Supplementary Material. 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).