboration.	

We conducted an exploratory case study to answer our research question. Case studies offer a great variety of techniques for data collection [33] for the DevOps teams. Their characteristics and effects on the firms are primarily studied through expert interviews. An expert is someone who has privileged and deep knowledge about a special topic [35]. Here, the experts have privileged knowledge about DevOps teams and the customer view in their organization and can assess their characteristics and outcomes. We tried to talk to people in different roles and responsibilities to find out how planning processes and customer view (Biz) are implemented and how the DevOps concept fosters continuous innovation.

During our research, we used qualitative data-coding processes for the interpretation of our data [32, 36]. Furthermore, we followed the guidelines of grounded theory for data collection and analysis, as described by Wiesche et al. [15].

To participate in our study, a precondition was that the teams had to be familiar with the DevOps concept and must already have integrated planning, development, and operations processes. Additionally, the teams should have integrated an agile method (e.g. Scrum or Kanban) to collaborate with customers. We conducted a multiple-case study to

analyze the relationships of business strategies with software development and operations. In short, we talked to 28 interview partners from 15 companies. Table 1 provides information about primary and secondary data.

A semi-structured interview was conducted with each participant, supported by guidelines and a list of questions or general topics that the interviewers wanted to touch upon [33]. The questions were mainly open-ended, giving the interviewees the possibility to explore their experience and views [33].

The interview guidelines helped to keep the interaction focused as data collection proceeded. It ensured comparability of data across individuals, settings, and researchers [37]. Although the interview process was systematic and comprehensive, the interviewer had a high degree of freedom to probe and explore these guidelines. Thus, questions were adjusted during the interviews to gain more in-depth knowledge for each case.

Each interview lasted about 45–75 minutes and was conducted through face-to-face meetings or by telephone. The interviews were held in German or English. German statements were translated into English for further analysis. Every interview was recorded and transcribed. Moreover, a lot of notes were taken during the interview.

Table 2. Coding process and core categories

Dimension	Definition	Statements (examples)
Team side	The responsible person for planning the backlog, integrated within the DevOps team.	<i>“From the developer's point of view, I'm very happy that the PO <u>now sits next to me and I can have a constant exchange with him</u>” (Team member).</i> <i>“We have a lot of <u>cross-functional teams. That means you have a team with product manager, developers, and OA</u>” (CTO).</i>
Customer side	The responsible person for planning, integrated at the customer/ business side	<i>“We <u>work together relatively closely with PMs from other company parts, although they do not sit with us</u>” (IT manager).</i> <i>„They are also [organizationally] <u>close to us in the holding company. [...] they call themselves business class</u>” (Team member).</i>
Team lead	The responsible person of the planning process.	<i>“I'm in <u>the team in some roles, i.e. in this product team as Product Owner. At first, I also did a lot of development by myself, but everything else the teams do, they just have to vote with the customer</u>” (PO/CTO).</i>

During our data analysis, we wanted to examine the relationships and concepts between business planning and software development [8]. The related literature presented in Chapter 2 was helpful for guiding our examination. For the coding process, we followed the guidelines approach presented by existing literature [15, 36] and used the software NVivo10. During the coding, the research team took notes to justify the coding process. Afterwards, we identified subcategories in the planning and collaboration processes with the help of axial coding. Finally, with the help of selective coding, we related the categories to mechanisms describing the effect of collaboration between BizDev and DevOps.

4. Findings

We started with an open coding process to identify core categories of relationships between BizDev and DevOps. The categories explain the process of planning in software development projects and different forms of customer integration into the SDLC. Table 2 presents our findings regarding the open-coding process. These findings confirm the results of existing research about integrating the customer view in software development projects with the help of a PO [38].

Table 3. Areas of planning relevant to DevOps teams

Area of Planning	Definition	Code Selection
Responsibility	Planning responsibility means that the team is now responsible and incorporated within the planning process and includes the impact of development and operations.	<ul style="list-style-type: none"> • Writing requirements • Service responsibility • Requirements implementation • Common understanding
Scope	Planning scope is defined as the size and extent of planned components that should be implemented by the team into the software in short iterations.	<ul style="list-style-type: none"> • Agile development meetings • Communication • Understanding of the software components and customer needs • Small increments
Dependency	Planning dependency is defined as the relationship of planning processes that are now integrated within the team with project success and running the software successfully.	<ul style="list-style-type: none"> • Collaboration between planning, building, and running • Team autonomy/flat hierarchies • Respect among team members • Consequences of failures • Shared tasks and understanding

4.1 Areas of Planning as Subcategories

We present different ways to integrate the customer perspective in the team. On examining the three possibilities of collaboration between business planning and software development and operations, we realized that planning is related to different processes in the SDLC and is connected to BizDev and DevOps activities.

Our findings confirm that the integration of a PO within a DevOps setting is important for achieving continuous innovation. The collected data show that apart from the establishment of the PO in the DevOps team, there are different areas of planning, as described in Table 3. Our interviewees stated that the responsibility for the tasks of planning, developing, and running the software is now integrated into one cross-functional IT project team.

“I know that the Product Owners, who came to us, were already relatively IT-savvy and partly from IT. They have taken a different career path and spent some time in marketing. [...] That brought a lot of responsibility into the team” (Team member).

Furthermore, the scope of planning changed to a high degree. In traditional software development projects, the customers only have the possibility to plan their requirements for very long release cycles. Hence, business people cannot introduce a new demand into the development cycle because the planning phase is already closed and they have to wait a long time for adding new demands to the next big release [8]. This problem can be avoided through the implementation of continuous planning with the help of the BizDevOps approach, because introduction of new ideas and requirements is possible at all times.

“So, what distinguishes our team is mainly the Product Owner. A good planning and coordination with the Product Line Management specify the requirements. A really good planning with user stories and not just reacting to requirements but working proactively. [...] We have always really attempted to show a minimal product finished in two weeks—a small increment that we were able to present” (IT manager).

Finally, we identified the area of planning dependencies as related to different processes in DevOps teams. Through the integration of a high degree of autonomy, the team is responsible for the planning process, as mentioned before, and must also be aware of the dependencies for the project success. Since the planning processes are now integrated in the team, the successful software delivery is in the hands of the BizDevOps team.

“Ultimately, the responsibility for the applications lies within the team, and that means—now that the Product Owner is also in the team—actually everything from writing the requirements to development to operation” (Team member).

We listed a row of areas of planning for BizDevOps teams. The introduction of these terms is necessary for describing the process of achieving continuous innovation in software development projects. These terms are dependent on the three core categories identified in Table 3. DevOps teams and business have already implemented planning configurations for achieving continuous innovations.

4.2 Mechanisms for Continuous Innovation

As mentioned before, a key prerequisite for continuous innovation is planning. Fitzgerald and Stol [6] state that innovation in business areas is connected to business value for the service recipient. Continuous innovation tries to enable processes that help to react to new market conditions and are related to metrics across the SDLC. Table 4 presents the mechanisms for meeting continuous innovations related to planning processes, identified with the help of our data. The following table depicts the mechanisms and the related area of planning with the key challenges that appear in our data.

For achieving continuous innovation, our cases initially implemented some mechanisms, which we identified as scalability, security, and quality. Innovation could be gained through scalable services. Scalability fosters speed to easily broaden the resources. The BizDevOps team is now able to do a lot of tasks by itself, because of responsibility for example, and hence to enhance scalability and speed of the service.

“Suddenly, we wanted to scale and that was not so easy in the old structure. With the scaling comes the fact that you want to bring things faster to the customers [...] e.g., during Christmas time” (Team member).

The data presents insights that combine planning processes in DevOps teams; a higher level of unique selling points can be achieved through planning. The team is able to plan its demands and efforts to be taken in case of problems; thus, the team has a great overview.

“Where we want to distinguish ourselves from competitors, which means where we have a higher level of competition there. We also want to have a higher level of agility, in other words, we specifically selected this DevOps-oriented approach because if we want the highest possible speed, then the team should be cross-functional” (CTO).

However, to achieve scalability, a tight exchange of planned requirements is necessary. Furthermore, one interview partner mentioned that they still lost speed because the business department wanted a manual acceptance test: *“Hence, we have to wait for implementation, and I am angry about that”* (Team member).

We identified security to foster continuous innovations as the next mechanism. This is related to the responsibility and scope that are now integrated in the teams. Planning responsibility delivers a feeling of safety to the team; the team is no longer concerned that new requirements would be introduced by external people because they are involved in the planning processes.

“Teams have a higher flexibility; they are more autonomous. They have a higher degree of safety regarding planning” (IT manager).

Security is important during the entire SDLC, and some team leads still make great efforts to *“claim operations responsibility”* (IT manager), because possible failures in the running systems need planning efforts as well.

The third mechanism that we identified is quality.

Our findings indicate that a stable running software, where changes are possible when necessary, fosters innovations. A high-quality planning process avoids failures in the implementation and running phase of the software. BizDevOps teams need the skills and awareness to deliver the complete SDLC.

“The team and its members have very a high level of personal maturity. They have high claims to themselves and to the quality of work. They have a very high degree of customer-oriented thinking” (IT manager).

However, for achieving high stability for a system, planning quality and dependent factors should be considered. *“We want the team to work self-responsibly and decide when things go live and they have to take over the complete responsibility for quality and operations”* (CTO).

5. Discussion

In our study, we present rich descriptions of the combination of planning areas and mechanisms for achieving continuous innovations through BizDevOps teams, as presented in Figure 2. We

Table 4. Mechanisms for continuous innovation and planning relation

Mechanisms	Definition	Manifestation
Scalability	BizDevOps teams want to achieve the highest agility and speed to stand out from competitors through integrated planning and solving problems during run time operations and scaling the software to a broader level if necessary.	<ul style="list-style-type: none"> • Planning of necessary proportional increase of service resources • Claim against competitors • Enhancement of speed through fast reactions and planning autonomy
<i>Related to</i>	Planning responsibility, scope, and dependencies	<ul style="list-style-type: none"> • Need for quicker communication • Avoiding of manual acceptance test by PO
Security	BizDevOps teams are responsible for planning and hence want to achieve high planning security, because the team defines the priority of the planned increments of the software if failures appear in the running software.	<ul style="list-style-type: none"> • Scope and autonomy lead to planning security • Responsibility for services and consideration in planning processes
<i>Related to</i>	Planning responsibility	<ul style="list-style-type: none"> • Need for communication within the team • Call for taking over complete service responsibility
Quality	BizDevOps teams are responsible for delivering and running the software. The team members need to be aware that they are responsible if problems appear. Proactive avoidance of failures is enhanced through a high-quality planning processes for the product.	<ul style="list-style-type: none"> • Product quality depends on planning, developing, and running tasks • Team members develop awareness of product quality
<i>Related to</i>	Planning scope and dependencies	<ul style="list-style-type: none"> • Need for accurate planning of the backlog • Call for awareness of failures

identified areas in BizDevOps and planning processes that trigger mechanisms for achieving innovations for customers, as shown in Tables 3 and 4.

We give insights into how the phenomenon of BizDevOps can be arranged with the related planning processes to understand how continuous innovation can be achieved. This is different from the findings of existing literature, where the need for further investigation of continuous planning and innovation and DevOps capabilities is highlighted [8, 19]. Thus, the grounded theory approach of achieving continuous innovation through DevOps teams developed in this paper is the first attempt to explain of how planning processes are integrated and how the BizDevOps concept can be achieved.

We propose that this continuous innovation is correlated with the planning of customer requirements. Furthermore, we find that the evolution of continuous innovation is a process that is combined with planning areas. There are some challenges described by our interviewees that need to be considered.

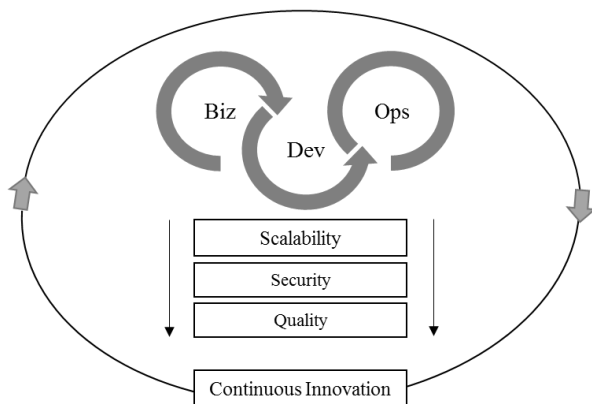


Figure 2. Rich descriptions of achieving continuous innovation through BizDevOps

The first mechanism is scalability. Scalability depends on the planning responsibility of BizDevOps teams. The team members have to feel responsible for the service. If the scaling of the service is necessary due to some peaks, it must be recognized proactively by the team members and accurately planned and realized. Hence, it involves planning responsibility and the corresponding steps. Furthermore, to achieve the necessary speed for scaling, planning dependencies are important as well. The awareness of collaboration and the dependencies of scaling for project success are important points that should be considered.

The second mechanism is security. Security is related to planning responsibility. The BizDevOps team is responsible for planning and operating the system. Hence, there is a need for accurate planning and the awareness must be fostered in the team.

The third mechanism is quality. Quality is dependent on planning scope and dependencies. Within BizDevOps teams, a new culture of collaboration is necessary. The team members need to have the attitude that the service is owned by them. The members have to decide which components are to be developed in each iteration. Therefore, high quality in the planning process is necessary, so that less failures appears.

Additionally, we identified three dimensions of integrating the customer view (Biz) into the DevOps team. Our data indicate that the PO is in the DevOps team, on the customer side, or the team lead. We found evidence that if the PO is integrated within the DevOps team, the best collaboration and planning processes are achieved. However, our findings also indicate that if the PO is settled on the business side, a high degree of exchange and close collaboration with the business are achieved. The third dimension is when the PO is the team lead. This setting appears because of the transformation of traditional IT functions to a cross-functional BizDevOps team setting. Our data highlight that middle management positions could break away, leaving the company with a “*social responsibility*” (Team lead) against managers and the PO position is handed over to them.

5.1 Limitations and Future Research

As mentioned in Chapter 3, we mainly talked to IT managers and other IT people. This is because some of our cases lack even a typical business department. Two cases mainly consist of IT departments and some support units. Further research should include cases that have a traditional business department which is involved in the planning processes. Generalization of this study is limited by a case-study approach; hence, validity is limited to our findings. Other studies in different settings might complement our examination. We present insights into how the planning process could be integrated in a DevOps teams with BizDevOps. We present mechanisms for continuous innovation and planning relation, but this needs further enhancements. One way to achieve some kind of evaluation is by conducting a quantitative study. Further research is needed for the replications of our findings across other settings [33]. Additionally, future examination is necessary to identify other continuous innovations mechanisms

that may be important for BizDevOps settings. Furthermore, we focus on intra-organizational collaboration of customers and IT functions. Other settings with inter-organizational, e.g. outsourcing, settings for development and operations tasks may provide new insights as well.

6. Conclusion

Our paper presents insights into continuous innovation steps that could be provided by BizDevOps teams by integrating the planning of customer demands. This is one of the first studies to investigate how planning processes and the customers view can be integrated in a DevOps team—i.e. integrated business processes. With the help of grounded theory, we derived rich descriptions for achieving continuous innovations through the concept of BizDevOps. Thus, we contribute to existing research [8, 27] and provide deeper insights into how the collaboration of business and IT functions could be implemented through new approaches like DevOps teams. Based on our explorative case study, we identified scalability, security, and quality as mechanisms for continuous innovation. Scalability includes the planning processes in case of an increase in service resources, accentuation against competitors, and the enhancement of speed through the possibility of fast reactions. Security in BizDevOps teams refers to the scope and autonomy that leads to planning security and the responsibility for the service and consideration in planning processes. Quality means that the product is dependent on planning, developing, and running tasks that are conducted by the BizDevOps teams and that team members need to develop an awareness of product quality. For practice, we present guidelines for closer collaboration and integration of planning processes and short cycle times in software development projects.

7. Acknowledgement

We thank the German Federal Ministry of Education and Research (BMBF) for funding this research (grant code 03FH005PX4).

8. References

[1] Holmqvist, M., and Pessi, K., "Agility through Scenario Development and Continuous Implementation: A Global Aftermarket Logistics

Case", *European Journal of Information Systems*, 15(2), 2006, pp. 146-158.

[2] Lowry, P.B., and Wilson, D., "Creating Agile Organizations through IT: The Influence of Internal IT Service Perceptions on IT Service Quality and IT Agility", *The Journal of Strategic Information Systems*, 25(3), 2016, pp. 211-226.

[3] Fitzgerald, B., and Stol, K.-J., "Continuous Software Engineering and Beyond: Trends and Challenges", in: *International Workshop on Rapid Continuous Software Engineering*, ACM, 2014.

[4] Wiedemann, A., and Wiesche, M., "Are You Ready for Devops? Required Skill Set for Devops Teams", *European Conference on Information Systems*, 2018.

[5] Markus, M.L., and Keil, M., "If We Build It, They Will Come: Designing Information Systems That People Want to Use", *Sloan Management Review*, 35(4), 1994, pp. 11-35.

[6] Recker, J., Rosemann, M., Indulska, M., and Green, P., "Business Process Modeling—a Comparative Analysis", *Journal of the Association for Information Systems*, 10(4), 2009, pp. 333-363.

[7] Wiesche, M., and Krcmar, H., "The Relationship of Personality Models and Development Tasks in Software Engineering", *ACM SIGMIS-CPR*, 2014, pp. 149-161.

[8] Fitzgerald, B., and Stol, K.-J., "Continuous Software Engineering: A Roadmap and Agenda", *Journal of Systems and Software*, 123, 2017, pp. 176–189.

[9] Lehtola, L., Kauppinen, M., Vähäniitty, J., and Komssi, M., "Linking Business and Requirements Engineering: Is Solution Planning a Missing Activity in Software Product Companies?", *Requirements Engineering*, 14(2), 2009, pp. 113-128.

[10] Chen, L., "Continuous Delivery: Overcoming Adoption Challenges", *The Journal of Systems and Software*, 128, 2017, pp. 72–86.

[11] Knight, S., Rabideau, G., Chien, S., Engelhardt, B., and Sherwood, R., "Casper: Space Exploration through Continuous Planning", *IEEE Intelligent Systems*, 16(5), 2001, pp. 70-75.

[12] Pflügler, C., Wiesche, M., and Krcmar, H., "Subgroups in Agile and Traditional IT Project Teams", *Hawaii International Conference on System Sciences*, 2018.

[13] Overby, E., Bharadwaj, A., and Sambamurthy, V., "Enterprise Agility and the Enabling Role of Information Technology", *European Journal of Information Systems*, 15(2), 2006, pp. 120-131.

[14] Silverthorne, V., "Business Process Management Aids Application Development", in: *Application Development Handbook*, TechTarget, E-Handbook, 2017.

- [15] Wiesche, M., Jurisch, M.C., Yetton, P.W., and Krcmar, H., "Grounded Theory Methodology in Information Systems Research", *MIS Quarterly*, 41(3), 2017, pp. 685-701.
- [16] Peppard, J., "Rethinking the Concept of the IS Organization", *Information Systems Journal*, 28(1), 2018, pp. 76-103.
- [17] Horlach, B., Drews, P., Schirmer, I., and Boehmann, T., "Increasing the Agility of IT Delivery: Five Types of Bimodal IT Organization", *Hawaii International Conference on System Sciences*, 2017.
- [18] Wiedemann, A., and Wiesche, M., "How to Implement Clan Control in DevOps Teams", *Americas Conference on Information Systems*, 2018
- [19] Sebastian, I.M., Ross, J.W., Beath, C., Mocker, M., Moloney, K.G., and Fonstad, N.O., "How Big Old Companies Navigate Digital Transformation", *MISQ Executive*, 16(3), 2017, pp. 197-213.
- [20] Debois, P.A., "Opening Statement", *Cutter IT Journal*, 24(8), 2011, pp. 3-5.
- [21] Przybilla, L., Wiesche, M., and Krcmar, H., "The Influence of Agile Practices on Performance in Software Engineering Teams: A Subgroup Perspective", *ACM SIGMIS-CPR*, 2018.
- [22] Lassak, S., Przybilla, L., Wiesche, M., and Krcmar, H., "Explaining How Agile Software Development Practices Moderate the Negative Effects of Faultlines in Teams", *Australasian Conference on Information Systems*, 2017.
- [23] Sambamurthy, V., Bharadwaj, A., and Grover, V., "Shaping Agility through Digital Options: Reconceptualizing the Role of Information Technology in Contemporary Firms", *MIS Quarterly*, 27(2), 2003, pp. 237-263.
- [24] Tripp, J.F., Riemenschneider, C., and Thatcher, J.B., "Job Satisfaction in Agile Development Teams: Agile Development as Work Redesign", *Journal of the Association for Information Systems*, 17(4), 2016, pp. 267 – 307.
- [25] Châlons, C., and Dufft, N., "The Role of IT as an Enabler of Digital Transformation", in: *The Drivers of Digital Transformation*, Springer, Saarbrücken, Germany 2017, pp. 13-22.
- [26] Brown, C.V., and Ross, J.W., "The Information Systems Balancing Act: Building Partnerships and Infrastructure", *Information Technology & People*, 9(1), 1996, pp. 49-62.
- [27] Ross, J.W., Sebastian, I., Beath, C., Mocker, M., Moloney, K., and Fonstad, N., "Designing and Executing Digital Strategies", *International Conference on Information Systems*, 2016.
- [28] Lwakatare, L.E., Karvonen, T., Sauvola, T., Kuvaja, P., Olsson, H.H., Bosch, J., and Oivo, M., "Towards DevOps in the Embedded Systems Domain: Why Is It So Hard?", *Hawaii International Conference on System Sciences* 2016.
- [29] Baskerville, R., and Pries-Heje, J., "Short Cycle Time Systems Development", *Information Systems Journal*, 14(3), 2004, pp. 237-264.
- [30] Christopher, M., and Towill, D.R., "Supply Chain Migration from Lean and Functional to Agile and Customised", *Supply Chain Management: An International Journal*, 5(4), 2000, pp. 206-213.
- [31] Ries, E., *The Lean Startup: How Today's Entrepreneurs Use Continuous Innovation to Create Radically Successful Businesses*, Crown Business, New York 2011.
- [32] Urquhart, C., Lehmann, H., and Myers, M.D., "Putting the 'Theory' Back into Grounded Theory: Guidelines for Grounded Theory Studies in Information Systems", *Information Systems Journal*, 20(4), 2010, pp. 357-381.
- [33] Yin, R.K., *Case Study Research: Design and Methods*, Sage Publications, Thousand Oaks, CA, 2014.
- [34] Flyvbjerg, B., "Five Misunderstandings About Case-Study Research", *Qualitative Inquiry*, 12(2), 2006, pp. 219-245.
- [35] Bogner, A., Littig, B., and Menz, W., "Introduction: Expert Interviews—an Introduction to a New Methodological Debate", in: *Interviewing Experts*, Springer, 2009, pp. 1-13.
- [36] Glaser, B.G., and Strauss, A.L., *Grounded Theory: Strategien Qualitativer Forschung*, Huber, Bern, 1998.
- [37] Maxwell, J.A., *Qualitative Research Design: An Interactive Approach*, Sage Publications, Thousand Oaks, 2012.
- [38] Pries-Heje, L., and Pries-Heje, J., "Agile & Distributed Project Management: A Case Study Revealing Why Scrum Is Useful", *European Conference on Information Systems*, 2011.