“What’s Going On” with BizDevOps: A Qualitative Review of BizDevOps Practice

Pedro Antunes[1], and Mary Tate[2]

[1]Faculty of Sciences, University of Lisbon
Campo Grande, 1749-016 Lisbon, Portugal
[2]School of Information Management, Victoria University of Wellington
PO Box 600, Wellington 6140, New Zealand

BizDevOps is an emerging trend that seeks to cut back the lag between product/service vision and implementation. However, so far this trend has been mainly unnoticed by research. This paper carries out a “grey literature” (non-academic) review on BizDevOps. Data is collected from reports, articles, webpages, and blog posts to capture the professionals’ insights on BizDevOps. We develop a conceptual framework for BizDevOps that organizes and integrates concepts and constructs embedded in the grey literature. Based on this, the paper offers insights for organizations aiming to move towards the BizDevOps approach and identifies research opportunities in the BizDevOps area.

Keywords: BizDevOps, DevOps 2.0, BizDevOps framework, grey literature review

1 Introduction

The organization’s ability to respond rapidly to changing technical and business requirements has been increasing steadily since Agile (first) and DevOps (next) approaches helped expedite software development [1,2]. BizDevOps accelerates this trend even further. It is a set of practices, tools, and a cultural philosophy that automates and integrates the processes between business development (Biz), software development (Dev), and operations teams (Ops) [3,4].

BizDevOps helps teams reach for the “holy grail” for decades in many organizations: the ability to develop and implement new and changed IT-based products and services, with no lag between vision and implementation, as soon as the organization thinks of them, users raise new demands, or the business environment changes. To illustrate the opportunities, imagine a business manager for a bank who would like to swiftly introduce a new offer for a credit product. After some quick brainstorming with a BizDevOps team about which features could make the product both popular and profitable, the team is able to launch the product within hours in the bank’s platform, including “push” marketing to selected customers, getting immediate feedback about its effectiveness. This is achievable with BizDevOps.

However, prior empirical research on BizDevOps is still scarce, mainly focusing on early conceptualizations from industry addressing the integration of business and IT through automated toolchains and various team-oriented collaborative practices [5,5,6]. A more holistic and comprehensive understanding of the implementation of BizDevOps by organizations is still missing.

This study explicitly aims to take a snapshot of the current practitioner discourse about BizDevOps through an analysis of “grey literature” sources, i.e., materials and research produced outside academic publishing. We next describe the motivations for this approach and how it differs from prior studies.

Antunes, P., and Tate, M. 2024. “‘What’s Going On’ with BizDevOps: A Qualitative Review of BizDevOps Practice.” Computers in Industry (to appear). ISSN: 0166-3615.

1.1 BizDevOps background in research and practice

In this section, we describe the background to our current understanding of BizDevOps in research and practice. We also briefly present the results of the literature search that was carried out continuously during the research to contextualize our emerging findings.

The BizDevOps approach has its origins in practice, in particular in software development circles that were already engaged in DevOps [5,7,8]. Processes, practices, and tools established by DevOps teams can be extended to cover the Biz side of software development, including, for example, close collaboration with the stakeholders involved in requirements definition and product/service design, quick development, integration, and testing of specific features based on automation, and adoption of collaboration and knowledge sharing tools [9].

A popular conceptualization is shown in Figure 1. Business specialists are responsible for defining requirements, ensuring alignment with strategy, and approving changes. Software developers plan, create (develop), verify (test), and package (organize for release) tranches of functionality endorsed by business specialists. Operations specialists release software into production, ensure it is correctly configured, and monitor operational performance. This occurs as an integrated and seamless process, often conducted by one cohesive team.

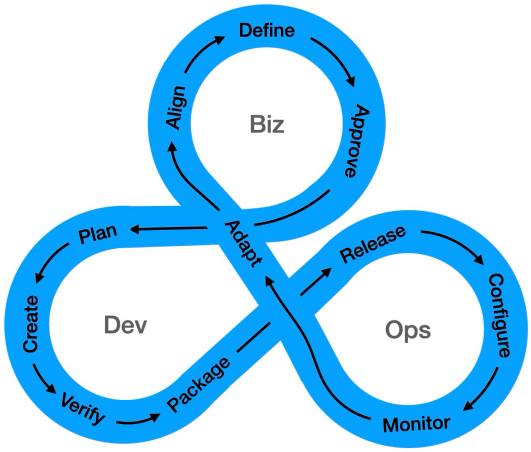

Figure 1: A popular conceptualization of BizDevOps [10]

From a research perspective, there have been relatively few academic studies that we were able to identify that examine broad trends in BizDevOps. Some significant exceptions include a multi-vocal (academic and grey literature) analysis from Lohrasbinasab et al. [4], a review of BizDevOps applications and platforms [8], and an in-depth discussion of the use of BizDevOps for continuous innovation [9]. These focused studies uncover a number of trends, including identifying the importance of tools, IT support, performance indicators, cultural issues, and breaking silos between business and IT [4,8,9] but fall short of offering a holistic and integrated framework that can be used to initiate research conversations about the BizDevOps, and guide implementation efforts.

There have also been useful studies that focus on specific aspects of BizDevOps, such as requirements engineering (e.g., [7]) and modeling (e.g., [11]), but these also do not offer the integrated framework we are seeking.

1.2 Sociotechnical perspective on BizDevOps

While we did not explicitly begin with a sociotechnical frame for our analysis, and our analysis does not fall “neatly” into the six dimensions of sociotechnical systems (namely: people,

infrastructure, technology, culture, processes, and goals), it is clear that BizDevOps implementation needs to be considered from a sociotechnical perspective. Integrating technological and organizational views has been found to be essential to understanding contemporary technology challenges [12]. The central role of effective management of sociotechnical change emerged as a key theme from our analysis. To contextualize this, we briefly review the role of change management in understanding the BizDevOps discourse. Using a broad research lens, BizDevOps can be seen as part of the accelerating pace of change. The trend leading from Agile to DevOps and BizDevOps underlines the recognition that organizations need to stay abreast of IT trends, including the scale of change, the pace of change, and the effects of interactions and combinations between multiple, rapidly evolving technologies [9]. These combinations create new business value propositions and affordances, disrupting many industries [13]. It has been argued that the pace of the pace of change itself is accelerating. "There's even exponential growth in the rate of exponential growth" [14].

More specifically, we can view BizDevOps adoption as a specific case of sociotechnical change. "Traditional" models of change (e.g., [15]) tend to focus on a "punctuated equilibrium" view of change, where a steady state is "unfrozen," change occurs, and a new normal is re-established [16]. However, other change models recognize the more networked, continuous, and interconnected nature of change. The McKinsey 7-s model conceptualizes change management as a network of strategy, skills, values, project and organizational structures, systems, values, staff, and leadership style [17].

Investigating rapidly changing sociotechnical phenomena poses challenges for empirical researchers. The issues and advantages gained by implementing one approach in a specific context may have changed considerably by the time empirical research has been carried out and passed through long peer review and publication pipelines. Therefore, it is a risk that knowledge based on empirical studies will date rapidly. The holistic, continuous nature of change associated with technologies such as BizDevOps challenges reductionist approaches to research and requires a broad purview. Consequently, it is essential that researchers critically evaluate the potential of emerging approaches, such as BizDevOps, which have yet to become mainstream in empirical research holistically. In this study, we address these concerns by focusing on grey literature.

1.3 Focus on grey literature in BizDevOps research

The focus on grey literature stems from a growing recognition that practitioner discourse about technologies can provide a useful source of data to yield theoretical insights. For example, the discourse carried out on social media has been used to study practitioner sense-making about new technologies [18], and grey literature has been increasingly incorporated into reviews on applied topics such as software engineering [19]. Grey literature reviews are ideal for capturing a point in the real-time process of sensemaking about new technology and identifying practitioners' current and emerging preoccupations while recognizing this is a constantly moving landscape.

A point-in-time study is appropriate because it is well-established that practitioner discourses about new and innovative technologies go through similar life cycles. These have been described in various ways but have important things in common. Probably one of the best-known is the Gartner "hype cycle" [20]. This posits that the discourse about a new technology goes through five stages. First, the "innovation trigger" creates interest. Next, the "peak of inflated expectations" reports startling success stories and sometimes some failures as organizations begin to adopt the technology. The "trough of disillusionment" occurs following this, as some experiments with the technology fail to deliver. After this low point, the "slope of enlightenment" occurs. Understanding of the benefits of the technology becomes more widespread. Finally, technology has become

accepted as part of the mainstream in the "plateau of productivity" [20]. Currently, BizDevOps is still in the first stage [5,21].

A less well-known explanation of the trajectory of discourse about technology innovation is provided by organizing vision theory. Organizing visions describe a cycle of interpretation, legitimation, and mobilization of new technologies [22]. Interpretation clarifies what the innovation is. Legitimation seeks to answer the question "Why do it?" and "Who is doing it?" while mobilization considers "How to do it?" This theory has been applied effectively to analyzing the arc of discourse on social media about new technologies such as blockchain [23]. Currently, BizDevOps still lacks an organizing vision [24].

Both conceptualizations recognize that there are story arcs involved in sensemaking by the practitioner community about the potential of new technology as they work through understanding the characteristics and potential of the technology itself, the motivations for adoption, the value propositions, the competitive landscape, and implementation strategies. These story arcs evolve over time, so any study drawing on data from practitioner discourse will always represent a snapshot in time, with the expectation that both the data sources and the focus of the discourse will change. Nevertheless, given the time taken in academic pipelines and the rapid developments of innovative technologies like BizDevOps, a snapshot of practitioner discourse on that topic will yield a more contemporary and better-grounded set of research challenges associated with BizDevOps than any study restricted to peer-reviewed academic sources [4].

1.4 Study aims

In this paper, we develop a conceptual framework for BizDevOps implementation by organizations. The focus is on how organizations can embrace the BizDevOps approach through a meaningful story arc. The framework is developed from knowledge elements extracted from a literature analysis of grey literature on BizDevOps. The conceptual framework gives "theoretical sensitivity," i.e., coherent narrative setting and structure [25], to concepts and constructs embedded in the grey literature. Based on this, we offer insights for organizations aiming to move towards a BizDevOps approach. We also identify research opportunities in the BizDevOps area.

The rest of this paper is structured as follows. We first discuss the research design. We then present the content analysis of the reviewed data, culminating in an integrated conceptual framework. We then offer a discussion and some concluding remarks.

2 Research Design

The study adheres to a grounded qualitative research approach [26]. This approach, codified by Wolfswinklel et al. [27], has become widely adopted for conducted literature analysis on technology-related topics, and builds on previous advice that argues for "concept-centric" reviews [28]. This approach is suited for creating depth and breadth and allowing key concepts to surface, rather than being deductively derived beforehand, so that "new ideas emerge" [27].

A qualitative research approach is adequate for this study for several reasons. One reason is that the study aligns with the exploratory nature of qualitative research [29]. In this study, we seek to explore what is going on with BizDevOps. Another reason is that qualitative research seeks to describe the world as it is perceived by different observers [29], paying attention to nuances and embedded meaning [30]. In this study, we focus on practitioners' commentaries about BizDevOps. We aim to characterize the state of practitioner discourse rather than investigate pre-defined constructs. Finally, this approach illuminates holistic accounts of collected data [26,30]. In this study, we elaborate a holistic account of BizDevOps, with a particular concern for BizDevOps capabilities and effects.

Researchers adopting a grounded approach collect data themselves through a variety of procedures. They review all data, make sense of it, and organize it into categories and themes from the bottom up, moving from raw data towards increasingly abstract units of information [27,31,32]. The research design follows the process suggested by Wolfswinkel et al. [27]: define the problem, search, select (including refining the sample), code iteratively, and present the results. Next, we further detail the process.

2.1 Define the problem

This step involves identifying the problem and area of research, the criteria for inclusion, appropriate sources, and deciding on initial search terms. As we discussed briefly earlier, the problem is to take a snapshot of the sensemaking that is going on in the practitioner discourse about BizDevOps. Criteria for inclusion were relatively broad and included individuals and forums where relevant discourse is occurring, provided the focus remains clearly on the BizDevOps phenomenon.

The choice of appropriate sources was decided collaboratively by the research team. At a high level, as we have discussed previously, grey literature (non-academic) sources were selected [19]. Such reviews allow researchers to "understand the practices and views of industry" and to "explore uncharted research areas" [33]. Recognizing that, generally, the grey literature evolves faster than the peer-reviewed literature, a grey literature review is valuable for providing insights into state-of-the-art approaches [33]. Grey literature is adequate for this study because BizDevOps is an emerging phenomenon driven by practice.

More specifically, grey literature reviews depend on ephemeral resources, coming in various forms, which can be hard to look up [34]. This study aims to capture the current practitioner experience "as it is happening in real-time," so we selected second and third-tier sources (with moderate and low retrievability) [34] available online as professional reports, articles, webpages, and blog posts. We excluded forum posts from the selection, as they can be highly ephemeral and challenging to identify and associate with expertise and experience.

Two procedures were adopted to control the selection of sources. We checked the availability of online documents in two different periods, separated by four months. We also stored the documents offline to ensure durability and transparency.

Keyword selection is usually critical in literature reviews, as it contributes to rigor, thoroughness, and transparency [35]. The selection of keywords in this study was eased by the strict focus on BizDevOps and the perceived consistent use of this term in the grey literature. Nevertheless, after preliminary analysis of the collected data, two other keywords were selected to increase thoroughness: "Biz Dev Ops" and "DevOps 2.0."

2.2 Search

Searches were done using two common search engines, Google and DuckDuckGo. We used two engines to avoid engine bias. The adoption of common search engines stems from the focus on grey literature, as they provide more coverage of second and third-tier sources than specialized search engines like Scopus and Web of Science. The search process was iterative.

2.3 Select

The search results were screened to check for document type and contents. Only professional reports, articles, web pages, and blog posts were considered. Only contents related to technology development were taken, excluding results related to publicity, consulting propositions, job offers, and brief commentary.

2.4 Analyze and code

Each document was analyzed in detail and specific data items were extracted into a database. Since qualitative research promotes focusing on the particularity rather than generalization [26], the main criteria for selecting data items was interestingness, i.e., new viewpoints, topics, insights, and elements for discussion.

All data items were coded using an open-coding approach [30]. All coding cycles were repeated multiple times. Coding was done iteratively until the categorization stabilized, the set of holistic themes emerged, and the essential aspects of BizDevOps crystallized. Each coding cycle was done by one author and checked and revised by the other author. The author revising the codes paid particular attention to coding quality, systematically going back to data to see if the connection could be inferred [29].

The whole process identified 95 documents and produced 364 data items, which had an average of 25 words. The reviewed documents, selected data items, and assigned codes are summarized in the Appendix and provided in GitHub (paantunes/bizdevops-study).

Frequency analysis was not intended for the review. Therefore, we avoided adding similar data items to the database. For instance, data items related to BizDevOps practices were only added if they said something new about that topic. Rather than picking entire paragraphs, we focused on the specific statements that raised interest. Besides, if multiple accounts from a document addressed the same topic, they were put together in one data item.

2.5 Open coding

After building the database of sources, we proceeded with coding (assigning labels) [26]. Adopting the grounded approach, coding was done in three analytic cycles [27,31,32]. The first cycle generated a large inventory of topics related to BizDevOps, potentially encompassing many perspectives. The first coding cycle identified 82 topics of interest.

2.6 Axial coding

The second cycle abstracted the identified topics using a smaller number of categories. The nature of the database, consisting of practitioner experience with BizDevOps, allowed us to focus on social structure, perceptions, experiences, and understanding how things occur; it did not allow for analyzing processes and causal relationships. Similarities in the properties of the initial categories were identified. For example, codes relating to the role and function of individuals and teams within a BizDevOps environment were assigned the axial code of “teams,” and factors identified as likely to contribute to the success of a BizDevOps implementation were coded as “adoption drivers.” The second coding cycle identified 13 categories.

2.7 Selective coding

The last step in grounded qualitative research involved integration and theory building. This required interpreting meaning, identifying holistic themes that cut across the whole database, and condensing and displaying data [27]. Sometimes tables are used. However, we were interested in illuminating the relationships between the themes we identified. For that purpose, we built a BizDevOps framework. This outcome is commonly used to synthesize exploratory research on a topic [36]. Note that this is “a” framework based on rigorous inductive analysis, not “the” framework. Other results might be achieved from the same dataset. The key is the richness and interestingness of the results. “The end result […] is […] the discovery of gaps in knowledge that are important for research explorations with a theory-building focus [that…] results not only in an account of what is empirically found, it also leads to explanations for the findings and offers

insights into what […] might be seen as relevant to the wider world and relevant for the sake of theorizing" [27]. Next, we discuss the outcomes of the coding cycles.

3 Categories Identified From the Grey Literature

The coding procedure identified several categories related to BizDevOps, which are now discussed in detail. For transparency, the data items used to elaborate the following statements are presented in parentheses (letters identify categories, and numbers identify the specific data items). This section shows the results of the first two rounds of coding—open coding, which yielded more than 80 detailed-level codes, and axial coding, which grouped them into eleven categories.

3.1 BizDevOps definition: focus on cooperation and agility

BizDevOps is characterized by practitioners as a strategy, mindset, and set of practices and processes that extend DevOps by bringing Biz (business side), Dev (IT development), and Ops (IT operations) to work together throughout the whole product/service development process (DF1-5). It seeks to provide a deeper understanding and shared knowledge of how IT impacts business performance from a standpoint deeply rooted in business operations (DF4-5,7). It establishes an agile organizational environment (DF1) that continuously pulls IT to deliver business value (DF5-8) in short time frames (DF6).

Given that prior research has already focused on the definitional aspects of the topic [4,37,38], we do not further explore the minor variations in definitions found in grey literature. Nevertheless, we note two different but complementary camps, one suggesting that BizDevOps promotes cooperation (DF3-5,15,23) and another suggesting that BizDevOps promotes agility (DF1,6,17,19).

3.2 Naming variations: focus on technical and business audiences

Some practitioners note that BizDevOps is a natural extension of DevOps, as it extends the benefits of DevOps to the entire organization, not just to IT delivery (DF9-10). Many refer to DevOps 2.0 [4] as an evolution of DevOps, which covers Biz and other concerns (e.g., design and security) (DF9,11-12,17). Others consider the two terms equivalent (DF12), which explains why we collected data about both BizDevOps and DevOps 2.0.

Nevertheless, we noted some nuances when referring to the two terms. When practitioners use the BizDevOps term, they often discuss business vision, strategy, and value delivery (DF2,6,8,18). On the other hand, practitioners using the DevOps 2.0 term usually discuss IT support, including platforms and tools (DF9,13-14). This may reflect different approaches towards "selling" the concept to different audiences, one for the business side and another for the technical side. While recognizing that BizDevOps and DevOps 2.0 may—in principle—be seen as equivalent, the broader adoption of DevOps 2.0 risks making the Biz side less relevant, as it will be diluted within a broad church that covers many other topics.

The term XOps also appears as an umbrella term for a wide variety of propositions (DF15-16), like DevSecOps (security), AIOps (artificial intelligence), GitOps (deployment), DesignOps (design), and others [39]. The term highlights a typical pattern behind all these propositions, emphasizing the urgency in delivering business value and delighting customers by heavily relying on IT (DF3,6,18,25).

3.3 Adoption drivers and goals

Practitioners comment on BizDevOps as a significant emerging trend (GO2-3), which leverages organizations to deliver software virtually any time (GO6,8) through **IT and extensive**

automation (GO10,14). BizDevOps is perceived as potentially changing how software is imagined (GO21), more around value-related concepts than technical aspects (GO23-24).

BizDevOps adoption is driven by the **success of DevOps** [3] (GO5). DevOps has a significant hype, which is demonstrated by the demand for DevOps engineers (GO19) and the number of available DevOps tools (e.g., build automation, continuous integration, continuous deployment, and monitoring) and frameworks (e.g., microservices and infrastructure-as-code). The number of success stories is also significant (GO5-6,8,17). Since BizDevOps extends DevOps, corporates naturally seek to exploit existing methods to support **new business goals** (GO10,16).

Practitioners mention several adoption drivers that are particular to BizDevOps. One is putting the users and **user benefits at the forefront** of everything teams do in organizations (GO12). Another is extending automated **real-time analytics** used in DevOps to the Biz side (e.g., application usage, service monitoring, user experience, and user behavior) (GO13). This makes businesses more data-intensive (GO14).

Another driver is **being ready for change** (GO15,22), which involves flexibility, freedom, and continuous focus on quality (GO18,20). Change and evolution are embraced at the organization's core (GO15). Other drivers include **extending agile practices** in software development to product and business development (GO5), e.g., implementing continuous, real-time quality management (GO18).

Finally, we highlight two aspirational goals. One is to **automate processes that do not need human thinking** or creativity (GO10). Another is to **eliminate the gap from vision to provision**, e.g., where an executive could share an idea with the team while commuting to work to find out the idea has been tested and deployed when arriving at the office[1] (GO17).

3.4 Tackled problems

Besides hype, drivers, and goals, BizDevOps is also presented as a solution to specific organizational problems. One problem solved is responding to more **demanding users** with greater expectations (TP1), **time pressures** (TP9), and **functional needs** (TP2). Another problem solved is quickly settling **service failures** and interruptions (TP3-4). BizDevOps is also presented as eliminating **operational costs** related to legacy, disaggregated, and labor-dependent IT (TP5). Finally, BizDevOps also reduces communication costs (TP6), managerial overheads (TP7), and change-related costs (TP9).

3.5 Cultural changes

BizDevOps is recognized to have strong cultural elements attached. The notion of "**bridging the gap**" is fundamental to understanding the culture behind BizDevOps (CU1). This involves bridging the gap between business, software development, and IT operations (CU2,18,20-21) and bridging the gap between human experience and product/service delivery (CU1). Bridging these gaps requires putting human experience at the center and delivering value through close collaboration with the users (CU1,4).

Even though BizDevOps involves a variety of processes (Biz, Dev, and Ops), the participants in these processes collectively cross over and contribute to deliver value (CU1,3,5). As such, BizDevOps promotes a **flat culture**, which avoids silos and hierarchical barriers, controls, and mentalities (CU6-8). Teams are empowered to define, develop, and evaluate *their* projects (CU1,6).

The preoccupation with **silo mentality** is referred by practitioners (CU7-8). Silo mentality is perceived as conducive to disruption (source of conflict, finger-pointing) and stagnation (strict

[1] This aspirational goal was signaled in the introductory remarks using an example.

control, lack of response). Breaking silos is perceived as conducive to understanding, honesty, openness, transparency, and togetherness CU5,10,18-19,22).

Practitioners recognize that cultural adherence to BizDevOps requires changing mindsets. In particular, the most important aspect is changing traditional hierarchical command and control structures and adopting more collective and **collaborative management** (CU1,6,10). Switching from conventional product workflows based on features/commits towards workflows centered on ideas/values is also mentioned (CU15).

Another change in mindset, which is transported from DevOps, is the focus on change and **continuous response to change** (CU16-17). The mindset toward better assurance (CU10), control (by teams) (CU1), and **immediate sharing** of changes (CU8) is also relevant. Finally, **situational awareness** is essential for effective teamwork in a BizDevOps environment (CU10).

3.6 Adoption barriers

We found in the grey literature several barriers to the adoption of BizDevOps. Organizations with **low tolerance to risk** (e.g., financial and medical) are seen as problematic (BA8) as a great deal of trust is required in automation. BizDevOps conflicts with **regulations and compliance** requirements (BA8). **Technical debt** (infrastructural and architectural) creates significant barriers (BA12). Practitioners note that business stakeholders still lack IT understanding and find it difficult to understand the language of software developers (BA2-3).

The BizDevOps ideal is regarded as challenging to achieve (BA5). At the organizational level, the fundamental problem is that BizDevOps is a **holistic approach** by nature, which means that, ideally, the whole organization should adopt the same set of practices (BA13). However, this makes it more difficult to incrementally change the organization in a way that keeps the implementation risks low. Organizations must also adopt **new business structures** focusing on product/service lines and value streams (BA15).

At both team and individual levels, it may be difficult to develop and maintain expertise in an **amalgamation of soft and hard skills** (BA14). Granting technical access to non-technical team members has some inherent risk (BA9). Biz, Dev and Ops members may also be thrown an endless set of requirements that may be hard to deliver (BA12).

3.7 Organizational changes

Practitioners note that organizations must strive to *be* BizDevOps rather than *do* BizDevOps (OC1). In other words, adopting a BizDevOps culture seems more relevant than implementing specific BizDevOps processes (OC15). It has been noted that delimited approaches to BizDevOps create an organization that operates with two different speeds and cultures (OC2). Unless teams are **entirely autonomous**, conflicts will emerge from dependencies between the two parts of the organization (OC4,7,9).

More holistic approaches to BizDevOps lead organizations to structure themselves around **value streams** (OC3,5). Practitioners suggest moving the more impactful products/services to BizDevOps first and then using their leverage to pressure all other products/services to embrace BizDevOps (OC6,13-14). Leverage comes from quick wins, release frequency, user satisfaction, leaner processes, and new IT platforms such as microservices (OC8).

Practitioners note that organizations embracing BizDevOps cannot depend heavily on partners, vendors, and outsourcing (OC10). Instead, the organization should focus on **insourcing** and reshoring the required capabilities (OC11-12,16). This includes people and technology. External IT services and platforms can be limiting, as BizDevOps teams must shape and control the whole IT in-house (OC12).

Practitioners note that BizDevOps implementation requires specific **governance rules**. BizDevOps requires autonomous and self-determined teams (OC4). Operational decisions, such as selecting methods and technologies, must be made by teams (OC17,21). Organizational structures must become flat, command and control should be eliminated, and more collaborative decision-making approaches should be assigned from the top (OC15,18). Risks and rewards should be taken by teams rather than distributed in layers (OC7). However, guardrails are necessary to avoid teams spinning out of focus (OC19).

In line with the BizDevOps mindset, **automation** should be used to monitor teams. Risk management is necessary and should be based on automation. Quick risk control mechanisms should be adopted using automation (OC20). This includes, for instance, continuous sensing and tracking, root cause analysis, and the ability to quickly flag problematic products/services (OC20).

3.8 Teams

Practitioners note that BizDevOps teams must transition from segmented, project-oriented, and tightly scheduled jobs towards more **continuous and collaborative functions** (TE1,8,10), with everybody working closer to the front line (TE8). Teams should also transition from specialized roles (e.g., requirements, development, and testing) toward **wide-ranging roles** covering all aspects of Biz, Dev, and Ops (TE2,4-7). However, this more rounded role is considered difficult to achieve in practice. Separation in two teams working together, one composed of business analysts and another by developers, seems common (TE3,9,11); the main reason is difficulties finding people capable of operating across the two roles (TE2).

Practitioners also recognize that it is difficult to move between roles without resistance; it involves more work and brings people out of their comfort zones (TE15,16). For these reasons, many see BizDevOps not as a single-team collaboration but as a set of **multiple-team collaborations**. One interesting metaphor suggests regarding BizDevOps teams as **micro-companies**, which are flexible to a certain point but still require some internal structure (TE17-18).

Besides the business and developer teams mentioned above, we have also seen mentions in grey literature of segment, service, and platform teams (TE13), and frontline and enabling teams (TE9). However, segmentation seems to go against the BizDevOps ideal and could result in the resurgence of silos, or what we would call “bogus” BizDevOps, where different teams steer in different directions (TE19). Regarding team size, we have seen mentions between 5-6 and 8-9, which seems to correspond to the so-called “two-pizza” rule [40] (TE20-21).

One challenging aspect of BizDevOps is developing a **common language** among team members, with a common terminology used across business and IT (TE45-46). The whole organization must share concepts such as service-level objects, minimum viable products, and product hypotheses. Teams must also share metadata (TE47).

At a more macro level, some organizations arrange teams in **product/service tribes**, each responsible for a specific product or service (TE12-14). Each tribe has a lead who can act as mini-CEO (TE14).

Discussions about team collaboration are also plentiful. **Interaction, collaboration, and alignment** are expected daily (TE25-26,28). Practitioners have regular breakdown sessions with all members in the same room, with the door shut, to make collective decisions (TE22-23,27). Teams also negotiate the meaning and desired levels of collaboration (TE24). One important aspect of collaboration and alignment is hearing the “voice of the customer” [41], which is omnipresent (TE29). Another aspect is the collective management of requirements (TE10,19,22,30-34). Indeed, one mentioned metaphor is that managing requirements is a “team sport” (they succeed and fail together) (TE33). Design thinking (exploration, ideation, sketching, visual modeling, prototyping) (TE36-37,41,43), agile thinking (user stories, feature requirements, hypotheses, minimal viable

product) (TE25,31-32), proactiveness (anticipating user needs) (TE11,25), and definition and constant control of metrics (TE40), have also been identified as regular elements of team collaboration.

Some practitioners regard BizDevOps as being divided between two roles. One role, which could be named junior BizDevOps, covers various operational tasks, such as business analysis, software development, testing, and automation (TE40). The other role, senior BizDevOps, is holistic and focused on advocacy (for BizDevOps culture and processes), steering (projects and teams), visioning, and architecting (TE40).

Regarding managerial roles, they tend to focus on product/service delivery (TE58). **Delivery managers** support teams rather than taking control and responsibility (TE54). **Capability leaders** focus on requirements (TE58). CIOs' responsibility is supervising technical debt and delivery quality (TE57).

3.9 Operational patterns

In this category, we highlight a set of best practices brought by the grey literature.

Speed in product/service updates is frequently mentioned (BP1-3,5,9,87). Processes are in place to deliver updates at any time as needed, on a weekly, daily, or hourly basis (BP1,69). In combination with speed, organizations constantly **sense-and-respond** [41] to the different parts of the organizational system, including teams, processes, and customer interactions (BP6-7). Sense-and-respond integrates with **instant-feedback** (BP9-17). Organizations sense-and-respond in real time to failed product/service updates, user complaints, development feedback, and testing (BP6,9). The combination of speed, sense-and-respond, and instant-feedback bypasses traditional (deferred) feedback channels such as sales and customer relationships (BP10). Another point of view is that the speed, sense-and-respond, and instant-feedback capabilities brought by BizDevOps can be carried through to business development, marketing, sales, customer relationships, and finance (BP9,17). These capabilities increase opportunities for innovation, new revenue growth, and brand exposure (BP11).

Visibility helps teams understand and explain what they are doing (BP18-19). For instance, developers can continuously nurture and test new features (BP18-19,94), and business analysts can constantly test the technical implementation of features (BP12). Related to visibility, utilizing real-time **dashboards** with technical and non-technical key performance indicators is a recognized best practice (BP20-21,23,89). Even though these dashboards heavily rely on automation, practitioners note that their holistic interpretation still requires considerable human insights (BP22,32).

BizDevOps involves building and maintaining **well-defined processes** (BP27-28). They establish key performance indicators to which BizDevOps teams are expected to respond dynamically (BP27). Well-defined processes are seen not as a roadblock to teams' responsibilities but to reduce risks (BP28). Furthermore, well-defined processes facilitate automation (BP48).

BizDevOps teams keep the **connection to users** at the center of their concerns (BP29-30,32-35), which allows tailoring technical solutions (BP30). However, this requires having accessible communication channels with users and being capable of gathering genuine insights from user experiences (BP32,34). Practitioners mention that using digital platforms to interact and collaborate with users is essential for success (BP36-37).

Good requirements help develop good roadmaps (BP38). **Continuous requirements management** is perceived to improve product/service delivery (BP39,41). It is also perceived to improve continuous monitoring, as operational indicators close the loop with requirements (BP11,36,40). Finally, BizDevOps also involves collaborative and holistic requirements management (BP36). User stories and product backlogs [42], which are common practices in Agile, may not be adequate for BizDevOps, as they are not collaborative enough.

Automation is central to BizDevOps (BP42,45). Automation enables speed, sense-and-respond, and instant-feedback (BP42). It also allows robotic process automation (BP43-44), which uses software robots to automate certain aspects of process execution, in particular automating routine code development, deployment, and quality assurance (BP46,70) [43]. Finally, automation also helps maintain and monitor the link between requirements and software development and delivery (BP48).

Tight integration promotes the alignment between the different processes involved in BizDevOps, avoiding long handovers and backlogs (BP49-50,52-53). Decisions and approvals of new features, priorities, considerations about value delivery, and performance analysis should be collaboratively made by teams (BP51). Tight integration includes more specific practices, such as "shift left," which brings testing closer to design and development by exploiting operational data (BP52-54) [44]. It also includes making corporate information more transparent and accessible to users (BP53), e.g., using self-service systems to help identify and resolve issues (BP15).

Practitioners note that teams must "liberate" data (BP84). Unlike traditional approaches to enterprise systems, which promote data integration, more recent approaches foster **data independence**, where teams take ownership of data assets for advantage (data-as-a-service) (BP57,60) and observe how data flows (BP85). Data independence also contributes to developing more flexible, modular, responsive, and agile microservices while increasing resilience (through distribution and replication) (BP55). BizDevOps teams define data management rules (BP56-57). Organizations must define reference architectures and governance rules for BizDevOps data independence (BP57). Data management is often considered missing from the BizDevOps picture (BP58). However, it is a strategic asset (BP58), which should be blended with data science techniques (BP59) to speed digital innovation (BP60).

Continuous value delivery relies on streamlined and controlled **feature rollouts** and quick rollbacks in case of failure (BP61,67,91). Practitioners recommend delivering features in stages to users (one-click deployment) (BP62) and using dashboards to monitor and control in real time the rollouts (BP63-65). If rollouts do not work, they can be immediately withdrawn (BP68). **Flag-driven development** goes even further. It involves making available features to user segments for testing purposes (BP66). This allows teams to dynamically test features by toggling experimental features on and off (BP67).

Embracing **user-centered deployment**, teams can sit down with users, discuss issues, and implement fixes or deploy specific functionality to selected users (BP92-93,96). A/B testing of different versions of the same functionality can be done with different users (BP94).

3.10 Techniques and tools

As an extension of DevOps, BizDevOps naturally inherits DevOps techniques and tools. Here, we report on techniques and tools that support the Biz extension.

Practitioners refer to **metadata management** tools (TO36) and **process modeling** tools (TO1) to detail the processes, tasks, responsible parties, and data assets involved in BizDevOps. These tools can be complemented by **application performance management** tools, which provide constant insights on various metrics related to processes (TO2). In fact, practitioners note that BizDevOps is only possible if there are tools supporting **real-time analytics** (TO3).

At an operational level, tools that **control available functionality** based on business rules are also advocated (TO4). Such tools extend change management to the whole workflow, from Biz to Dev and Ops (TO5). As multiple metrics can be used to measure performance, a **metrics map** covering business requirements is also a suggested tool (TO6).

At a more strategic level, using **Kanban boards** helps keep teams focused on an environment that can be quite complex (TO7). **Chaos engineering** tools [45], which help monitoring, logging,

and failure detection, can also be used from a Biz perspective (TO8). **Value stream mapping** [46] helps teams define and verify their business capabilities and operational objectives using diagrams (TO9-10). Signoffs from business leads are expected to be automated based on executable policies (TO11).

Low-code platforms are also regarded as necessary to BizDevOps (TO12). They allow businesspeople to change products/services without tapping into technical details.

3.11 Challenges and risks

Practitioners note that BizDevOps is intimidating, as it includes an **overwhelming list of requirements**, and finding where to start can be challenging (CH1). It involves many stakeholders with different needs (CH6). There are **too many tools** to choose from (CH2). The **cognitive load** is high (CH2): products/services can become too complex to be owned by individual teams (CH3); and given the continuous development pace, finding breaks to be creative can be a load (CH13).

As BizDevOps primarily focuses on operations, **upfront planning** is challenging (CH5). The stable parts of the organization may clash with the BizDevOps dynamics (CH7). Running dozens, even hundreds of products and services while continuously deploying new and redeploying existing ones can be problematic for some organizations, especially the ones more constrained by compliance and regulation (CH4,6).

While BizDevOps seeks to break down silos, **bridging business and technical views** may still be difficult (CH9-10). The main reason is that, while Dev and Ops are strongly tied, i.e., Ops starts exactly where Dev finishes, the same cannot be said about Biz (CH9).

The **complexity** associated with BizDevOps is high. Even though BizDevOps processes aim to monitor product/service delivery continuously, there is the risk of constantly spinning out of control, particularly when failing is not accompanied by reflection and improvement (CH1).

3.12 Wide implementation

Practitioners also make specific comments regarding the wide implementation of BizDevOps by organizations. Implementation requires **balancing the business and technical** facets (IM1). Another key aspect to consider is to relentlessly **streamline and automate** internal operations (IM2-3).

Practitioners note that BizDevOps environments have **highly standardized processes and IT** (IM3-4). A suggested implementation approach is to establish core IT teams that encapsulate legacy technology and service teams focused on delivering reusable services (IM5).

Change, time sensitivity, and product orientation are considered essential implementation criteria. Embracing change is a key principle framing the BizDevOps implementation (IM8). Time sensitivity refers to the understanding that if new initiatives take too long to develop, then they should be redefined or abandoned (IM6). Product orientation refers to the understanding that all teams and initiatives should be product-centered, with active involvement from business and few handovers (IM9-11,17).

Practitioners emphasize that BizDevOps implementation requires changing the organizational structure to support **flat, self-determined teams** (IM10-12). The organizational structure also needs to manage **streamlined requirements practices** (IM16), with **stakeholder participation** in decision-making (IM15) and **continuous response** to internal and external feedback (IM14).

3.13 Open issues

A small number of open issues have been identified in the data set. One interesting issue relates to the utilization of low-code development by BizDevOps teams, which is seen as eroding the **"idea"**

of software developers (OI1). Another open issue relates to **expectations around observability**, where various technical and non-technical metrics can be used to test ideas in various ways (OI2).

Another open issue is related to the **sociotechnical evolution** of organizational systems, which are architected around a large number of teams sensitive to the environment (OI3). Finally, it has also been noted the increasing **demand for new types of engineers** with the ability to use a variety of open-source software to build and assemble independent bits and pieces of code, instead of performing traditional development activities (OI4).

Figure 2 provides a synthesis of the categories and associated topics extracted from the grey literature, not considering the first two categories as they relate to definitions and naming variations. Next, we discuss the links between these categories.

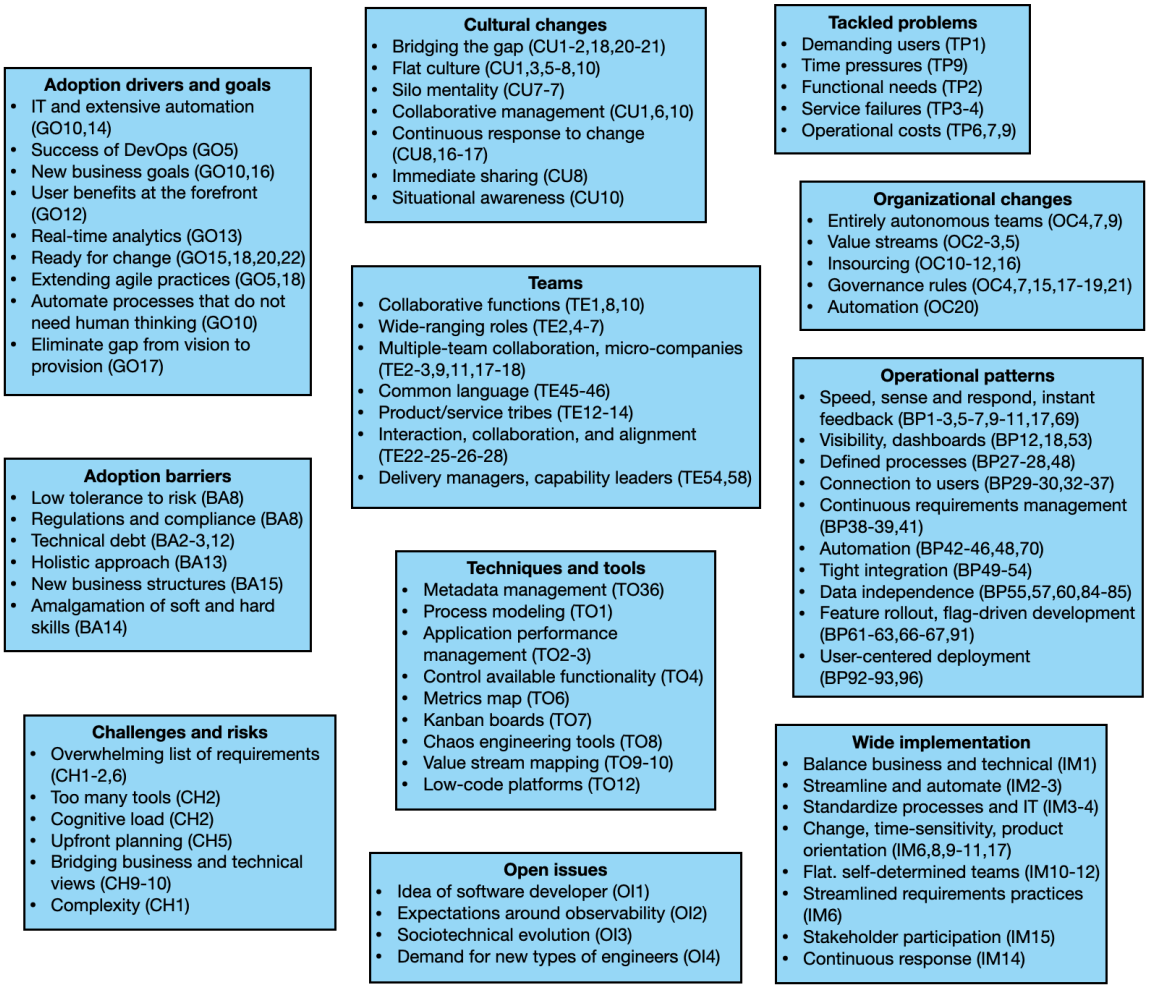

Figure 2: Categories identified in the grey literature

4 Conceptual Framework for BizDevOps Planning and Implementation

We now identify a set of themes and critical links [30] between the categories discussed in the previous section and use those links to build a conceptual framework for BizDevOps planning and implementation (Figure 3). We recognize three interconnected themes: front line, BizDevOps as changes to a sociotechnical system, and an embedding pattern.

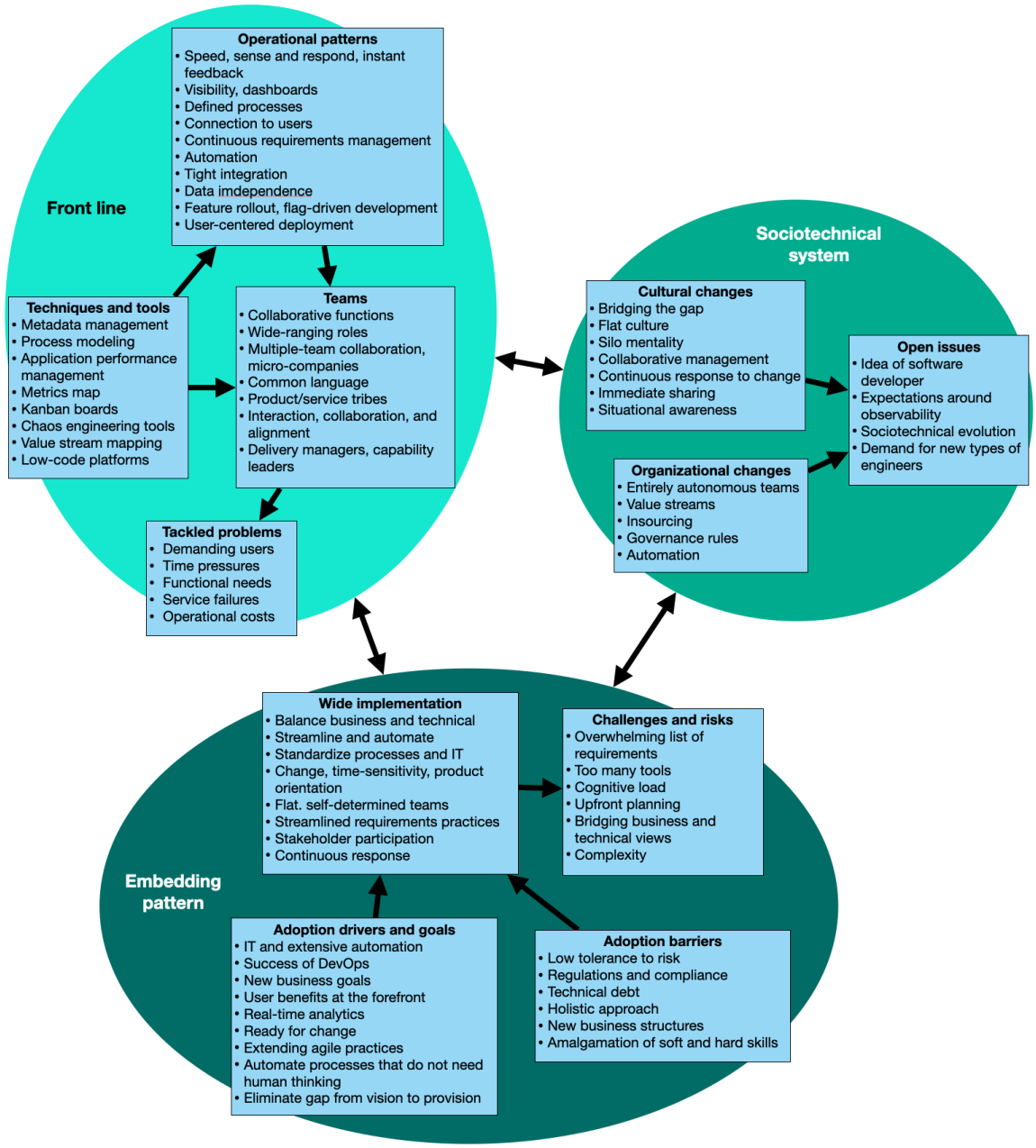

Figure 3: Conceptual framework for BizDevOps planning and implementation

4.1 Origin in the front line

We found in the reviewed data that BizDevOps planning and implementation emerges at the front line, i.e., close to where the vision is built into the software, put in front of the users, and evaluated by the BizDevOps team (TE8,11,25,28). This is in contrast to many strategic IT adoption decisions that are often made top-down, as it requires changes to the "traditional" dynamics of strategic technology adoption.

Various techniques and tools supporting end-to-end process automation make this possible, from requirements identification to deployment and real-time feedback (TO3,5,11,16-21). For

instance, one source notes: “I found the new testing tools […] just ideal to create a tightly integrated flow between Business (Biz), Development, Testing and Operations – BizDevOps.” Interestingly, both tools for technical functions, such as low or no-code platforms, and tools for management, planning, and monitoring in complex environments, such as Kanban boards, value stream mapping, and chaos engineering tools, were identified as essential front-line tools. Suites of tools that enable BizDevOps functions to be not only performed but simultaneously managed and evaluated in real-time at the front-line form the starting point of BizDevOps implementations.

Effective use of these tools requires an effective team structure with wide-ranging roles that span the range of business and technical competencies required. As pointed out by one source, “BizDevOps not only means getting together during the start or design of a project: it also means getting together during the run phase. Sit behind the desk of end users. Feel what they are experiencing when they have to wait five seconds during each and every login” (TE28). BizDevOps is predominantly driven by IT, which encourages teams to develop specific characteristics, e.g., “working collaboratively to maintain and further develop a service for its entire use cycle” (TE1), and to exhibit certain operational patterns, like “speedy and interdisciplinary communication and decision-making” (BP5). One source notes that “the pressure to deliver great customer experiences while spending money wisely has led […] to […] setting up dedicated BizDevOps teams to create granular and specialized services” (TP5).

Building from suites of tools and an appropriate team structure, operational patterns for work at the front line begin to emerge. For instance, practitioners note that “[tools] inform yourself in real time around what is working and what is not” (BP15), which then creates the “ability to control, through a control panel interface, the launch of new features of applications in production environments […] by both technical and non-technical people” (BP61).

Relying on operational patterns, techniques, and tools, teams can tackle problems related to increasing user demands, time pressures, and system failures. One source notes that “reliance on legacy technologies that require manual orchestration […] cause high operating costs” (TP5) while another notes that with BizDevOps, “[r]eal-world services can be up to date, available, and robust without interruption” (TP4). A demonstrated ability at the front-line to solve problems rapidly while controlling operational costs motivates organizations to understand how to make the necessary organizational changes required for wider adoption and embedding of a BizDevOps approach.

4.2 Managing sociotechnical systems

The reviewed data emphasizes that adopting BizDevOps comes with significant sociotechnical implications. BizDevOps teams, with their collective and flat structures, require high levels of collaboration (CU1). As noted by one source, “[s]teps that evolve culture by promoting structures that are not hierarchical but instead flat and empowered, focusing on teams and people instead of projects, and preventing us against them mentalities or command and control management styles will be what truly enables BizDevOps adoption” (CU6). At the same time, the organization needs to embrace significant organizational changes regarding a collaborative approach to performance (CU13), outcomes and value (CU15), and rapid responsiveness to change (CU1,9).

The “sociotechnical” term highlights that cultural and organizational changes are entangled and technologically driven. The reviewed data highlights the importance of the use of IT to improve processes, such as requirements management (TE31-32), speedy product releases (BP3,5), streamlined communication (T354), instant feedback (TE31,50, BP10-14), real-time visibility and situation awareness (BP18-19), and transparency (CU18,53). The dynamic utilization of various metrics (BP23), in combination with monitoring and visualization tools, is also a defining characteristic of BizDevOps (TE42-43, BP20-22). A source notes that “[i]n contrast to a product

manufacturer with clearly defined departments in a classic vertical structure, digital services call for workflows that prioritize speedy and interdisciplinary communication and decision-making" (TE49). Another source notes that "BizDevOps also has a significant dependency on tools that give real-time business metrics" and "the focus here is to implement a real-time dashboard of business KPIs that provides a clear indication of the business value delivered with every release" (BP23).

Various sources point out that IT platforms drive many sociotechnical changes, including software automation (TO16), process orchestration and monitoring (TO20-21), and microservices (TO22). Microservices, "an architecture that breaks an application into many small and loosely collected services" (TO22), provide the foundation for a variety of sociotechnical features of BizDevOps, such as "setting up dedicated BizDevOps teams to create granular and specialized services" (TP9), "enable frequent releases" (BP76), and "shift left" (BP71).

More radical views over BizDevOps indicate profound changes in software development, moving toward low-code (OI1), management, moving away from planning and control toward observability (OI2) and "shared understanding" (CU10). A more holistic view of organizations as sociotechnical architectures and systems is also envisaged (OI3). This implies, for instance, organizing businesses around available functionalities (BP90), defining business priorities and goals around very short experimental and exploratory practices (BP74-75,83,86-87), and envisioning value delivery around individual users (BP90).

However, these changes are challenging. A number of issues were identified. In particular, there is some evidence that enacting the necessary changes requires "new types of engineers" and a different idea of what it means to be a software developer.

4.3 Embedding pattern

The success of BizDevOps in the front line leads organizations to see the approach as a "competitive advantage" (IM2), which "balances both technical and business needs" (IM1) and "streamlines and automates internal operations" (IM2). Therefore, it seems natural that organizations seek wide implementation (OS6).

Wide implementation leads toward what we characterize as the "embedding pattern." The term highlights the methodologic traditionalism identified in the data sources, where organizations ponder the adoption drivers against barriers and consider opportunities and risks. Drivers include features such as automated feedback (GO5), embracing change (GO15), and the possibility "to automatically provide everything needed to handle a new feature" (GO14). Barriers include "technical debt" (BA12), as organizations need to secure particular types of technologists who embrace the "bits and pieces needed for a solution" (GO19) and non-technical personnel capable of understanding application software (BA2,9). Opportunities include the capacity to better deliver business value (OS3) and do it faster (OS2,8) and with better customer experiences (OS6). Risks include organizational complexity, where you may not be able to control when and how things change (CH4).

Technological platforms are also an aspect to consider in the relationships between the front line and the wider organization. Organizations need to "[t]ransform the core IT landscape by distributing IT systems to teams of teams and gradually replacing them by granular services" (OS13). This can require different ways of working and different knowledge areas and generates cognitive load for both teams and organizations (CH3).

One interesting aspect of embedding is standardization. One practitioner notes that "everything is pretty standardized almost like in the mainframe era" (IM4). Such a level of standardization is required to allow different teams to operate together and to take advantage of automation. Standardization can be accomplished at different levels, for instance, regarding technical service provision (IM13), product thinking (IM9), feedback (IM14), stakeholder participation (IM15), and

standard metrics (IM14). It should be noted, however, that standardization does not imply hierarchical command and control, as BizDevOps teams are expected to operate independently (IM10).

Still considering the relationships between the front-line and the wider organization with regard to the embedding pattern, we note that the front-line pushes the organization to embrace and expand the BizDevOps approach but there is also a need for a significant "pull" from the wider organization towards the front-line. In particular, the organization needs to provide commitment and leadership on issues related to command and control and the team's self-determination (IM10,12), risk tolerance (BA9), trust (BA14), and "minimum governance and maximum synchronization and autonomy" (BA15). Sources also note that organizations need to be "100% BizDevOps" (OS2); otherwise, different parts of the organization with different speeds will clash, and "[d]ifferent groups pulling in various directions create an ongoing battle" (OS2).

Finally, both teams and organizations need to find ways to manage a complex, "overwhelming and intimidating list of suggested initiatives (or lack thereof)" (CH1). Re-platforming or replacing technologies is not only an implementation requirement for BizDevOps (CH11), but it also becomes a relentless operating model for teams and organizations (IM8). Embedding is not a single step or process that has an end-point. Developing and absorbing new sociotechnical systems needs to be a continuous part of organizational practice.

Overall, the framework is not intended to be used as a prescription, where organizations are expected to move from A to B in a stepwise manner. Nevertheless, the interconnections between the three identified themes allow organizations to approach BizDevOps in a structured way. The framework maps the BizDevOps territory, supports organizing vision, and helps build story arcs. Organizations wishing to implement BizDevOps are expected to *make* sense and *give* sense to this map by creating plausible implementation strategies [47].

5 Discussion

As the research and practice of BizDevOps becomes more widespread, researchers and practitioners can contribute to developing more significant insights about appropriate decisional, organizational, and team structures. Next, we reflect on the proposed framework, identifying the main issues and challenges with BizDevOps, pointing out the main study contributions for research and practice, and suggesting future research.

From a research perspective, a number of areas of theoretical interest emerge. We touched earlier on story arcs in grey literature and theories of change. As a story arc in grey literature, BizDevOps is still in the early stages of the Gartner "hype cycle" [20]. However, the grey literature has already identified the challenges of managing the associated, ongoing sociotechnical changes and the significant factors in successful implementation. The powerful business impacts of BizDevOps technologies and practices, even in their current early stages, suggest that a technology-focused view of the "hype-cycle" that builds from less mature to more mature technologies may have limited applicability to the BizDevOps story, as it seems that the sociotechnical changes, rather than the technology, are already perceived the main constraints of successful adoption. Organizing vision theory appears to be a better fit for explaining the insights that can be obtained from the practitioner discourse. Important themes from our analysis include questions about why BizDevOps is valuable and how organizations can embed the BizDevOps approach. "Mobilizing" the tools that already exist is identified in the practitioner discourse as a significant challenge. Managing complexity is another.

Considering the management of technology-related change, interestingly, the origins of BizDevOps implementation are frequently technical, operational, and front-line, rather than top-down. However, in order to become embedded in the organization, a continuous, networked

approach to change management needs to be adopted. Traditional "punctuated equilibrium" change cycles seem to have limited applicability, while continual, real-time, networked, and whole-of-organization approaches to change are required.

There are also some broad areas of interest for research and practice where significant changes to existing understandings are suggested. These are summarized in Table 1 and discussed below. The discussion is then concluded with threats to validity.

Table 1: Higher-level implications to practice and research from this study

Topic	Implications to practice	Implications to research
Role of management in organizations: management-as-a-service	Managers to operate as facilitators, promoters, and stakeholders in an everything-as-a-service organization	Research how management can operate as a service
Processes for planning, forecasting, and process management	Continuous flow of idea-to-value	Investigate the impact of continuous requirements management
Design processes	Design process driven by feedback automatically generated by the IT infrastructure	Investigate the rise and potential long-term impacts of "hit & run" design
Organizational capabilities: "agility debt"	Organization are quickly accruing agility debt	Investigate the agility debt

5.1 Management-as-a-service

We do not claim that as traditionally understood, management in organizations will disappear. However, organizational structures associated with managing BizDevOps will be more sociotechnical, disintermediated, blurred, and tactical, continuously changing in multiple places and directions according to different interests and opportunities (Figure 4). The grey literature indicates that BizDevOps promotes technical and organizational decentralization (CU6-10, OS3,6,9,15,18, TE1,4). Decentralized infrastructures and decision-making lead BizDevOps teams to pay more attention to feedback coming directly from users, products/services, platforms, and other BizDevOps teams than managerial guidance (CU4-5,10, TE14,17-18). The whole organization uses communication to increase continuous listening, visibility, honesty, transparency, and trust (CU22, OS15, TE26,28,36,49-50, BP6,17). It also puts experts in the lead, even though like in team sports (TE30, TE33,44). This raises interesting questions regarding traditional managerial roles in organizations, e.g., leading, planning, and people management. BizDevOps challenges managers to become less preemptive and prescriptive and instead operate as facilitators (between BizDevOps teams and other parts of the organization), promoters (of values, ideas, and goals), and stakeholders (with business requirements and constraints) with a greater understanding of how software affects the business (TE22-23,32-33,52). Future research should focus on how management can operate as another service in an everything-as-a-service organization.

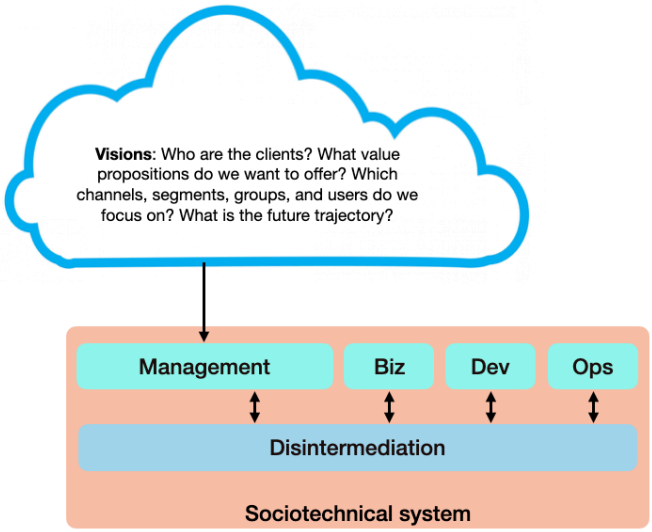

Figure 4: Management in BizDevOps organizations

5.2 Planning, forecasting, and process management

BizDevOps may disrupt our understanding of business planning cycles. Business management usually thrives on integrated planning, forecasting, and process management [48]. For instance, approaches to business process improvement are typically planned in two stages: one, where existing business processes with potential for improvement are analyzed and modeled (as-is), and another, where optimizations and innovations are applied in relation to value claims (to-be). [49]. However, the decentralized, speedy, and autonomous decisions made by BizDevOps teams make it challenging to operate in a fully planned manner. BizDevOps teams operate close to the front-line and thrive on quick testing, immediate decision-making, and a continuous flow of idea-to-value (CU9-10,15, TE4,25,27,49). Prior research suggests the inclusion of a planning component into BizDevOps [5,37]. For instance, the popular visualization of BizDevOps shown in Figure 1 includes a planning stage. The BizDevOps workflow proposed by Chasioti [6] also includes planning, even though the author notes that planning in the long run "is not suitable, due to the rapidly changing market and customer needs" [6]. Interestingly, however, the analyzed grey literature *does not* support a formal, linear or iterative, planning component. Instead, it supports integrated continuous requirements management as a more focused and dynamic approach to business management performed on the front line (TE19,22,26,32-35). Future empirical research should investigate the impact of continuous and simultaneous planning, requirements management, implementation, and evaluation brought by BizDevOps.

5.3 Design processes

Our analysis suggests a significant change in design processes. Design is traditionally performed before software development, generating and validating whole products/services with users before they are brought into the software, e.g., through prototyping. However, the grey literature indicates that design will be embedded in BizDevOps teams and mainly driven by feedback automatically generated by the IT infrastructure about what is working and not working (BP3,5,9-12,14-16,52-53,55,68). BizDevOps seems to promote a "hit & run" approach to design [51] that is fast, pragmatic, opportunistic, localized, and scrutinizing rather than collaborative and relatable. Future research should investigate the rise and potential long-term impacts of such a "hit & run" design approach. The related literature points out that BizDevOps addresses user needs through

continuous software engineering, where product/service requirements, increments, and reviews are quickly and continuously performed with the participation of users [5,50]. The grey literature points out a variety of practices and tools adopted to accomplish this endeavor, such as feature rollouts, user-centered deployment, streamlined requirements, automated workflows, and real-time dashboards (BP20-21,23,28,32,35-36,45-48,52,61-62,66-70). These are essentially engineering practices. They address the user needs with a strong focus on automation, real-time analytics, and performance metrics.

5.4 Agility debt

The grey literature indicates that BizDevOps represents another significant step in speed and responsiveness, accomplishing an almost real-time ability to develop or change products and services (TE49-50, BP2-3,5,42,68,73-74,94). It also highlights the challenges in managing the associated sociotechnical system changes and embedding new patterns. However, these changes can also bee seen as an extension of existing Agile practices. Agile software development practices are now mainstream [52]. It is likely that organizations that are not already progressing down the path of integration and automation of software development and operations are quickly accruing an "agility debt" [53] (Figure 5). DevOps increased the speed and responsiveness with which organizations realize changes [1,54]. This will likely further improve the competitive position of organizations with this capability and further increase the difference between them and organizations still stuck in Agile software development practices without the integration, automation, and feedback offered by BizDevOps. Future empirical research could investigate the agility debt as the new generation of "technology debt" From a practical point of view, we offer a comprehensive list of resources required for organizations implementing BizDevOps.

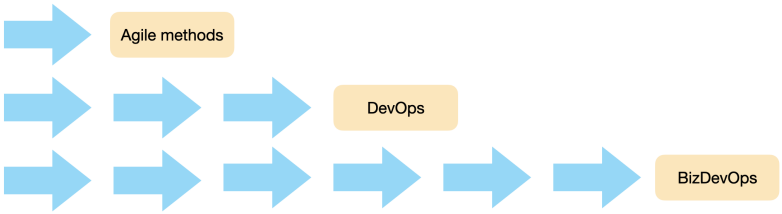

Figure 5: Increasing "agility debt"

5.5 Threats to validity

Grey literature can be used to (among other things) increase breadth, offer a practitioner perspective, and reduce publication bias [19]. However, the use of grey literature has some challenges, including difficulties measuring quality, ongoing availability, and approaches for searching grey literature [19]. It is, therefore, essential that grey literature analysis is conducted only where it is appropriate to the research question and with full cognizance of the limitations of the approach. Given the recognized difficulties in achieving ongoing access to grey literature sources, grey literature sources have limited value for repeatable structured literature reviews.

Another limitation concerns the type of data collected and the extent it reflects the practices and views of the industry. Professional insights offered in grey literature can be biased by hype and the need to offer a positive outlook on the phenomenon and sources. It may also lack in-depth reflection on BizDevOps. As researchers cannot guide the sources to focus on certain concepts and constructs, some may have been overanalyzed while others may have been neglected.

6 Conclusion

Interestingly, although BizDevOps has substantial organizational implications, it has mainly emerged bottom-up from Agile and DevOps. Our review suggests that fully agile and immediately responsive business-technology teams that can respond in near-real-time to contextual forces and opportunities can now be a reality. However, we also emphasize that this ability is hard-won and requires integrating a demanding combination of components. It seems that organizations need to "pass-through" DevOps on their journey to BizDevOps.

Organizations already developing "bundles" of different but related value propositions based on many flexible, decentralized products/services may have a natural advantage in moving to BizDevOps. This can include (for example) digital platforms, cloud marketplaces, media outlets, banking, and insurance. Some practitioners have suggested that new offerings can be "flag-driven," so they are offered first to a small subset of users to obtain real-time feedback. Organizations moving towards an everything-as-a-service paradigm [55] may already have the proper adoption context for BizDevOps. Examples include "gig economy" platforms, subscription-based software providers, and crowdsourcing.

From an organizational perspective, the entrustment of products/services to teams begs the question of the role of management in organizations. BizDevOps is a much faster, more responsive, and, in some ways, more chaotic form of conducting business, linked by independent visions rather than an articulated strategy.

Many of our sources commented on the complexity, even chaos, associated with BizDevOps and potential difficulties with planning and management—to which we add design. However, the practices of continual monitoring and continual user engagement should enable problems to be identified and corrected rapidly.

Where are we at with BizDevOps? The practitioner community believes it has arrived, and with it, a degree of IT responsiveness to the business and overall business agility that were previously unthinkable. However, an effective BizDevOps system is highly complex and many-faceted. It requires a dynamic front-line, effective management of the sociotechnical system, and embedding into a new, thriving organizational form.

References

[1] E. Dörnenburg, The path to devops, IEEE Software 35 (2018) 71–75.

[2] S. Misra, V. Kumar, U. Kumar, K. Fantazy, M. Akhter, Agile software development practices: evolution, principles, and criticisms, International Journal of Quality & Reliability Management 29 (2012) 972–980. https://doi.org/10.1108/02656711211272863.

[3] L. Leite, C. Rocha, F. Kon, D. Milojicic, P. Meirelles, A survey of DevOps concepts and challenges, ACM Computing Surveys 52 (2019) 1–35.

[4] I. Lohrasbinasab, P. Acharya, R. Colomo-Palacios, BizDevOps: A Multivocal Literature Review, in: International Conference on Computational Science and Its Applications, Springer, 2020: pp. 698–713.

[5] B. Fitzgerald, K. Stol, Continuous software engineering: A roadmap and agenda, Journal of Systems and Software 123 (2017) 176–189.

[6] K. Chasioti, BizDevOps: a process model for the alignment of DevOps with business goals, Utrecht University, 2019.

[7] P. Forbrig, Does Continuous Requirements Engineering Need Continuous Software Engineering?, in: REFSQ Workshops, 2017.

[8] V. Gruhn, C. Schäfer, BizDevOps: because DevOps is not the end of the story, in: International Conference on Intelligent Software Methodologies, Tools, and Techniques, Springer, 2015: pp. 388–398.

[9] R. Alt, G. Auth, C. Kögler, Continuous Innovation with DevOps, Springer Nature, Cham, Switzerland, 2021.
[10] Blueprint, BizDevOps Enabling Digital Transformation, Blueprint (2017). https://www.blueprintsys.com/blog/bizdevops-digital-transformation (accessed January 17, 2023).
[11] P. Forbrig, BizDevOps and the role of S-BPM, in: Proceedings of the 10th International Conference on Subject-Oriented Business Process Management, 2018: pp. 1–8.
[12] M. Sony, S. Naik, Critical factors for the successful implementation of Industry 4.0: a review and future research direction, Production Planning & Control 31 (2020) 799–815.
[13] M. Kersten, A cambrian explosion of DevOps tools, IEEE Software 35 (2018) 14–17.
[14] R. Kurzweil, The law of accelerating returns, in: Alan Turing: Life and Legacy of a Great Thinker, Springer, 2004: pp. 381–416.
[15] C. Loch, B. Huberman, A punctuated-equilibrium model of technology diffusion, Management Science 45 (1999) 160–177.
[16] S. Cummings, T. Bridgman, K. Brown, Unfreezing change as three steps: Rethinking Kurt Lewin's legacy for change management, Human Relations 69 (2016) 33–60.
[17] T. Peters, R. Waterman Jr, McKinsey 7-S model, Leadership Excellence 28 (2011) 2011.
[18] M. Amadoru, E. Fielt, M. Kowalkiewicz, R. Nayak, Organizing visions in online social networks: The role of community heterogeneity and real-time engagement, in: Proceedings of the 39th International Conference on Information Systems, Association for Information Systems, 2018: pp. 1–9.
[19] F. Kamei, I. Wiese, C. Lima, I. Polato, V. Nepomuceno, W. Ferreira, M. Ribeiro, C. Pena, B. Cartaxo, G. Pinto, others, Grey literature in software engineering: A critical review, Information and Software Technology 138 (2021) 106609.
[20] Gartner Group, Gartner Hype Cycle, (2023). https://www.gartner.com/en/research/methodologies/gartner-hype-cycle.
[21] V. Stray, N. Moe, M. Noroozi, Slack me if you can! using enterprise social networking tools in virtual agile teams, in: 2019 ACM/IEEE 14th International Conference on Global Software Engineering, IEEE, 2019: pp. 111–121.
[22] E. Swanson, N. Ramiller, The organizing vision in information systems innovation, Organization Science 8 (1997) 458–474.
[23] M. Amadoru, E. Fielt, M. Kowalkiewicz, Organizing visions in the digital world: The case of the blockchain discourse on Twitter, in: Proceedings of the 42nd International Conference on Information Systems, Association for Information Systems, 2021.
[24] Harvard Business Review Analytic Services, BizOps: Connecting IT to Business Outcomes, Harvard Business Review (2020). https://hbr.org/sponsored/2020/06/bizops-connecting-it-to-business-outcomes (accessed July 20, 2023).
[25] D. Shepherd, R. Suddaby, Theory building: A review and integration, Journal of Management 43 (2017) 59–86.
[26] J. Creswell, Research design: Qualitative, quantitative, and mixed methods approaches, 3rd ed., SAGE Publications, 2009.
[27] J. Wolfswinkel, E. Furtmueller, C. Wilderom, Using grounded theory as a method for rigorously reviewing literature, European Journal of Information Systems 22 (2013) 45–55.
[28] J. Webster, R. Watson, Analyzing the past to prepare for the future: Writing a literature review, MIS Quarterly 26 (2002) xiii–xxiii.
[29] I. Dey, Qualitative Data Analysis A User-Friendly Guide for Social Scientist, Routledge, London and New York, n.d.
[30] M. Miles, A. Huberman, J. Saldaña, Qualitative Data Analysis: A Methods Sourcebook, Sage Publications, Thousand Oaks, CA, 2014.

[31] W. Fernandez, H. Lehmann, Case studies and grounded theory method in information systems research: Issues and use, Journal of Information Technology Case and Application Research 13 (2011) 4–15.
[32] B. Glaser, A. Strauss, Grounded theory: The discovery of grounded theory, Sociology the Journal of the British Sociological Association 12 (1967) 27–49.
[33] H. Zhang, X. Zhou, X. Huang, H. Huang, M.A. Babar, An evidence-based inquiry into the use of grey literature in software engineering, in: 2020 IEEE/ACM 42nd International Conference on Software Engineering, IEEE, 2020: pp. 1422–1434.
[34] R. Adams, P. Smart, A. Huff, Shades of grey: guidelines for working with the grey literature in systematic reviews for management and organizational studies, International Journal of Management Reviews 19 (2017) 432–454.
[35] G. Paré, M. Trudel, M. Jaana, S. Kitsiou, Synthesizing information systems knowledge: A typology of literature reviews, Information & Management 52 (2015) 183–199.
[36] A. Schwarz, M. Mehta, N. Johnson, W. Chin, Understanding frameworks and reviews: a commentary to assist us in moving our field forward by analyzing our past, ACM SIGMIS Database: The DATABASE for Advances in Information Systems 38 (2007) 29–50.
[37] A. Wiedemann, M. Wiesche, H. Gewald, H. Krcmar, Implementing the Planning Process within DevOps Teams to Achieve Continuous Innovation, Hawaii International Conference on System Sciences (2019). https://aisel.aisnet.org/hicss-52/st/agile_development/9.
[38] P. Forbrig, Use Cases, User Stories and BizDevOps., in: REFSQ Workshops, 2018.
[39] Gartner, Demystifying XOps: From DataOps to ModelOps and Platform Ops for AI, Gartner, 2021.
[40] A. Hern, The two-pizza rule and the secret of Amazon's success, The Guardian 24 (2018).
[41] J. Singh, S. Nambisan, R. Bridge, J. Brock, One-voice strategy for customer engagement, Journal of Service Research 24 (2021) 42–65.
[42] T. Sedano, P. Ralph, C. Péraire, The product backlog, in: 2019 IEEE/ACM 41st International Conference on Software Engineering, IEEE, 2019: pp. 200–211.
[43] J. Ribeiro, R. Lima, T. Eckhardt, S. Paiva, Robotic process automation and artificial intelligence in industry 4.0–a literature review, Procedia Computer Science 181 (2021) 51–58.
[44] M. Jiménez, L. Rivera, N. Villegas, G. Tamura, H. Müller, P. Gallego, DevOps' shift-left in practice: An industrial case of application, in: International Workshop on Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment, Springer, 2018: pp. 205–220.
[45] H. Jernberg, P. Runeson, E. Engström, Getting Started with Chaos Engineering-design of an implementation framework in practice, in: Proceedings of the 14th ACM/IEEE International Symposium on Empirical Software Engineering and Measurement, 2020: pp. 1–10.
[46] J. Horsthofer-Rauch, M. Schumann, M. Milde, S. Vernim, G. Reinhart, Digitalized value stream mapping: review and outlook, Procedia CIRP 112 (2022) 244–249.
[47] D. Ravasi, Visualizing our way through theory building, Journal of Management Inquiry 26 (2017) 240–243.
[48] T. Toor, T. Dhir, Benefits of integrated business planning, forecasting, and process management, Business Strategy Series 12 (2011) 275–288.
[49] J. vom Brocke, C. Sonnenberg, Value-orientation in business process management, in: Handbook on Business Process Management 2: Strategic Alignment, Governance, People and Culture, Springer, 2014: pp. 101–132.
[50] C. Wolff, P. Tendyra, C. Wiecher, Agile Systems Engineering in Complex Scenarios, in: 2021 11th IEEE International Conference on Intelligent Data Acquisition and Advanced Computing Systems: Technology and Applications, IEEE, 2021: pp. 323–328.

[51] G. Baltes, I. Leibing, Guerrilla marketing for information services?, New Library World 109 (2008) 46–55.
[52] R. Hoda, N. Salleh, J. Grundy, The Rise and Evolution of Agile Software Development, IEEE Software 35 (2018) 58–63. https://doi.org/10.1109/MS.2018.290111318.
[53] R. Moraga-Díaz, A. Leiva-Araos, J. García, A Robust Statistical Methodology for Measuring Enterprise Agility, Applied Sciences 13 (2023) 8445. https://doi.org/10.3390/app13148445.
[54] C. Ebert, G. Gallardo, J. Hernantes, N. Serrano, DevOps, IEEE Software 33 (2016) 94–100.
[55] G. Li, M. Wei, Everything-as-a-service platform for on-demand virtual enterprises, Information Systems Frontiers 16 (2014) 435–452.

Appendix. Selected Documents and Data Items

Table 1 – BizDevOps definition

Code	Ref.	Data
DF1	[68]	BizDevOps is a **mindset** characterized by agile core values and shared responsibility
DF2	[2]	BizDevOps **strategies** so they can address issues that lead to, or stem from, poor user experiences – through practicing collaboration, ensuring a continuous feedback loop, and developing and achieving user-centric goals
DF3	[3]	Wikipedia has a nice definition for it: It is "a **practice** that emphasizes the collaboration and communication of both software developers and other IT professionals while **automating the process** of software delivery and infrastructure changes. It aims at establishing a culture and environment where building, testing, and releasing software, can happen rapidly, frequently, and more reliably."
DF4	[4]	BizDevOps will bring the **business side**, **developers** and **operations** people to the table at the very start and, unlike what happens today, they will all remain, at that same "table" throughout the entire process
DF5	[5]	BizDevOps is an approach to product development that promotes close collaboration and **shared knowledge** between the business team, developers, and operational team. It ditches the division between those departments to get rid of unnecessary knowledge silos that only disrupt the information flow
DF6	[6]	BizDevOps is about **organizing a short time-to-value**, and it actually reduces risk as it allows things to fail early, and to fail fast—together with the business
DF7	[7]	BizDevOps **bridges operational data with business data** to provide a deeper understanding of how application performance and user experience directly impact business outcomes
DF8	[8]	these days business expects more from project teams than the by-now standard DevOps way of working, the DevOps paradigm focuses on the **more technical** aspects of **delivering value** as a team
DF9	[9]	**DevOps 2.0 or BizDevOps**, DevOps 2.0 is now focused on extending the benefits of feedback to the entire organization (marketing, sales, product, etc.
DF10	[10]	DevOps is evolving to become BizDevOps
DF11	[11]	DevOps 2.0: BizDevOps
DF12	[12]	BizDevOps or DevOps 2.0
DF13	[13]	DevOps 3.0. digital transformation. Orchestrating solutions like RBC Wealth Management's requires a meta level of organization to project management. give app production teams **practical tools** that deliver organizational value. value stream maps
DF14	[14]	Features of BizDevOps. Click Funnels. Landing Pages
DF15	[15]	**XOps** has emerged as the umbrella term for defining a combination of IT disciplines such as DevOps, DevSecOps, AIOps, MLOps, GitOps, and BizDevOps
DF16	[16]	XOps, an **umbrella** name for a collection of IT operational disciplines
DF17	[17]	BizDevOps is a practical way of implementing the **Design Thinking** ethos into your existing DevOps capability
DF18	[18]	BizDevOps is often viewed as way to improve the **value** IT delivers by instilling a shared collaborative mindset between business and IT
DF19	[19]	BizDevOps is ensuring that as technologists we are actually following the original vision of agile – involving the customer early and often in our approach
DF20	[20]	the term feels superfluous to me. The entire reason we do Dev and Ops is to serve a business need, Scrum largely fills the role of the "Biz" in "BizDevOps"
DF22	[21]	Biz are not software engineers nor operations engineers, as these roles are already filled by Dev and Ops respectively. Biz are business analysts and requirements engineers
DF23	[22]	In 2008, Patrick Debois laid the foundations for DevOps at an Agile conference in Toronto. A year later, Paul Hammond and John Allspaw gave a talk at the Velocity '09 conference that highlighted the necessity for cooperation between Dev and Ops. This inspired Debois to coin the term "DevOps" (#DevOps), which quickly picked up momentum (and a fair share of controversy).
DF24	[23]	DevOps 2.0 Technology, process and organization. Every team is **working on their own product**, which is available to other departments and teams as a service via an API.
DF25	[13]	how the company **delivers value** and **delights a customer**

Table 2 – Adoption drivers and goals

Code	Ref.	Data
GO2	[24]	BizDevOps (Business + Development + Operations) is like blockchain: **it's all the rage** in modern business and tech best practice
GO3	[25]	BizDevOps will **emerge big time** (2021)

GO5	[9]	DevOps 2.0 or BizDevOps, DevOps 2.0 is now focused on **extending the benefits** of feedback to the entire organization (marketing, sales, product, etc.
GO6	[26]	most digital start-ups can **release at virtually any time** as needed—weekly, daily, or hourly
GO8	[27]	Real-world services can be up to date, available, and robust **without interruption**
GO10	[24]	**automating** processes that don't need human thinking or creativity
GO12	[27]	unconditional **focus on customer benefits**
GO13	[28]	More businesses are also likely to adopt BizDevOps practices thanks to the faster **real-time analytics**
GO14	[29]	business becomes more **technologically intensive**
GO15	[30]	**embrace change** and evolution as key design principles for organizational operating models
GO16	[31]	A powerful BizDevOps practice shifts Agile product thinking from the success of the 'software feature' to the success of the entire system. We use our approach to add another set of system requirements (SRs) on top of the list of software feature requirements (FRs). The objective is to **automatically provide everything** needed to handle a new feature in a production system.
GO17	[32]	**Imagine** a senior executive typing a sudden idea of a feature while commuting by a metro in Chennai and before they reach office, the feature has been tested, deployed for approval, approved and the development team sitting in Ukraine (because of the closeness of the time zones) have started implementing the feature is under way
GO18	[33]	**Continuous quality** for enterprise applications can only be achieved by taking a BizDevOps approach
GO19	[13]	The **demand** for a "DevOps Engineer" that assembles the FOSS bits and pieces needed for a solution is stronger than ever
GO20	[3]	**flexibility and freedom** can lead to innovative solutions
GO21	[4]	**dramatically change the way software is imagined**, developed and released
GO22	[34]	**being ready** for a change
GO23	[7]	**key to success** for software-driven businesses **is delivering value**
GO24	[8]	DevOps paradigm focuses on the more **technical aspects** of delivering value as a team

Table 3 – Tackled problems

TP1	[9]	**end users** are becoming more and **more demanding.** users wi th greater technological knowledge and greater expectations
TP2	[8]	fulfill our customers' **more functional, less technical needs**
TP3	[7]	not to **fear failure**
TP4	[27]	Real-world services can be up to date, available, and robust **without interruption**
TP5	[24]	reliance on legacy technologies that require **manual orchestration**, by engineers like Barry and Jenny, and thus cause high operating costs. disaggregated IT landscape
TP6	[3]	a **platform** that delivers structure and up-to-date information on discussions, plans and decisions
TP7	[3]	**approval cascades** create too much overhead
TP8	[34]	**common ground in between**. The **metric** that would allow the DevOps team to measure the real-life effects of their work and business stakeholders to realize how the work of an IT team actually influences business
TP9	[26]	the **pressure to deliver great customer experiences** while spending money wisely has led a number of agile companies to adopt the "Strangler pattern." This approach involves selecting the most frequently changing functionalities (such as loan-origination journeys, product catalogs or tariff modules, scoring engines, data models, or customer-facing journeys), assigning ownership for these functionalities to business or platform tribes, and setting up dedicated BizDevOps teams to create granular and specialized services (often called microservices).

Table 4 – Cultural changes

Code	Ref.	Data
CU1	[35]	In a typical BizDevOps environment, the business, development, and operations teams analyze the business problem, **collectively** evaluate the business value created by each requirement, and prioritize accordingly. This not only allows the business to have better **control** over the changes but also provides **more say** to the development and operations teams who get to see the business value created by their code
CU2	[36]	one of the biggest challenges in undertaking the shift to BizDevOps is **finding common terminology and understanding** among the teams and bridging the divide between business stakeholders and developer teams
CU3	[24]	In a cultural sense, BizDevOps is the **shared understanding, responsibility and collaboration** between business, software development and technology operations team members.
CU4	[24]	**humans at the centre** of the approach
CU5	[8]	**Crossing over** to the other side. true teammates

CU6	[37]	Steps that evolve culture by promoting structures that are **not hierarchical** but instead **flat** and empowered, focusing on teams and people instead of projects, and preventing us against them mentalities or **command and control** management styles will be what truly enables BizDevOps adoption
CU7	[19]	BizDevOps is Needed to Break **Siloes** Preventing Success
CU8	[2]	breaking down disruptive **silos** within organisations and end-user data can be disseminated **immediately** across the business, optimising performance and improving efficiency
CU9		by incorporating Business Operations into the DevOps loop, your **ability to react** to the **changing needs** of the customer improves dramatically as the **silo mentality** that often exists between the business and technology teams becomes less significant
CU10	[17, 38]	**situational awareness** and **shared understanding** and commitment, moving from "**command and control**" and traditional hierarchic management models to new leadership styles and behaviors
CU12	[39]	**shared language** of service-level objectives (SLOs)
CU13	[7]	Business transactions are the **common language** that brings DevOps and business teams into productive collaboration. A business transaction is the interaction between a business and its customers, vendors, partners or employees that provides a desired outcome of mutual benefit
CU14	[36]	Another part of forming the **common language** among BizDevOps participants revolves around metadata
CU15	[38]	making the shift from "features" to "**outcomes**" and from a flow of "code-to-commit to a flow of "idea-to-value."
CU16	[6]	The BizDevOps view of work culture requires that you continually review past results, are ready for **change**, and agree on whether collaboration and product creation are moving in the right direction
CU17	[38]	Teams need to be able to respond to **change**
CU18	[40]	**bridging the gap** boils down to things like vulnerability, **honesty** and **transparency**. Getting out of your comfort zone. Genuinely trying to **understand the other person**. When such an atmosphere is established, there's no finger pointing—and the other party praises you for the partnership you've created together
CU19	[36]	giving business process professionals, enterprise architects, IT teams and developers greater **understanding** of how software changes and deployments affect the business
CU20	[34]	BizDevOps is not a trend. The approach grew from the **real need** development teams had — the need for integration between business and tech that's key in avoiding costly development failures and building products people truly seek
CU21	[41]	It's in our DNA. to incorporate **business priorities** in all projects' development and operations
CU22	[41]	It takes people committing to **honesty and vulnerability** to genuinely understand the other side

Table 5 – Adoption barriers

Code	Ref.	Data
BA2	[10]	few business stakeholders **understand** application software code
BA3	[10]	**language** of developers
BA8	[42]	You may have stringent **compliance requirements** and extremely low tolerance to risk. You will need to know how to manage the needs of a complex set of stakeholders, including end users, business analysts, project and program manager, enterprise architects, and more
BA9	[22]	any time you grant non-technical team members access to any aspect of your application, there will be some inherent **risk**. Hence, one of the main purposes of DevOps 2.0 is to mitigate that risk through proper checks, permissions, and unencumbered collaboration
BA12	[26]	business often throws an endless string of new requirements over the fence that IT doesn't have capacity to deliver, let alone manage the corresponding **technical debt**
BA13	[30]	we need to have a more **holistic** view and co-design of the organization structures and technical architecture
BA14	[22]	It has become an **amalgamation** of soft and hard skills: trust, cross-functional teams
BA15	[43]	structured their IT portfolio around **products and value streams**, allowing at portfolio level for minimum governance and maximum synchronization and autonomy

Table 6 - Organizational Changes

Code	Ref.	Data
OC1	[18]	organizations should be striving to "**Be BizDevOps**" rather than simply "Doing BizDevOps"
OC2	[43]	**100% "BizDevOps".** 20% of the development teams were leveraging agile and 80% were still waterfall. It became apparent that having two different ways of working and collaborating within IT meant for BMW having two **different speeds and cultures**. Teams on a two-week sprint were delayed and impeded by the waterfall teams still working towards annual releases

OC3	[43]	structured their IT portfolio around products and **value streams**, allowing at portfolio level for minimum governance and maximum synchronization and autonomy
OC4	[27]	Bonifaz Maag, managing partner of Kugler Maag Cie, puts it this way: “Digital services depend on **self-determination**; these specialists need the freedom to act independently.
OC5	[30]	we need to have a more **holistic view** and co-design of the organization structures and technical architecture
OC6	[26]	the pressure to deliver great customer experiences while spending money wisely has led a number of agile companies to adopt the “Strangler pattern.” This approach **involves selecting the most frequently changing functionalities** (such as loan-origination journeys, product catalogs or tariff modules, scoring engines, data models, or customer-facing journeys), **assigning ownership** for these functionalities to business or platform tribes, and setting up dedicated BizDevOps teams to create granular and specialized services (often called microservices).
OC7	[44]	EverythingOps, many competing ways to do one thing, Different groups **pulling in various directions** create an ongoing battle of EverythingOps, FinOps is emerging to control spending, ITOps is about delivering services, DevOps is improving release fluidity and DevSecOps aims to bring **security** to the forefront of the release process. Lastly, BizDevOps is about increasing observability for business outcomes
OC8	[43]	increase in **release frequency** that went from 12 per year to two per month, and they saw a significant decrease in defects or in time to resolution
OC9	[23]	service-oriented organizations, where each team develops and manages their **own service** end-to-end from development to production
OC10	[26]	companies that embrace enterprise agility cannot lean too hard on **vendors** and **partners** to provide turnkey IT services
OC11	[26]	An international telecom company internalized hundreds of engineers, mostly by **insourcing**
OC12	[45]	Differentiating engineering capabilities should be **reshored and built in-house**. Having engineers close to the frontlines improves time to value
OC13	[26]	Transform the core IT landscape by distributing IT systems to 'teams of teams and gradually replacing them by granular services
OC14	[26]	one bank was able to make its monolith core banking systems leaner by approximately 35 percent by **separating noncore functions** into a microservice layer or specialized applications
OC15	[40]	The **hierarchical structure** of many organizations doesn’t help. It creates a comfort zone that discourages transparency and vulnerability
OC16	[26]	creating a diamond-shaped talent composition. increasing the **share** of coders from around 10 percent to 80 percent
OC17	[3]	**Decisions** about new technologies or frameworks are now taken by the team, taking risks and benefits into consideration
OC18	[46]	You've got to **move hierarchy to community**. Command and control is dead. If you think you can maintain that you will not retain any great people
OC19	[44]	some things can get out of control , built-in **guardrails** for provisioning tools are necessary, We have too many tools—we need better frameworks to tie this together
OC20	[9]	techniques that helps to decouple the software delivery of new functionalities. In other words, it is about making these new functionalities available to the end user based on **business rules** instead of the operation teams, Flag Driven Development, we do not make available the new functionality to 100 percent of users. It will be delivery in stages, starting with 1 percent of users, then with 10 percent, then with 30 percent, etc. with the ability to quickly enable or disable the functionality when something does not work as expected
OC21	[27]	In contrast to a product manufacturer with clearly defined departments in a classic vertical structure, digital services call for workflows that **prioritize** speedy and interdisciplinary communication and **decision-making**

Table 7 - Teams

Code	Ref.	Data
TE1	[27]	**transition** away from completing different project tasks on a tight schedule towards working **collaboratively** to maintain and further develop a service for its entire use cycle
TE2	[40]	So the challenge is to find people from IT and the business who can operate in the **purple shaded area**
TE3	[12]	The **development team** is composed of business analysts and professional developers
TE4	[6]	The **role** of a business team **broadens** and evolves from specifying requirements to closely collaborating with development and operations teams. The whole team regularly evaluates risks and seeks opportunities with the ultimate goal to modify the product’s vision and adapt it to users’ needs even more accurately
TE6	[26]	“BizDevOps” teams of **five to nine people** that have all the required skills to deliver a mission: business, developing and testing, and site reliability engineering. Business team members include product owners, product experts, and customer experience experts who drive product needs based on the voice of the customer and ROI.

		Engineers drive production of shippable software on a daily basis, as well as automation to release and operate reliably in production
TE7	[8]	find people from IT and business who can operate in this bridged **middle ground** area
TE8	[45]	Software product engineers, for example, need to be close to **frontline** workers, working day to day with them to build and deploy leading algorithms quickly
TE9	[30]	we have "**structural enabling teams**" in the form of product managers, engineering managers and tech leads, who look at the different aspects of the sociotechnical architecture (product, people and tech architecture).
TE10	[35]	In a typical BizDevOps environment, the business, development, and operations teams analyze the business problem, **collectively** evaluate the business value created by each requirement, and prioritize accordingly
TE11	[31]	customer business experts **work closely with** our IT experts using a Design Thinking approach
TE12	[26]	teams that ladder up into "teams of teams" known as "**tribes**." segment tribes bundle products for specific business segments and support commercial activities, while product tribes develop product features and product-specific customer journeys
TE13	[26]	To counterbalance the autonomy of the segment and **product tribes** and to preserve architectural consistency and IT cost efficiency, companies also establish **platform tribes** that deliver common services, providing reusable components to facilitate the work of engineers in business tribes
TE14	[26]	To achieve a balance, companies can ensure each tribe has both a **business lead** ("mini CEO") and an **IT lead** ("mini CIO"). Often, the business-tribe leads report to the head of business (typically an executive committee member such as the chief commercial officer), and the IT leads report to the CIO, ensuring a level of control and accountability by the CIO.
TE17	[27]	team is a **microcompany**
TE18	[27]	A service team with BizDevOps capabilities, by contrast, does its work in a comprehensive, independent, and accountable way. Such a team is a **microcompany**, so to speak, within the larger corporate structure
TE19	[31]	the **business team** sets requirements and works directly with developers to establish priorities for Agile software development product backlogs
TE20	[29]	**2 pizza team**
TE21	[47]	squads of max. **6 to 9 people**
TE22	[48]	We held **regular breakdown sessions** that put team members, both analysts and developers, into the same room, **often with business owners, where they made prioritized decisions based on business requirements**. With this approach, all of the team members had a shared understanding of the business needs and purpose of the solution with the product owner presenting a clear pathway for what needed to be built and maintained
TE23	[8]	Getting in a room and **shutting the door**. putting the people with the right knowledge, expertise, vision, passion, and mandate together, getting the whole system in the room
TE24	[38]	need to define, agree and commit to "**desired behaviors**" and what is effective **collaboration**
TE25	[49]	gather user research and a **hypothesis** from it, introduce it into the application, and quickly get it in front of users with real-time measurement and telemetry
TE26	[26]	**Daily interaction** allows the team to reduce requirements alignment time from months to days or even hours, radically reducing time to market and the need for communicating through bureaucracy
TE27	[40]	Getting into the **same room** from Day One creates an atmosphere of trust and transparency, which helps us realise short time-to-value together
TE28	[40]	BizDevOps not only means getting together during the start or design of a project: it also means **getting together** during the **run phase**. Sit behind the desk of end users. Feel what they are experiencing when they have to wait five seconds during each and every login
TE29	[26]	Business team members include product owners, product experts, and customer experience experts who drive product needs based on the voice of the customer and ROI
TE30	[8]	there's not much **hierarchy**. It's all about moving unnecessary management and overhead out of the way and putting experts in the lead
TE31	[31]	A powerful BizDevOps practice shifts Agile product **thinking** from the success of the 'software feature' to the success of the **entire system**. We use our approach to add another set of system requirements (SRs) on top of the list of software **feature requirements** (FRs). The objective is to automatically provide everything needed to handle a new feature in a production system.
TE32	[12]	Integrated requirement management. The business provides their **requirements** and feedback on the live app (minimal viable product) through a user-friendly feedback mechanism
TE33	[18]	Requirements are a **team sport** and management and stakeholders must be committed to building a culture that fosters this behavior
TE34	[18]	Understand a requirement's actual scope and risks by incorporating and integrating the appropriate roles and teams through facilitated and coached backlog refinement and planning ceremonies/activities

TE35	[40]	Prepare. Well begun is half the work. This phase typically starts before the pressure cooker starts and is performed by the more solution- and/or technically oriented team members. With BizDevOps, it all starts with a **business need**. Within the team, the business defines that need in the form of requirements, which should be detailed and refined enough for the technical members of the team to plan and build them
TE36	[40]	**Ideate**. This phase is where the business takes the stage, and shares their knowledge, experience, frustrations, wishes, ideas. IT is listening, in an emphatic way, trying to ask smart questions
TE37	[40]	**Prototype**. This is where the magic happens. Based on all the notes, drawings, sketches, and other input from the previous phases, an initial prototype is built
TE40	[50]	At its core, a **senior** DevOps Engineer is looking at any given problem in a holistic manner and trying to understand how this change can be done at an enterprise-scale and not simply trying to solve the given problem one time. Frankly, this skill is less technical and more about evaluating and problem-solving.
TE41	[40]	**Sketch**. After the problem domain has been laid out by the business, it's time for IT to reflect and share how they understood the explanation made by the business. Visualizing this interpretation helps mutual understanding
TE42	[37]	There are a few practices that will help you overcome the wall of confusion separating an IT department from the rest of a business. Define **metrics** that measure business value, and make sure your deployment and release strategies take traditional business concerns, such as geography, community, and other internal and external factors, into account
TE43	[12]	**Visual modeling**. the business analyst is enabled to visually build apps and work together with the professional developer on a common model in a shared environment with ongoing real-time feedback
TE44	[51]	it feels more natural that someone a bit more **senior** is in a ops/devops/architect position
TE47	[24]	**automating** processes that don't need human thinking or creativity
TE48	[26]	In practice, these BizDevOps teams work in parallel to support different areas of the business
TE49	[27]	In contrast to a product manufacturer with clearly defined departments in a classic vertical structure, digital services call for **workflows** that prioritize **speedy** and interdisciplinary **communication** and **decision-making**
TE50	[9]	"**Fast Feedback**" practices
TE51	[52]	Can we provide a certain **feature ad-hoc** to win a new customer?
TE52	[36]	giving business process professionals, enterprise architects, IT teams and developers **greater understanding** of how software changes and deployments affect the business
TE53	[36]	both business and technical users can **see how data flows** through their business processes
TE54	[34]	**delivery manager** supports the team and streamlines the work, but the tasks are prioritized and distributed by the whole team
TE55	[41]	The product owner, also known as a component owner or a value stream owner, basically acts like an **orchestra conductor**, directing the harmony and tempo of business, development, and operations
TE56	[24]	**redesign of** separate product and service **teams** into a team that is multidisciplinary and autonomous by nature
TE57	[26]	responsibility of the CIO—remains to supervise technical debt and the technical quality of delivery and uptime
TE58	[29]	**product-centric roles** such as capability leader, product manager, engineering manager, Agile coach and DevOps architect

Table 8 – Operational patterns

Code	Ref.	Data
BP1	[26]	most digital start-ups can release at virtually any time as needed—**weekly, daily, or hourly**
BP2	[37]	BizDevOps can be seen as a combination of cultural philosophies, practices, and tools that increase an organization's ability to deliver applications and services at **high velocity**
BP3	[46]	What if your customers are giving you real-time feedback and you're actually releasing the code into production in an hour's time, and they're using it? That's agility, that's **speed**. I never thought I would see it in my lifetime, but it's here.
BP5	[27]	In contrast to a product manufacturer with clearly defined departments in a classic vertical structure, digital services call for workflows that prioritize **speedy** and interdisciplinary communication and decision-making
BP6	[30]	we need to continuously **sense** the different parts of the sociotechnical architecture and make sure they are not at "odds" (as Ruth Malan says). This can be achieved in different forms, e.g.: track Accelerate metrics, measure teams cognitive load (or team health), etc. We want to have continuous **feedback** loops to sense the sociotechnical architecture. With this we are continuously learning how the different parts of the system are and with that form an holistic understanding of the system, from which we can drive its evolution
BP7	[39]	enhance remediation and incident **response** efforts
BP8	[27]	Real-world services can be up to date, available, and robust **without interruption**

BP9	[46]	What if your customers are giving you **real-time feedback** and you're actually releasing the code into production in an hour's time, and they're using it? That's agility, that's **speed**. I never thought I would see it in my lifetime, but it's here.
BP10	[53]	They were actually bypassing marketing, sales, communications, risk, finance … they were talking to customers directly. They are getting feedback, **instant feedback**.
BP11	[2]	quickly connects important end-user and customer data into the development **feedback loop**. this increases opportunities for innovation, new revenue growth, and potentially more brand exposure
BP12	[12]	**instant feedback** loop between the business analysts and the developers
BP13	[12]	**feedback loop** of less then a month
BP14	[30]	people interpreting these **feedback loops** at different levels
BP15	[49]	inform yourself in **real time** around what is working and what is not
BP16	[54]	**immediate feedback** on all the new applications, features and services
BP17	[9]	DevOps 2.0 is now focused on extending the benefits of **feedback** to the entire organization (marketing, sales, product, etc.)
BP18	[39]	giving developers feedback about the outcome of their work and real-time **visibility** into their business KPIs, answers at their fingertips to make data-backed decisions that consistently deliver better business outcomes
BP19	[54]	move development and production teams away from nursing applications, new **visibility** for developers to see how their work is being received by users and impacting business value for the organization as a whole
BP20	[35]	implement a real-time **dashboard** of business KPIs that provides a clear indication of the business value delivered with every release
BP21	[9]	one of the major pillars of the DevOps 2.0 approach is the ability to control, through a **control panel** interface, the launch of new features of applications in production environments
BP22	[55]	A key component missing among today's plethora of monitoring tools is genuine **human insight**. Yes, there are tools that alert when exceptions or slowdowns happen, but they don't forge that human connection with the end user
BP23	[35]	BizDevOps also has a significant dependency on tools that give **real-time business metrics**. While there are several APM tools, the focus here is to implement a real-time dashboard of business KPIs that provides a clear indication of the business value delivered with every release. Capgemini's Business Command Center provides a holistic, insight-driven, business-focused application management approach that helps business get a real-time view of value delivered
BP27	[6]	The **process** of implementing BizDevOps should begin with inviting business stakeholders to take part in the development process and discussion about the product vision, goals, and priorities. Your team needs a common goal, a clear process, and mutual KPIs
BP28	[56]	connecting containerization, and continuous integration platforms to create continuous delivery **pipelines** that give new functionality quicker with better quality and less risk
BP29	[57]	Multi-dimensional moments-of-truth for customers
BP30	[5]	**Customer needs** are put at the center. Everyone on the project understands them well, which allows tailoring technical solutions, along with frameworks and methodology
BP31	[58]	Adopting notion of **shifting left** helps to recognize issues earlier
BP32	[2]	Another key component missing among today's plethora of monitoring tools is genuine **human insight**. Yes, there are tools that alert when exceptions or slowdowns happen, but they don't forge that **human connection** with the end user. To this end, it's worth taking a small step back to identify what really matters to customers
BP33	[2]	opportunities for the business beyond tech resources. This happens because employees develop a systems-based approach that has a very real impact on **user experience**
BP34	[8]	stand in the shoes (or sit in the chairs) of **end users**, so everyone can feel what they experience
BP35	[22]	taking the principles of **user-centered** design and applying them to a state of continuous delivery and release. A user-centered deployment, therefore, is a way to frame continuous delivery from the perspective of your product's end-user
BP36	[18]	Your requirements management practice must be in a good state before you adopt BizDevOps. build and foster a culture around collaborative and collective ownership of **requirements** and the delivery artifacts that are created from them
BP37	[18]	many application lifecycle management tools do not have the most appropriate features to manage requirements in a collaborative and holistic way
BP38	[59]	Good requirements and **roadmap** – clear business rationale of the problem IT is requested to solve, which is then used to commit to an achievable delivery plan
BP39	[41]	Our whole team explores the business domain, asking questions to define the **requirements** that the desire demands

BP40	[59]	Post-production support – continuous monitoring of the production environment in order to proactively address risky areas and identify application and system optimization opportunities. Work closely with support staff and end users to expediate the feedback loop
BP41	[59]	Formal and continuous engagements of all participants – bringing business and IT together at the most opportune times to improve delivery effectiveness using the appropriate method for the team, application, and type of work
BP42	[11]	**automation** tools are essential for making the **speed** and agility possible. Performance testing, functional testing and monitoring tools are necessary along the entire software delivery chain to get the data and turnaround needed for an agile environment
BP43	[11]	**Robotic Process Automation** (RPA) where digital software robots perform repetitive tasks across applications to improve business processes execution
BP44	[11]	single platform for **Robotic Automation**. Why don't we fulfill the needs of business, development and operations with a single piece of technology? it also bolsters the efficiency of the continuous delivery process by enabling seamless coordination between teams
BP45	[42]	Business-driven **automation** is the key to executing this approach efficiently and effectively. Most large enterprises have fifty or more enterprise apps for every billion dollars in sales, so if an IT organization aims to achieve continuous deployment, then automation becomes a must
BP46	[11]	automated Performance Testing and Functional Testing to development teams and automated Application Monitoring to operations teams
BP47	[56]	increase automation, mainly in **testing** and quality assurance
BP48	[60]	map their policy prose to automation. A system that runs continuously across the entire organization and software delivery lifecycle (SDLC), including production, comparing the digital estate against those policies
BP49	[61]	business working across the whole life cycle, ability to do course correction and steering during the lifetime of a project
BP50	[26]	achieving missions with as few **handovers** as possible
BP51	[37]	BizDevOps is accomplished by encouraging the business team to work directly with product owners, developers, and operators to set **priorities** for sprints and backlogs. **Collaboration** with the business team is encouraged throughout the entire release cycle
BP52	[3]	The consequence is that Business should be tightly integrated into the DevOps team. Do we need this new feature or shall we move the button from left to right, do we need to change the way a user is searching? Can we provide a certain feature ad-hoc to win a new customer? How does the downtime of an application or server affect the company bottom-line? Why does the conversion rate go down?
BP53	[3]	In the past the Business was reduced to create functional and non-functional requirements, which are translated into source code by Development and operated on a standardized environment by operations. A throw over the fence culture with a lot of ping pong processes of who is right and who is wrong. But in the above defined environment where you have to react in seconds, minutes or days, streamlined processes and defined communication and approval cascades create too much overhead and detract the people from focusing on what really needs to be done
BP54	[60]	"**Shift left**" as a best practice for catching code issues earlier in the development cycle. shift left needs to be reimagined with a new mindset, a new approach, and some innovative automation to deliver on the promise
BP55	[45]	pull data quickly from myriad sources and combine them in new and creative ways
BP56	[45]	Fundamental to this setup is a DDP "reference architecture." This architecture separates data from core transactional systems, the DDP approach puts data in the hands of the business, business and technology teams to combine internal and external data to gain advantage, and then continue with incremental builds and delivery. Modularity facilitates rapid use of blended data. All components within the DDP work together using APIs
BP57	[45]	Data Governance and Management, which **data assets** exist today and which critical datasets should be **owned or acquired for advantage**
BP58	[62]	Data Management is often missing from the DevOps picture. safety vs flexibility tradeoff
BP59	[45]	blend new data science techniques with a deep understanding of business processes and value drivers
BP60	[45]	**data as a service**, DDP makes curated data available as a service across products, and supplies data for use cases to speed up digital initiatives and reduce complexity
BP61	[9]	ability to control, through a **control panel interface**, the launch of new features of applications in production environments. This process would be launched in a controlled way by both technical and non-technical people. Also, the process will be separate from the development and continuous deployment
BP62	[12]	**One-click deployment**. The one-click deployment to any cloud ensures the app can be released in minutes
BP63	[9]	this type of coding techniques will allow to perform **real-time analytics**, making changes to the functionalities of a system that may also impact in the application performance monitoring (APM) tools

BP64	[9]	making changes to the functionalities of a system that may also impact in the **application performance monitoring** (APM) tools
BP65	[63]	Having an executable, visual and understandable model has benefits for business stakeholders, developers and operators. It also improves the communication and collaboration between them big time.
BP66	[9]	Using this simple best practice for development, called **Flag Driven Development**, we do not make available the new functionality to 100 percent of users. It will be delivery in stages, starting with 1 percent of users, then with 10 percent, then with 30 percent, etc.
BP67	[22]	**feature rollout** will be decoupled from code deployment, non-technical team members would be able to control the visibility of particular features without compromising the app's integrity. A major cornerstone of DevOps 2.0 is the ability to control feature releases independently from your code deployments. Designers can conduct user testing by toggling experimental features on and off for test users
BP68	[22]	If we launch a feature and no one likes it, then we can instantly roll it back
BP69	[11]	application automation is an integral part of your customer experience. application automation should be treated with the same weight as the all the application artifacts being changed and tested on a continuous basis
BP70	[56]	The need for quick delivery is also expected to **automate** some domains of routine code development
BP71	[64]	increasing adoption of microservices architecture and **shift-left**
BP72	[45]	put data into action and to build strategic assets, combine the data (i.e., to do something creative with the data)
BP73	[45]	liberate data, an approach that prioritizes data speed, agility, and faster learning for competitive advantage. This new approach, which we refer to as data and digital platforms (DDP), decouples digital business transformation from core IT transformation. It creates a data layer to liberate data from core systems that are scattered across the enterprise
BP74	[59]	61% of highly mature DevOps organizations with fully integrated security practices were able to **deploy on demand**
BP75	[26]	Several traditional banks and telecoms in Europe and Asia followed this path and reach as many as **20,000 releases per quarter**, even on back-end systems
BP76	[26]	Getting competent engineers working on autonomous microservices unlocks the true power of continuous integration and continuous delivery (CI/CD). The secret to making this shift lies in **automating tasks** to enable frequent incremental releases
BP77	[62]	DevOps isn't just for teams following an agile or continuous delivery lifecycle but is potentially applicable to any team following a **lifecycle** that supports incremental delivery
BP78	[4]	What do low-code/no-code tools have to do with BizDevOps? The introduction of these platforms is **bringing development to the business side**
BP79	[59]	**High fidelity environments** – non-production environments (e.g. development, testing, staging) mirror production in order to generate and test features and changes against realistic conditions
BP80	[8]	business requirements (let alone desired outcomes) tend to get **lost in translation**
BP81	[23]	**Auto-detect** when key business transactions aren't working as expected
BP82	[65]	**full-stack monitoring**
BP83	[23]	**Track** revenue, conversion rates, availability, user experience, drop-off rates and other relevant metrics
BP84	[45]	**liberate data**, an approach that prioritizes data speed, agility, and faster learning for competitive advantage. This new approach, which we refer to as data and digital platforms (DDP), decouples digital business transformation from core IT transformation. It creates a data layer to liberate data from core systems that are scattered across the enterprise
BP85	[36]	both business and technical users can **see how data flows** through their business processes
BP86	[36]	"With **'drag and drop' approaches**, these tools enable developers to easily create new digital solutions, integrations and automations by reusing existing IT capabilities," says Dorato. "The composable enterprise strategy that emerges from this eliminates the need for developers to write every line of code
BP87	[34]	**react in seconds** to every influence. very short time-to-market. In the extreme the deployment of an eCommerce Website could happen every minute or hour and every day
BP88	[34]	backlog with **only business features** rather than IT features
BP89	[7]	**Unified transaction monitoring** with big data scalability and management is the only way IT and business owners can ensure that end-to-end user experience and business objectives are being met. It drives customer satisfaction and improves competitiveness, strengthening financial performance and market valuation
BP90	[9]	**functionalities available to the end user based on business rules**. using "flags," when to release or enable a new functionality to the end user. It will allow us to deploy new functionalities to the production environment more frequently without enabling them to the end user. This removes the "fear" generated by the daily deployment processes in production environments. This is what is known, in DevOps 2.0, as the rollout
BP91	[9]	determine when a particular functionality is degrading the global system performance and needs to be **disabled quickly**, as well as identify which functionalities are improving the end user experience

BP92	[10]	a developer can sit with an end user to discuss and review functionality, validate assumptions and identify improvements
BP93	[10]	a DevOps engineer who has identified an issue with a production app can work with the developer and business stakeholder to identify and implement a fix that balances both technical and business needs
BP94	[26]	A/B test different versions of the same functionality with different clients, test MVPs any time, incorporate customer feedback at pace, and continually evolve the business, reaching a **true level of agility**
BP95	[39]	BizDevOps teams can use SLOs to enhance **remediation and incident response** efforts
BP96	[22]	rise of **user-centered deployments**

Table 9 – Techniques and tools

Code	Ref.	Data
TO1	[36]	A business **process modelling tool** can detail the tasks, responsible parties, information elements involved in processes, and the interactions which can occur across systems, procedures and organisational hierarchies
TO2	[56]	**Application Performance Management** tools
TO3	[56]	BizDevOps is only possible because of the technology that gives real-time software **analytics** to enterprises
TO4	[9]	making these new functionalities available to the end user based on **business rules** instead of the operation teams
TO5	[23]	tooling to implement the **end-to-end change management** process from requirement management, source code repository, CI server, test harness, continuous deployment infrastructure and others
TO6	[23]	BizDevOps **metrics map**, mapping of IT metrics with business requirements
TO7	[57]	Methods such as **Scrum** and **Kanban** provide valuable tools to keep the focus even in a complex environment
TO8	[66]	**Chaos engineering** is both a process and technology capability. From a technology perspective, the chaos engineering platform should include necessary monitoring, logging, and failure induction tooling. The most common tools include Chaos Toolkit, Gremlin and Simian Army
TO9	[13]	**Value stream mapping** starts at the app production team level. It assesses the way elements within a complex project interact to achieve an operational objective
TO10	[29]	This approach starts with **value stream mapping** of each business capability, and it provides an opportunity to identify white spaces that require greenfield products to optimize the value stream
TO11	[23]	All sign-offs from development leads, test leads, security leads and operations leads are now implemented as executable policies and embedded into the pipeline
TO12	[56]	As business people integrated into the development lifecycle, they need the ability to make changes. **Low-code** platforms help a company build custom applications for a fraction of the time and money
TO13	[29]	using **application portfolio rationalization** (APR) techniques
TO14	[66]	**hypothesis-driven** approach. What is the minimum product that provides the customer with a benefit?
TO15	[66]	Regular **retrospectives** with a focus on action items (measures) for continuous improvement show what has worked in the past – and what has not
TO16	[67]	BUILD AUTOMATION – An **automated code** is prepared to be deployed in a live environment
TO17	[67]	CI/ CD – Continuous Integration and Continuous Deployment deals with frequent merging of codes and unit testing
TO18	[67]	INFRASTRUCTURE AS CODE – This is usually to manage and provision IT infrastructure through code and automation
TO19	[67]	CONFIGURATION MANAGEMENT – This is the process where you can manage and change the state of infrastructure in constant and maintainable ways
TO20	[67]	ORCHESTRATION – Orchestration is nothing but an automation that supports processes and workflows
TO21	[67]	MONITORING – By this method, you collect and present data about the performance and stability of services and infrastructure, while also detecting problems
TO22	[67]	MICROSERVICES – It is basically an architecture that breaks an application into many small and loosely collected services
TO25	[36]	Jenkins and GitHub are the two tools that we use for review and code management in the cloud
TO26	[36]	organisations can use cloud-native CI/CD pipeline tools that work with software deployed in containers and which are managed by container orchestrators like Docker Swarm, OpenShift or Kubernetes
TO27	[36]	In our experience of building observability solutions for 1GB of inserted data per second, building it with technologies like Kafka and ClickHouse is the most efficient option
TO28	[23]	CICD (change management) implements core tooling for build, code review, continuous integration, continuous deployment, test harness, audit trail, management dashboards and others
TO29	[9]	make available **different versions** of a functionality for different types of users (customer segmentation)

TO30	[32]	The commonality in all DevOps tools is in the integration of a CI tool with a SCM repository and with a Cloud storage container to enable Continuous Deployment (CD)
TO31	[23]	Testing in production
TO32	[32]	I found the new testing tools in mocha, chai, Cucumber and jest just ideal to create a tightly integrated flow between Business (Biz), Development, Testing and Operations - BizDevOps
TO33	[9]	use techniques that helps to decouple the software delivery of new functionalities
TO34	[68]	Capturing and delivery **value** is about running the backlog and sprints and ensuring delivery until the end users have access
TO35	[69]	**Living documentation**, the business is responsible for the documentation of these business interfaces. Process, use cases, data, events, business requirements
TO36	[36]	Metadata management tools
TO37	[36]	"With 'drag and drop' approaches, these tools enable developers to easily create new digital solutions, integrations and automations by reusing existing IT capabilities," says Dorato. "The composable enterprise strategy that emerges from this eliminates the need for developers to write every line of code
TO38	[22]	tools specifically designed to **coordinate the skillsets of both developers and non-developers**
TO39	[49]	**connecting** AppDynamics (**application performance management**) to the tools that are below it in the stack so that each team uses a tool that is comfortable but the data sets are connected. "Instead of finger-pointing when something goes wrong or [teams] need to optimize, they're able to actually have a shared source of truth,". "That makes it easier for them to collaborate in this closed-loop operating model."

Table 10 – Challenges and risks

Code	Ref.	Data
CH1	[59]	Delivery tool vendors and delivery practice thought leaders attempt to clarify BizDevOps as a set of collaborative activities and/or as an integrated pipeline, but they stop short of providing consumable and achievable implementation guidance. The overwhelming and **intimidating** list of suggested initiatives (or lack thereof) create a perceived barrier to entry and confusion about where to start
CH2	[36]	**too many tools** which essentially do the same job as each other
CH3	[30]	**Cognitive load**. the product became too complex to be owned by the team (i.e.: team is reaching its maximal cognitive load)
CH4	[42]	But what if you are a **large company** running dozens, even hundreds of applications? What if you leverage an increasing number of cloud-based packaged applications, where you can't control when and how these applications change?
CH5	[70]	The upfront **planning** aspects of the workflow continue to exist in a vacuum, with limited ability to pivot based on ongoing market trends
CH6	[33]	You may have stringent compliance requirements and extremely low tolerance to risk. You will need to know how to manage the needs of a **complex set of stakeholders**, including end users, business analysts, project and program manager, enterprise architects, and more
CH7	[30]	When you combine that with another common trait of striving for "**fixed org structures**", i.e.: neglecting that sociotechnical systems are in continuous change, this becomes an even bigger challenge
CH9	[21]	My concerns with BizDevOps are that it implies the inclusion of 'The Business' in the product delivery process, breaking down the silos of Business and IT. Instead, BizDevOps breaks down the silos of requirements, development, and operations. Unfortunately, **it does not even bridge the gap** between Business and IT and therefore does not improve the alignment between Business and IT either
CH10	[71]	**Doing DevOps the wrong way** has become industry standard, which means of course the reality of devops is divorced from the theory of devops, meaning all of our jobs are that much harder
CH11	[45]	**Large programs, re-platforming, and complex replacements** are interesting for systems integrators but they take many years, cost more than most companies can afford, pose risks, and are highly unlikely to deliver on the promise
CH12	[9]	reducing the risk associated to each new delivery of functionality
CH13	[4]	In a BizDevOps world, software developers will have clearer priorities and shorter backlogs, but **fewer opportunities for creativity** and autonomy
CH14	[34]	the pieces added for Biz: Adapt, Align, Define, and Approve, are pretty **redundant**

Table 11 – Wide implementation

IM1	[10]	a DevOps engineer who has identified an issue with a production app can work with the developer and business stakeholder to identify and implement a fix that **balances both technical and business needs**

IM2	[10]	developing digital applications to **relentlessly streamline and automate internal operations** too, combining the speed of DevOps with business objective alignment by putting custom software development at the core of your business to achieve a competitive advantage for your organisation
IM3	[11]	Why don't we fulfill the needs of business, development and operations with a **single piece of technology**?
IM4	[25]	**everything is** pretty **standardized** almost like the mainframe era
IM5	[26]	companies also establish platform tribes that deliver **common services**, providing reusable components to facilitate the work of engineers in business tribes. Examples include cybersecurity-as-a-service, infrastructure-as-a-service, and data-as-a-service tribes that provide automated self-service tools, as well as core IT tribes that hold complex legacy systems that span multiple tribes and can't (yet) be distributed
IM6	[45]	if the digital initiatives are going to **take too long**, they should be reconsidered and possibly abandoned
IM7	[27]	unconditional **focus on customer benefits**
IM8	[30]	**embrace change** and evolution as key design principles for organizational operating models
IM9	[35]	adopt a **product-thinking** approach over a traditional project-thinking approach, it puts all the more focus on the active involvement of business in prioritizing and strategizing the transformation roadmap
IM10	[37]	Steps that evolve culture by promoting structures that are **not hierarchical** but instead **flat** and empowered, focusing on teams and people instead of projects, and preventing us against them mentalities or **command and control** management styles will be what truly enables BizDevOps adoption
IM11	[36]	not creating software that doesn't **align with business processes** – or worse, creating solutions for problems that don't exist
IM12	[27]	Bonifaz Maag, managing partner of Kugler Maag Cie, puts it this way: "Digital services depend on **self-determination**; these specialists need the freedom to act independently.
IM13	[26]	one bank was able to make its monolith core banking systems leaner by approximately 35 percent by **separating noncore functions** into a microservice layer or specialized applications
IM14	[30]	we need to continuously **sense** the different parts of the sociotechnical architecture and make sure they are not at "odds" (as Ruth Malan says). This can be achieved in different forms, e.g.: track Accelerate metrics, measure teams cognitive load (or team health), etc. We want to have continuous **feedback** loops to sense the sociotechnical architecture. With this we are continuously learning how the different parts of the system are and with that form an holistic understanding of the system, from which we can drive its evolution
IM15	[6]	The **process** of implementing BizDevOps should begin with inviting business stakeholders to take part in the development process and discussion about the product vision, goals, and priorities. Your team needs a common goal, a clear process, and mutual KPIs
IM16	[18]	Your requirements management practice must be in a good state before you adopt BizDevOps. build and foster a culture around collaborative and collective ownership of **requirements** and the delivery artifacts that are created from them
IM17	[26]	achieving missions with as few **handovers** as possible

Table 12 – Open issues

OI1	[4]	low-code/no-code development **platforms may also erode the "idea" of developer** as time passes
OI2	[25]	The intersection between **Observability**, Performance Testing, and Resilience Testing will become mainstream
OI3	[30]	**sociotechnical evolution**. sociotechnical architecture is: "taking an holistic co-design approach to technical and organizational systems. continuously sense the different parts of the sociotechnical architecture**. Systems thinking** and sociotechnical architecture are topics that are not commonly used
OI4	[13]	The **demand** for a "DevOps Engineer" that assembles the FOSS bits and pieces needed for a solution is stronger than ever

References

[1] EFS Consulting, BizDevOps, EFS Consulting. (2023). https://efs.consulting/en/bizdevops/ (accessed January 16, 2023).

[2] INDVSTRVS, BizDevOps Breaks Down Silos, Improves User Experience, INDVSTRVS. (2019). https://indvstrvs.org/bizdevops-breaks-down-silos-improves-user-experience/ (accessed January 16, 2023).

[3] R. Reinicke, What is Biz, Dev and Ops or BizDevOps?, LeanIX. (2016). https://www.leanix.net/en/blog/what-is-biz-dev-and-ops-or-bizdevops (accessed January 16, 2023).

[4] TechTarget, Why -- and how -- BizDevOps is going to change everything, TechTarget. (2016). https://www.techtarget.com/searchsoftwarequality/essentialguide/Why-and-how-BizDevOps-is-going-to-change-everything (accessed January 16, 2023).

[5] Brainhub, BizDevOps - a Bridge Between Business and Tech, (n.d.). https://brainhub.eu/library/bizdevops-in-nutshell (accessed January 16, 2023).

[6] Brainhub, BizDevOps (DevOps 2.0) Is the New Iteration of DevOps, HackerNoon. (2022). https://hackernoon.com/bizdevops-devops-20-is-the-new-iteration-of-devops (accessed January 16, 2023).
[7] G. Lowy, Delivering Value with BizDevOps, DEVOPSdigest. (2015). https://www.devopsdigest.com/delivering-value-with-bizdevops (accessed January 16, 2023).
[8] H. van der Schuur, BizDevOps: The way to bring technology to the heart of business (and…, Schuberg Philis. (2021). https://schubergphilis.com/en/stories/bizdevops-the-way-to-bring-technology-to-the-heart-of-business-and-build-bridges-along-the-way (accessed January 16, 2023).
[9] Contributor, DevOps 2.0 for Digital Transformation, DevOps.Com. (2017). https://devops.com/devops-2-0-digital-transformation/ (accessed January 16, 2023).
[10] N. Ismail, BizDevOps for business and IT collaboration, Information Age. (2018). https://www.information-age.com/devops-bizdevops-business-10116/ (accessed January 16, 2023).
[11] C. Kummer, Automation for BizDevOps as a Whole. Are you fully ready for the Agile Enterprise?, Robotic Automation Solutions. (2018). https://www.automai.com/automation-for-bizdevops-as-a-whole-are-you-fully-ready-for-the-agile-enterprise-1/ (accessed January 16, 2023).
[12] Mendix, Development Process, Mendix. (n.d.). https://www.mendix.com/evaluation-guide/dev-process/ (accessed January 16, 2023).
[13] V. Keenan, Devops 3.0 Puts Management into Focus, SalesforceDevops.Net. (2021). https://salesforcedevops.net/index.php/2021/09/08/devops-3-0-puts-management-into-focus/ (accessed January 16, 2023).
[14] BizDevOps, Supercharge Your Business Efforts with Smooth Development and Operations, BizDevOps. (2021). https://bizdevops.com/about/ (accessed January 16, 2023).
[15] S. Bhattacharyya, XOps has emerged as a popular term–what is it?, Analytics India Magazine. (2022). https://analyticsindiamag.com/xops-has-emerged-as-a-popular-term-what-is-it/ (accessed January 16, 2023).
[16] O. Nath, What's the Next Big Leap in Digital Innovation: Analysts' Take, Spiceworks. (2022). https://www.spiceworks.com/tech/innovation/articles/next-big-leap-in-digital-innovation/ (accessed January 16, 2023).
[17] PMCOE, BizDevOps is Trending!, PMCOE. (n.d.). https://www.pm-coe.com/bizdevops-is-trending/ (accessed September 6, 2023).
[18] A. Kum-Seun, BizDevOps Starts With Great Requirements, SoftwareReviews. (2020). https://www.softwarereviews.com/categories/application-lifecycle-management/research/bizdevops-starts-with-great-requirements (accessed January 17, 2023).
[19] J. Morain, BizDevOps: Solving the Digital Dilemma for Transformational Success, DEVOPSdigest. (2017). https://www.devopsdigest.com/bizdevops-solving-the-digital-dilemma-for-transformational-success (accessed January 17, 2023).
[20] J. Hall, What do I think about BizDevOps?, Jonathan Hall. (2023). https://jhall.io/archive/2021/07/09/what-do-i-think-about-bizdevops/ (accessed September 6, 2023).
[21] Arc-E-Tect, BizDevSecOps, Medium. (2021). https://medium.datadriveninvestor.com/bizdevsecops-did-i-miss-somebody-6948433847c3 (accessed January 17, 2023).
[22] J. Baker, DevOps 2.0, LaunchDarkly. (2016). https://launchdarkly.com/blog/devops2/ (accessed January 17, 2023).
[23] M. Martynov, DevOps 2.0 is here, and it's time to put end-to-end continuous delivery pipelines behind every project., Grid Dynamics. (2019). https://blog.griddynamics.com/the-continuous-delivery-problem-has-been-solved-and-its-time-to-implement/ (accessed January 17, 2023).
[24] A. van der Graaff, V. Sequeira, A. Gatti, Working together: BizDevOps for competitive advantage, PwC. (2019). https://www.pwc.com.au/digitalpulse/bizdevops-competitive-advantage.html (accessed January 16, 2023).
[25] P. Belagatti, DevOps Trends to Anticipate in 2021, DEV Community. (2021). https://dev.to/pavanbelagatti/devops-trends-to-anticipate-in-2021-4cfd (accessed January 16, 2023).
[26] Q. Jadoul, D. Róna, A. Sukharevsky, The five core IT shifts of scaled agile organizations, McKinsey. (2021). https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-five-core-it-shifts-of-scaled-agile-organizations (accessed January 16, 2023).
[27] D. Strube, BizDevOps for digital services, Bosch ConnectedWorld Blog. (2018). https://blog.bosch-si.com/business-models/bizdevops-for-digital-services/ (accessed January 16, 2023).
[28] B. Putano, 6 Software Development Trends for 2020: Developers Needed, Stackify. (2020). https://stackify.com/software-development-trends-2018/ (accessed January 16, 2023).
[29] Cognizant, 17 Must-Do's to Create a Product-Centric IT Organization, Cognizant, 2019.
[30] E. Silva, M. Pais, DevOps is Not Enough for Scaling and Evolving Tech-Driven Organizations: a Q&A with Eduardo da Silva, InfoQ. (2021). https://www.infoq.com/articles/devops-not-enough-scaling-tech-driven-organizations/ (accessed January 16, 2023).

[31] AZUR Group, Using BizDevOps and Microsoft Tools to Develop Cloud Solutions More Efficiently, AZUR Group. (2018). https://www.groupeazur.ca/utiliser-les-outils-bizdevops-et-microsoft-pour-developper-des-solutions-cloud-plus-efficacement/ (accessed January 16, 2023).
[32] Rjv, RJV 's Blog: Automating BDD - CI/CD, BizDevOps, RJV 's Blog. (2019). https://ravichandranjv.blogspot.com/2019/10/automating-bdd-ccd-bizdevops.html (accessed January 16, 2023).
[33] S. Javed, Introducing BizDevOps- Why DevOps doesn't work for enterprise applications, (n.d.). https://www.worksoft.com/corporate-blog/introducing-bizdevops-why-devops-doesnt-work-for-enterprise-applications (accessed January 16, 2023).
[34] M. Dryka, O. Gierszal, M. Warcholinski, What is BizDevOps – when Business meets Technology, (2023). https://brainhub.eu/library/bizdevops-in-nutshell (accessed July 17, 2023).
[35] A. Muchhala, BizDevOps to enable a product-thinking approach, Capgemini India. (2020). https://www.capgemini.com/in-en/insights/expert-perspectives/bizdevops-to-enable-a-product-thinking-approach/ (accessed January 17, 2023).
[36] A. Savvas, The most important DevOps tools for tech workers, Information Age. (2022). https://www.information-age.com/the-most-important-devops-tools-for-tech-workers-20123/ (accessed January 17, 2023).
[37] CloudOps, Everything You Need to Know About BizDevOps, CloudOps. (2019). https://www.cloudops.com/blog/everything-you-need-to-know-about-bizdevops/ (accessed January 17, 2023).
[38] P. Wilkinson, Is Your Organization 'Fit for the Future'?, DevOps.Com. (2019). https://devops.com/is-your-organization-fit-for-the-future/ (accessed January 17, 2023).
[39] A. Grabner, Empowering BizDevOps Teams With SLOs, DevOps.Com. (2021). https://devops.com/empowering-bizdevops-teams-with-slos/ (accessed January 17, 2023).
[40] H. van der Schuur, BizDevOps: Bridging that Dominant Divide between Business and IT, Medium. (2020). https://stories.schubergphilis.com/bizdevops-bridging-that-dominant-divide-between-business-and-it-c1194297a706 (accessed January 17, 2023).
[41] Schuberg Philis, BizDevOps, Schuberg Philis. (n.d.). https://schubergphilis.com/en/how-we-work/bizdevops (accessed January 17, 2023).
[42] S. Javed, Why DevOps Doesn't Work for Enterprise Applications, Dzone.Com. (2020). https://dzone.com/articles/why-devops-doesnt-work-for-enterprise-applications (accessed January 17, 2023).
[43] S. Alvares, DOES 2019: BMW Journey to 100% Agile and BizDevOps Product Portfolio, InfoQ. (2019). https://www.infoq.com/news/2019/11/bmw-devops/ (accessed January 17, 2023).
[44] B. Doerrfeld, How Does IT Management Govern EverythingOps?, DevOps.Com. (2022). https://devops.com/how-does-it-management-govern-everythingops/ (accessed January 17, 2023).
[45] K. Close, A. Gourévitch, M. Schuuring, M. Sterman, L. Quarta, A. Sawadogo, Digital Acceleration Is Just a Dream Without a New Approach to Tech, BCG Global. (2020). https://www.bcg.com/publications/2020/how-to-successfully-accelerate-digital-transformation (accessed January 17, 2023).
[46] P. Wainewright, How IT leads disruptive innovation at banking giant ING, Diginomica. (2016). https://diginomica.com/how-it-leads-disruptive-innovation-at-banking-giant-ing (accessed January 17, 2023).
[47] ING Belgium, (33) Feature Engineer Payments - BizDevOps Engineer, LinkedIn. (n.d.). https://www.linkedin.com/jobs/view/feature-engineer-payments-bizdevops-engineer-at-ing-belgium-3206787770/?originalSubdomain=be (accessed January 17, 2023).
[48] T. Regulski, S. Humes, How CVP Puts the Biz in DevOps, CVP. (2017). https://www.cvpcorp.com/blog/how-cvp-puts-the-biz-in-devops (accessed January 17, 2023).
[49] S. Fregoni, BizDevOps: Gaining a competitive advantage in an app-centric world, SiliconANGLE. (2020). https://siliconangle.com/2020/01/29/bizdevops-gaining-competitive-advantage-app-centric-world-cleur/ (accessed January 17, 2023).
[50] T. Blogumas, Differences Between Junior DevOps and Senior DevOps Engineers, DevOps Dudes. (2020). https://medium.com/devops-dudes/differences-between-junior-devops-and-senior-devops-engineers-8d0f28b8b30b (accessed January 17, 2023).
[51] 8ersgonna8, Starting career in DevOps with no professional tech exp?. I think it will be d…, R/Devops. (2022). www.reddit.com/r/devops/comments/tah72r/starting_career_in_devops_with_no_professional/i0125u6/ (accessed January 17, 2023).
[52] J. Beyer, What is Biz, Dev and Ops or BizDevOps?, (2016). https://www.linkedin.com/pulse/what-biz-dev-ops-bizdevops-j%C3%B6rg-g-beyer/ (accessed September 6, 2023).
[53] L. Chortarias, ING's new way of working through BizDevOps, Business Architecture & Consultancy BAC. (2016). http://www.digitalsocialstrategy.org/bac/2016/12/07/ings-new-way-of-working-through-bizdevops/ (accessed January 17, 2023).
[54] A. Grabner, Digital Transformation Is More About a BizDevOps Culture than Tech, DevOps.Com. (2020). https://devops.com/digital-transformation-is-more-about-a-bizdevops-culture-than-tech/ (accessed January 17, 2023).

[55] M. Yap, User experience done right with BizDevOps, Frontier Enterprise. (2020). https://www.frontier-enterprise.com/user-experience-done-right-with-bizdevops/ (accessed January 17, 2023).
[56] N. Gill, BizDevOps – The Evolution from Agile and DevOps, XENONSTACK. (2020). https://www.xenonstack.com/insights/bizdevops (accessed January 17, 2023).
[57] M. Shome, R. Bose, The next disruption in BFSI: Embracing Industry 4.0 with BizdevOps, (n.d.). https://www.tcs.com/what-we-do/industries/banking/white-paper/adopting-bizdevops-financial-services (accessed January 17, 2023).
[58] jameswanbook, What Is The Future Of DevOps?. Based on the market, TechRepublic. (2018). https://www.techrepublic.com/forums/discussions/what-is-the-future-of-devops/ (accessed January 17, 2023).
[59] A. Kum-Seun, Bridge the Business-IT Chasm With BizDevOps, Info-Tech Research Group. (2020). https://www.infotech.com/software-reviews/research/bridge-the-business-it-chasm-with-bizdevops (accessed January 17, 2023).
[60] T. Johnson, DevOps Has Evolved Beyond Shift Left, CloudBees. (2022). https://www.cloudbees.com/blog/shift-left-done-right (accessed January 17, 2023).
[61] Contributor, Putting the "Biz" in DevOps, DevOps.Com. (2016). https://devops.com/putting-biz-devops/ (accessed January 17, 2023).
[62] Project Management Institute, The Workflow of Disciplined DevOps, PMI. (n.d.). https://www.pmi.org/disciplined-agile/process/disciplined-devops/the-workflow-of-disciplined-devops (accessed January 17, 2023).
[63] B. Rücker, BizDevOps — the true value proposition of workflow engines, Medium. (2018). https://blog.bernd-ruecker.com/bizdevops-the-true-value-proposition-of-workflow-engines-f342509ba8bb (accessed January 17, 2023).
[64] sophieclark, What Is The Future Of DevOps?. Increasing adoption of microservices architecture, TechRepublic. (2019). https://www.techrepublic.com/forums/discussions/what-is-the-future-of-devops/ (accessed January 17, 2023).
[65] Dynatrace, BizDevOps (Business, Development, & Operations), Dynatrace. (n.d.). https://www.dynatrace.com/solutions/bizdevops/ (accessed September 6, 2023).
[66] S. Schulze, J. Tiedemann, DevOps becomes BizDevOps, DevOps Conference. (2018). https://devopscon.io/blog/devops-becomes-bizdevops/ (accessed January 17, 2023).
[67] S. Dutta, An Introduction To DevOps, (2022). https://www.nimblework.com/blog/introduction-to-devops/ (accessed September 6, 2023).
[68] S. Dash, DesOps is "DevOps 2.0," Red Hat Developer. (2018). https://developers.redhat.com/blog/2018/06/27/desops-is-devops-2-0 (accessed September 6, 2023).
[69] A. Geertsema, BizDevOps, aligning business and IT, Medium. (2021). https://arjangeertsema.medium.com/bizdevops-aligning-business-and-it-ea00ada05966 (accessed September 6, 2023).
[70] Blueprint, BizDevOps Enabling Digital Transformation, Blueprint. (2017). https://www.blueprintsys.com/blog/bizdevops-digital-transformation (accessed January 17, 2023).
[71] heavyrain123, DevOps is dead, long live DevOps. I agree with you, R/Devops. (2022). www.reddit.com/r/devops/comments/u0yo1h/devops_is_dead_long_live_devops/i4aucjm/ (accessed January 17, 2023).