A  APPENDIX - Selected Sources

Table 1 - BizDevOps definition

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| DF1 | [68] | BizDevOps is a **mindset** characterized by agile core values and shared responsibility |
| DF2 | [32] | 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 | [61] | 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 | [75] | 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 | [12] | 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 | [11] | 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 | [44] | BizDevOps **bridges operational data with business data** to provide a deeper understanding of how application performance and user experience directly impact business outcomes |
| DF8 | [69] | 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 | [19] | **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 | [34] | DevOps is evolving to become BizDevOps |
| DF11 | [41] | DevOps 2.0: BizDevOps |
| DF12 | [46] | BizDevOps or DevOps 2.0 |
| DF13 | [40] | 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 | [8] | Features of BizDevOps. Click Funnels. Landing Pages |
| DF15 | [7] | **XOps** has emerged as the umbrella term for defining a combination of IT disciplines such as DevOps, DevSecOps, AIOps, MLOps, GitOps, and BizDevOps |
| DF16 | [52] | XOps, an **umbrella** name for a collection of IT operational disciplines |

Table 2 - Hype, drivers and goals

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| GO1 | [55] | It's the next evolution of the "full-stack" engineer. That's whywe are in **high demand** and get paid a lot of money |
| GO2 | [28] | BizDevOps (Business + Development + Operations) is like blockchain: **it’s all the rage** in modern business and tech best practice |
| GO3 | [6] | BizDevOps **will emerge big time** (2021) |
| GO4 | [13] | Literally **billions of dollars** of perceived market value has been pegged to the DevOps term |
| GO5 | [19] | 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 | [35] | most digital start-ups can **release at virtually any time** as needed—weekly, daily, or hourly |
| GO7 | [51] | **minimizes the time** it takes to deliver value to customers |
| GO8 | [74] | Real-world services can be up to date, available, and robust **without interruption** |
| GO9 | [76] | The management in my old company thinks DevOps means: all software engineers **know everything** about development, testing, producing and deploying software |
| GO10 | [28] | **automating processes that don’t need human thinking or creativity** |
| GO12 | [74] | unconditional **focus on customer benefits** |
| GO13 | [58] | More businesses are also likely to adopt BizDevOps practices thanks to the faster **real-time analytics** |
| GO14 | [17] | business becomes more **technologically intensive** |
| GO15 | [72] | **embrace change** and evolution as key design principles for organizational operating models |
| GO16 | [4] | 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 | [62] | 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 | [38] | **Continuous quality** for enterprise applications can only be achieved by taking a BizDevOps approach |

Table 3 - Culture and changing mindsets

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| CU1 | [50] | 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 | [65] | 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 | [28] | In a cultural sense, BizDevOps is the **shared understanding, responsibility and collaboration** between business, software development and technology operations team members. |
| CU4 | [28] | **humans at the centre** of the approach |
| CU5 | [69] | **Crossing over** to the other side. true teammates |
| CU6 | [16] | 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 | [48] | BizDevOps is Needed to Break **Siloes** Preventing Success |
| CU8 | [32] | breaking down disruptive **silos** within organisations and end-user data can be disseminated **immediately** across the business, optimising performance and improving efficiency |
| CU10 | [79] | **situational awareness** and **shared understanding** and commitment, moving from “**command and control**” and traditional hierarchic management models to new leadership styles and behaviors |
| CU11 | [20] | The most important part, which is having the right people with the right **mindset** of continuous feedback, self-improvement and that create and foster an culture of **openness**, well… that’s just too damn hard work, so go back to your Jenkins and fix that pipeline ASAP, some manager just promised something unrealistic to someone and now it’s up to you to deliver |
| CU15 | [79] | making the shift from “features” to “**outcomes**” and from a flow of “code-to-commit to a flow of “**idea-to-value**.” |
| CU16 | [11] | 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 | [79] | Teams need to be able to respond to **change** |
| CU18 | [68] | **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 |

Table 4 - Barriers

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| BA1 | [22] | Some organisations that are **heavily regulated** (financial, medical), DevOps can be a burden. **ITIL** |
| BA2 | [34] | few business stakeholders **understand** application software code |
| BA3 | [34] | **language** of developers |
| BA4 | [49] | **People don't get the term**. Unless you are a consultant working on Digital Transformation, you won't learn what DevOps actually encompasses, and probably will never work on all aspects of it as an IC |
| BA5 | [63] | Over the last ten plus companies I consulted at, **not one implemented** any of what you call DevOps |
| BA6 | [53] | no matter how nice the ideal of true DevOps philosophy is, in the real world, everyone doesn't have time to **perform all the tasks** that would be required of them and the work keeps getting split up |
| BA7 | [54] | This is the core goal of DevOps: that everyone is both a full fledged developer and knows operations. There is no other way to **reach the DevOps ideal** without this concept. BUT that's simply not possible in today's world. There is too much **overhead** for any single individual to master all those skillsets and be able to maintain them. Some day, cloud and automation may become so refined it's achievable, but right now, it's simply too much work |
| BA8 | [37] | 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 | [5] | 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 |
| BA10 | [22] | a full DevOps practice is impossible, reasons: **ITIL** |
| BA11 | [17] | Organizations with legacy platforms, architectures and systems **carry infrastructure and architecture debts** that prevent them from moving into an Agile-DevOps way of working |

Table 5 - Organizational strategy and governance

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| OS1 | [43] | organizations should be striving to “**Be BizDevOps**” rather than simply “Doing BizDevOps” |
| OS2 | [2] | **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 |
| OS3 | [2] | structured their IT portfolio around products and **value streams**, allowing at portfolio level for minimum governance and maximum synchronization and **autonomy** |
| OS4 | [74] | 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. |
| OS5 | [72] | we need to have a more **holistic view** and co-design of the organization structures and technical architecture |
| OS6 | [35] | 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). |
| OS7 | [23] | 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 |
| OS8 | [2] | 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 |
| OS9 | [45] | service-oriented organizations, where each team develops and manages their **own service** end-to-end from development to production |
| OS10 | [35] | companies that embrace enterprise agility cannot lean too hard on **vendors** and **partners** to provide turnkey IT services |
| OS11 | [35] | An international telecom company internalized hundreds of engineers, mostly by **insourcing** |
| OS12 | [15] | Differentiating engineering capabilities should be **reshored and built in-house**. Having engineers close to the frontlines improves time to value |
| OS13 | [35] | Transform the core IT landscape by distributing IT systems to 'teams of teams and gradually replacing them by **granular services** |
| OS14 | [35] | 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 |
| OS15 | [68] | The **hierarchical structure** of many organizations doesn’t help. It creates a comfort zone that discourages **transparency** and vulnerability |
| OS16 | [35] | creating a diamond-shaped talent composition. increasing the **share** of coders from around 10 percent to 80 percent |
| OS17 | [61] | **Decisions** about new technologies or frameworks are now taken by the team, taking risks and benefits into consideration |
| OS18 | [78] | 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 |
| OS19 | [23] | 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 |
| OS20 | [19] | 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 |
| OS21 | [74] | 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 6 - Teams, collaboration and roles

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| TE1 | [74] | **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 | [68] | So the challenge is to find people from IT and the business who can operate in the **purple shaded area** |
| TE3 | [46] | The **development team** is composed of business analysts and professional developers |
| TE4 | [11] | 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 |
| TE5 | [56] | DevOps team can be a great team to join if you want to get to do a **wide variety of tasks** and keep your options open for lateral movement down the road |
| TE6 | [35] | “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 | [69] | find people from IT and business who can operate in this bridged **middle ground** area |
| TE8 | [15] | 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 | [72] | 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 | [50] | 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 | [4] | customer business experts **work closely** with our IT experts using a Design Thinking approach |
| TE12 | [35] | 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 | [35] | 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 | [35] | 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. |
| TE15 | [21] | The most important part, which is having the right people with the right mindset of continuous feedback, self-improvement and that create and foster an culture of openness, well… that’s **just too damn hard work**, so go back to your Jenkins and fix that pipeline ASAP, some manager just promised something unrealistic to someone and now it’s up to you to deliver |
| TE16 | [54] | This is the core goal of DevOps: that everyone is both a full fledged developer and knows operations. There is **no other way to reach the DevOps ideal** without this concept. BUT that's simply not possible in today's world. There is **too much overhead** for any single individual to master all those skillsets and be able to maintain them. Some day, cloud and automation may become so refined it's achievable, but right now, it's simply too much work |
| TE17 | [74] | team is a **microcompany** |
| TE18 | [74] | 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 | [4] | the **business team** sets requirements and works directly with developers to establish priorities for Agile software development product backlogs |
| TE20 | [17] | **2 pizza team** |
| TE21 | [33] | squads of max. **6 to 9 people** |
| TE22 | [60] | 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 | [69] | 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 | [79] | need to define, agree and commit to “**desired behaviors**” and what is effective **collaboration** |
| TE25 | [25] | 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 | [35] | **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 | [68] | 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 | [68] | 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 | [35] | 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 | [69] | 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 | [4] | 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 | [46] | Integrated requirement management. The business provides their **requirements** and feedback on the live app (minimal viable product) through a user-friendly feedback mechanism |
| TE33 | [43] | **Requirements are a team sport** and management and stakeholders must be committed to building a culture that fosters this behavior |
| TE34 | [43] | 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 | [68] | 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 | [68] | **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 | [68] | **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 |
| TE38 | [56] | why a lot people don’t agree with taking on juniors is because you can **end up doing a wide variety of stuff** across a lot of disciplines requiring both a good **depth and breadth of knowledge** and skills to execute |
| TE39 | [77] | DevOps is about working with a team and optimizing flow of value to the market. Lots of **skills and skillsets** will go into that optimization. For **Jrs** understanding suboptimal delivery flow can also be a challenge, but any **Sr** worth their weight will be happy to tell you all about their delivery challenge horror stories to help give you some things to focus on and unravel |
| TE40 | [9] | 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 | [68] | **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 | [16] | 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 | [46] | **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 | [1] | it feels more natural that someone a bit more **senior** is in a ops/devops/architect position |
| TE45 | [30] | **shared language** of service-level objectives (SLOs) |
| TE46 | [44] | 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 |
| TE47 | [65] | Another part of forming the **common language** among BizDevOps participants revolves around metadata |

Table 7 - Best practices

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| BP1 | [35] | most digital start-ups can **release** at virtually any time as needed—**weekly, daily, or hourly** |
| BP2 | [16] | 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 | [78] | 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. |
| BP4 | [59] | remove **organizational and time barriers** between development teams and other software lifecycle participants so that they can build, test, and release software products **faster** and more reliably |
| BP5 | [74] | 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 | [72] | 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 | [30] | enhance remediation and incident **response** efforts |
| BP8 | [74] | Real-world services can be **up to date**, available, and robust **without interruption** |
| BP9 | [78] | 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 | [14] | They were actually bypassing marketing, sales, communications, risk, finance … they were **talking to customers directly**. They are getting feedback, **instant feedback**. |
| BP11 | [32] | 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 | [46] | **instant feedback** loop between the business analysts and the developers |
| BP13 | [46] | **feedback loop** of less then a month |
| BP14 | [72] | people interpreting these **feedback loops** at different levels |
| BP15 | [25] | inform yourself in **real time** around what is working and what is not |
| BP16 | [29] | **immediate feedback** on all the new applications, features and services |
| BP17 | [19] | DevOps 2.0 is now focused on **extending the benefits of feedback to the entire organization** (marketing, sales, product, etc.) |
| BP18 | [30] | 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 | [29] | 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 | [50] | implement a real-time **dashboard** of business KPIs that provides a clear indication of the business value delivered with every release |
| BP21 | [19] | 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 | [80] | 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 | [50] | 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 |
| BP24 | [26] | Your people are going to build and implement a **well defined process**. Because the process is defined, you can build automation in support of it. The result of the automation is speed and efficiency. |
| BP25 | [47] | Unplanned work is a reality that every team faces–a reality that most often impacts team productivity. Development and operations teams can better manage unplanned work with **established processes** and clear prioritization while focusing on planned tasks |
| BP26 | [70] | DevOps involves a set of **technical processes** such as: Continuous Development, Continuous Integration (CI), Continuous Testing (CI), Continuous Deployment (CD) and Continuous Monitoring |
| BP27 | [11] | 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 | [27] | connecting containerization, and continuous integration platforms to create continuous delivery **pipelines** that give new functionality quicker with better quality and less risk |
| BP29 | [71] | Multi-dimensional **moments-of-truth** for customers |
| BP30 | [12] | **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 | [36] | Adopting notion of **shifting left** helps to recognize issues earlier |
| BP32 | [32] | 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 | [32] | 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 | [69] | stand in the shoes (or sit in the chairs) of **end users**, so everyone can feel what they experience |
| BP35 | [5] | 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 | [43] | 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 | [43] | many application lifecycle management tools do not have the most appropriate features to **manage requirements in a collaborative and holistic way** |
| BP38 | [42] | 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 | [66] | Our whole team explores the business domain, asking questions to define the **requirements** that the desire demands |
| BP40 | [42] | 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 | [42] | 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 | [41] | **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 | [41] | **Robotic Process Automation** (RPA) where digital software robots perform repetitive tasks across applications to improve business processes execution |
| BP44 | [41] | 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 | [37] | 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 | [41] | **automated Performance Testing** and Functional Testing to development teams and **automated Application Monitoring** to operations teams |
| BP47 | [27] | increase **automation**, mainly in **testing** and quality assurance |
| BP48 | [39] | 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 | [18] | business working across the whole life cycle, ability to do course correction and steering during the lifetime of a project |
| BP50 | [35] | achieving missions with as few **handovers** as possible |
| BP51 | [16] | 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 | [61] | 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 | [61] | 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 | [39] | "**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 | [15] | **pull data quickly** from myriad sources and combine them in new and creative ways |
| BP56 | [15] | 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 | [15] | **Data Governance** and Management, which data assets exist today and which critical datasets should be owned or acquired for advantage |
| BP58 | [57] | **Data Management is often missing from the DevOps picture**. safety vs flexibility tradeoff |
| BP59 | [15] | blend new data science techniques with a **deep understanding of business processes and value drivers** |
| BP60 | [15] | 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 | [19] | 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 | [46] | **One-click deployment**. The one-click deployment to any cloud ensures the app can be released in minutes |
| BP63 | [19] | 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 | [19] | making changes to the functionalities of a system that may also impact in the **application performance monitoring** (APM) tools |
| BP65 | [64] | 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 | [19] | 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 | [5] | **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 | [5] | If we **launch a feature** and no one likes it, then we can instantly roll it back |
| BP69 | [41] | **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 | [27] | The need for quick delivery is also expected to **automate** some domains of routine code development |
| BP71 | [73] | increasing adoption of **microservices** architecture and **shift-left** |
| BP72 | [15] | put data into action and to build **strategic assets**, combine the data (i.e., to do something creative with the data) |
| BP73 | [15] | **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 |

Table 8 - Techniques and tools

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| TO1 | [65] | 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 | [27] | **Application Performance Management** tools |
| TO3 | [27] | BizDevOps is only possible because of the technology that gives real-time software **analytics** to enterprises |
| TO4 | [19] | making these new functionalities available to the end user based on **business rules** instead of the operation teams |
| TO5 | [45] | 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 | [45] | BizDevOps **metrics map**, mapping of IT metrics with business requirements |
| TO7 | [71] | Methods such as **Scrum** and **Kanban** provide valuable tools to keep the focus even in a complex environment |
| TO8 | [67] | **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 | [40] | **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 | [17] | 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 | [45] | 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 | [27] | 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 | [17] | using **application portfolio rationalization** (APR) techniques |
| TO14 | [67] | **hypothesis-driven** approach. What is the minimum product that provides the customer with a benefit? |
| TO15 | [67] | Regular **retrospectives** with a focus on action items (measures) for continuous improvement show what has worked in the past – and what has not |

Table 9 - Problems and risks

|  |  |  |
| --- | --- | --- |
| Code | Ref. | Data |
| PR1 | [42] | 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 |
| PR 2 | [65] | **too many tools** which essentially do the same job as each other |
| PR 3 | [72] | **Cognitive load**. the product became too complex to be owned by the team (i.e.: team is reaching its maximal cognitive load) |
| PR 4 | [37] | 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? |
| PR 5 | [10] | The **upfront planning** aspects of the workflow continue to exist in a vacuum, with limited ability to pivot based on ongoing market trends |
| PR 6 | [38] | 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 |
| PR 7 | [72] | 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 |
| PR 9 | [3] | 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 |
| PR 10 | [31] | **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 |

REFERENCED GREY LITERATURE

[1] 8ersgonna8. 2022. Starting career in DevOps with no professional tech exp?. I think it will be d…. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/tah72r/starting\_career\_in\_devops\_with\_no\_professional/i0125u6/

[2] Shaaron Alvares. 2019. DOES 2019: BMW Journey to 100% Agile and BizDevOps Product Portfolio. *InfoQ*. Retrieved January 17, 2023 from https://www.infoq.com/news/2019/11/bmw-devops/

[3] Arc-E-Tect. 2021. BizDevSecOps. *Medium*. Retrieved January 17, 2023 from https://medium.datadriveninvestor.com/bizdevsecops-did-i-miss-somebody-6948433847c3

[4] AZUR Group. 2018. Using BizDevOps and Microsoft Tools to Develop Cloud Solutions More Efficiently. *AZUR Group*. Retrieved January 16, 2023 from https://www.groupeazur.ca/utiliser-les-outils-bizdevops-et-microsoft-pour-developper-des-solutions-cloud-plus-efficacement/

[5] Justin Baker. 2016. DevOps 2.0. *LaunchDarkly*. Retrieved January 17, 2023 from https://launchdarkly.com/blog/devops2/

[6] Pavan Belagatti. 2021. DevOps Trends to Anticipate in 2021. *DEV Community*. Retrieved January 16, 2023 from https://dev.to/pavanbelagatti/devops-trends-to-anticipate-in-2021-4cfd

[7] Sreejani Bhattacharyya. 2022. XOps has emerged as a popular term–what is it? *Analytics India Magazine*. Retrieved January 16, 2023 from https://analyticsindiamag.com/xops-has-emerged-as-a-popular-term-what-is-it/

[8] BizDevOps. 2021. Supercharge Your Business Efforts with Smooth Development and Operations. *BizDevOps*. Retrieved January 16, 2023 from https://bizdevops.com/about/

[9] T. Blogumas. 2020. Differences Between Junior DevOps and Senior DevOps Engineers. *DevOps Dudes*. Retrieved January 17, 2023 from https://medium.com/devops-dudes/differences-between-junior-devops-and-senior-devops-engineers-8d0f28b8b30b

[10] Blueprint. 2017. BizDevOps Enabling Digital Transformation. *Blueprint*. Retrieved January 17, 2023 from https://www.blueprintsys.com/blog/bizdevops-digital-transformation

[11] Brainhub. 2022. BizDevOps (DevOps 2.0) Is the New Iteration of DevOps. *HackerNoon*. Retrieved January 16, 2023 from https://hackernoon.com/bizdevops-devops-20-is-the-new-iteration-of-devops

[12] Brainhub. BizDevOps - a Bridge Between Business and Tech. Retrieved January 16, 2023 from https://brainhub.eu/library/bizdevops-in-nutshell

[13] BuddahsThumb. 2022. DevOps is dead, long live DevOps. *r/devops*. Retrieved January 16, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[14] Lampros Chortarias. 2016. ING’s new way of working through BizDevOps. *Business Architecture & Consultancy BAC*. Retrieved January 17, 2023 from http://www.digitalsocialstrategy.org/bac/2016/12/07/ings-new-way-of-working-through-bizdevops/

[15] Karalee Close, Antoine Gourévitch, Marc Schuuring, Marc Sterman, Lucas Quarta, and Aziz Sawadogo. 2020. Digital Acceleration Is Just a Dream Without a New Approach to Tech. *BCG Global*. Retrieved January 17, 2023 from https://www.bcg.com/publications/2020/how-to-successfully-accelerate-digital-transformation

[16] CloudOps. 2019. Everything You Need to Know About BizDevOps. *CloudOps*. Retrieved January 17, 2023 from https://www.cloudops.com/blog/everything-you-need-to-know-about-bizdevops/

[17] Cognizant. 2019. *17 Must-Do’s to Create a Product-Centric IT Organization*. Cognizant.

[18] Contributor. 2016. Putting the “Biz” in DevOps. *DevOps.com*. Retrieved January 17, 2023 from https://devops.com/putting-biz-devops/

[19] Contributor. 2017. DevOps 2.0 for Digital Transformation. *DevOps.com*. Retrieved January 16, 2023 from https://devops.com/devops-2-0-digital-transformation/

[20] Deleted. 2022. DevOps is dead, long live DevOps. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[21] Deleted. 2022. DevOps is dead, long live DevOps. Honestly this ship has sailed. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/i4aeo5q/

[22] Allan Denot. 2017. When should you not use DevOps? *Quora*. Retrieved January 17, 2023 from https://www.quora.com/What-is-DevOps-2-0

[23] Bill Doerrfeld. 2022. How Does IT Management Govern EverythingOps? *DevOps.com*. Retrieved January 17, 2023 from https://devops.com/how-does-it-management-govern-everythingops/

[24] EFS Consulting. 2023. BizDevOps. *EFS Consulting*. Retrieved January 16, 2023 from https://efs.consulting/en/bizdevops/

[25] Silvia Fregoni. 2020. BizDevOps: Gaining a competitive advantage in an app-centric world. *SiliconANGLE*. Retrieved January 17, 2023 from https://siliconangle.com/2020/01/29/bizdevops-gaining-competitive-advantage-app-centric-world-cleur/

[26] Grant Fritchey. 2021. What makes DevOps so efficient and fast?. DevOps is not magic. *Quora*. Retrieved January 17, 2023 from https://www.quora.com/What-makes-DevOps-so-efficient-and-fast/answer/Grant-Fritchey

[27] Navdeep Gill. 2020. BizDevOps – The Evolution from Agile and DevOps. *XENONSTACK*. Retrieved January 17, 2023 from https://www.xenonstack.com/insights/bizdevops

[28] Ad van der Graaff, Vanessa Sequeira, and Andrea Gatti. 2019. Working together: BizDevOps for competitive advantage. *PwC*. Retrieved January 16, 2023 from https://www.pwc.com.au/digitalpulse/bizdevops-competitive-advantage.html

[29] Andreas Grabner. 2020. Digital Transformation Is More About a BizDevOps Culture than Tech. *DevOps.com*. Retrieved January 17, 2023 from https://devops.com/digital-transformation-is-more-about-a-bizdevops-culture-than-tech/

[30] Andreas Grabner. 2021. Empowering BizDevOps Teams With SLOs. *DevOps.com*. Retrieved January 17, 2023 from https://devops.com/empowering-bizdevops-teams-with-slos/

[31] heavyrain123. 2022. DevOps is dead, long live DevOps. I agree with you. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/i4aucjm/

[32] INDVSTRVS. 2019. BizDevOps Breaks Down Silos, Improves User Experience. *INDVSTRVS*. Retrieved January 16, 2023 from https://indvstrvs.org/bizdevops-breaks-down-silos-improves-user-experience/

[33] ING Belgium. (33) Feature Engineer Payments - BizDevOps Engineer. *LinkedIn*. Retrieved January 17, 2023 from https://www.linkedin.com/jobs/view/feature-engineer-payments-bizdevops-engineer-at-ing-belgium-3206787770/?originalSubdomain=be

[34] Nick Ismail. 2018. BizDevOps for business and IT collaboration. *Information Age*. Retrieved January 16, 2023 from https://www.information-age.com/devops-bizdevops-business-10116/

[35] Quentin Jadoul, Dániel Róna, and Alexander Sukharevsky. 2021. The five core IT shifts of scaled agile organizations. *McKinsey*. Retrieved January 16, 2023 from https://www.mckinsey.com/capabilities/people-and-organizational-performance/our-insights/the-five-core-it-shifts-of-scaled-agile-organizations

[36] jameswanbook. 2018. What Is The Future Of DevOps?. Based on the market. *TechRepublic*. Retrieved January 17, 2023 from https://www.techrepublic.com/forums/discussions/what-is-the-future-of-devops/

[37] Shoeb Javed. 2020. Why DevOps Doesn’t Work for Enterprise Applications. *dzone.com*. Retrieved January 17, 2023 from https://dzone.com/articles/why-devops-doesnt-work-for-enterprise-applications

[38] Shoeb Javed. Introducing BizDevOps- Why DevOps doesn’t work for enterprise applications. Retrieved January 16, 2023 from https://www.worksoft.com/corporate-blog/introducing-bizdevops-why-devops-doesnt-work-for-enterprise-applications

[39] Tim Johnson. 2022. DevOps Has Evolved Beyond Shift Left. *CloudBees*. Retrieved January 17, 2023 from https://www.cloudbees.com/blog/shift-left-done-right

[40] Vernon Keenan. 2021. Devops 3.0 Puts Management into Focus. *SalesforceDevops.net*. Retrieved January 16, 2023 from https://salesforcedevops.net/index.php/2021/09/08/devops-3-0-puts-management-into-focus/

[41] Charlotte Kummer. 2018. Automation for BizDevOps as a Whole. Are you fully ready for the Agile Enterprise? *Robotic automation solutions*. Retrieved January 16, 2023 from https://www.automai.com/automation-for-bizdevops-as-a-whole-are-you-fully-ready-for-the-agile-enterprise-1/

[42] Andrew Kum-Seun. 2020. Bridge the Business-IT Chasm With BizDevOps. *Info-Tech Research Group*. Retrieved January 17, 2023 from https://www.infotech.com/software-reviews/research/bridge-the-business-it-chasm-with-bizdevops

[43] Andrew Kum-Seun. 2020. BizDevOps Starts With Great Requirements. *SoftwareReviews*. Retrieved January 17, 2023 from https://www.softwarereviews.com/categories/application-lifecycle-management/research/bizdevops-starts-with-great-requirements

[44] Gabriel Lowy. 2015. Delivering Value with BizDevOps. *DEVOPSdigest*. Retrieved January 16, 2023 from https://www.devopsdigest.com/delivering-value-with-bizdevops

[45] Max Martynov. 2019. DevOps 2.0 is here, and it’s time to put end-to-end continuous delivery pipelines behind every project. *Grid Dynamics*. Retrieved January 17, 2023 from https://blog.griddynamics.com/the-continuous-delivery-problem-has-been-solved-and-its-time-to-implement/

[46] Mendix. Development Process. *Mendix*. Retrieved January 16, 2023 from https://www.mendix.com/evaluation-guide/dev-process/

[47] John Michael. 2022. What are the DevOps benefits and uses?. DevOps had a positive impact. *Quora*. Retrieved January 17, 2023 from https://www.quora.com/What-are-the-DevOps-benefits-and-uses/answer/John-Michael-1687

[48] Jeanne Morain. 2017. BizDevOps: Solving the Digital Dilemma for Transformational Success. *DEVOPSdigest*. Retrieved January 17, 2023 from https://www.devopsdigest.com/bizdevops-solving-the-digital-dilemma-for-transformational-success

[49] MrScotchyScotch. 2022. DevOps is dead, long live DevOps. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[50] Aliasgar Muchhala. 2020. BizDevOps to enable a product-thinking approach. *Capgemini India*. Retrieved January 17, 2023 from https://www.capgemini.com/in-en/insights/expert-perspectives/bizdevops-to-enable-a-product-thinking-approach/

[51] Siraj Muneera. 2020. What is DevOps 2.0? *Quora*. Retrieved January 16, 2023 from https://www.quora.com/What-is-DevOps-2-0

[52] Ojasvi Nath. 2022. What’s the Next Big Leap in Digital Innovation: Analysts’ Take. *Spiceworks*. Retrieved January 16, 2023 from https://www.spiceworks.com/tech/innovation/articles/next-big-leap-in-digital-innovation/

[53] No-Safety-4715. 2022. DevOps is dead, long live DevOps. The core reality of your complaint. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[54] No-Safety-4715. 2022. DevOps is dead, long live DevOps. This is the core goal of DevOps. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/i4aiy3o/

[55] Obsidian743. 2022. DevOps is dead, long live DevOps : devops. Retrieved January 16, 2023 from https://www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[56] Paddy\_does\_stuff. 2022. Starting career in DevOps with no professional tech exp?. It can depend on. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/tah72r/starting\_career\_in\_devops\_with\_no\_professional/

[57] Project Management Institute. The Workflow of Disciplined DevOps. *PMI*. Retrieved January 17, 2023 from https://www.pmi.org/disciplined-agile/process/disciplined-devops/the-workflow-of-disciplined-devops

[58] Ben Putano. 2020. 6 Software Development Trends for 2020: Developers Needed. *Stackify*. Retrieved January 16, 2023 from https://stackify.com/software-development-trends-2018/

[59] Andy Raindell. 2022. What is DevOps, and why is it necessary today? What are the advantages and disadvantages of DevOps?. DevOps is a set of practices. *Quora*. Retrieved January 17, 2023 from https://www.quora.com/What-is-DevOps-and-why-is-it-necessary-today-What-are-the-advantages-and-disadvantages-of-DevOps/answer/Andy-Raindell

[60] Tim Regulski and Sana Humes. 2017. How CVP Puts the Biz in DevOps. *CVP*. Retrieved January 17, 2023 from https://www.cvpcorp.com/blog/how-cvp-puts-the-biz-in-devops

[61] Ruth Reinicke. 2016. What is Biz, Dev and Ops or BizDevOps? *LeanIX*. Retrieved January 16, 2023 from https://www.leanix.net/en/blog/what-is-biz-dev-and-ops-or-bizdevops

[62] Rjv. 2019. RJV ’s Blog: Automating BDD - CI/CD, BizDevOps. *RJV ’s Blog*. Retrieved January 16, 2023 from https://ravichandranjv.blogspot.com/2019/10/automating-bdd-ccd-bizdevops.html

[63] rproffitt. 2022. Why DevOps Adoption Should Matter to Developers in 2022? *TechRepublic*. Retrieved January 17, 2023 from https://www.techrepublic.com/forums/discussions/why-devops-adoption-should-matter-to-developers-in-2022/

[64] Bernd Rücker. 2018. BizDevOps — the true value proposition of workflow engines. *Medium*. Retrieved January 17, 2023 from https://blog.bernd-ruecker.com/bizdevops-the-true-value-proposition-of-workflow-engines-f342509ba8bb

[65] Antony Savvas. 2022. The most important DevOps tools for tech workers. *Information Age*. Retrieved January 17, 2023 from https://www.information-age.com/the-most-important-devops-tools-for-tech-workers-20123/

[66] Schuberg Philis. BizDevOps. *Schuberg Philis*. Retrieved January 17, 2023 from https://schubergphilis.com/en/how-we-work/bizdevops

[67] Sebastian Schulze and Jacob Tiedemann. 2018. DevOps becomes BizDevOps. *DevOps Conference*. Retrieved January 17, 2023 from https://devopscon.io/blog/devops-becomes-bizdevops/

[68] Henk van der Schuur. 2020. BizDevOps: Bridging that Dominant Divide between Business and IT. *Medium*. Retrieved January 17, 2023 from https://stories.schubergphilis.com/bizdevops-bridging-that-dominant-divide-between-business-and-it-c1194297a706

[69] Henk van der Schuur. 2021. BizDevOps: The way to bring technology to the heart of business (and…. *Schuberg Philis*. Retrieved January 16, 2023 from https://schubergphilis.com/en/stories/bizdevops-the-way-to-bring-technology-to-the-heart-of-business-and-build-bridges-along-the-way

[70] Dev Shah. 2020. Is DevOps agile?. DevOps vs Agile. *Quora*. Retrieved January 17, 2023 from https://www.quora.com/Is-DevOps-agile-1/answer/Dev-Shah-234

[71] Manas Shome and Raghubir Bose. The next disruption in BFSI: Embracing Industry 4.0 with BizdevOps. Retrieved January 17, 2023 from https://www.tcs.com/what-we-do/industries/banking/white-paper/adopting-bizdevops-financial-services

[72] Eduardo Silva and Manuel Pais. 2021. DevOps is Not Enough for Scaling and Evolving Tech-Driven Organizations: a Q&A with Eduardo da Silva. *InfoQ*. Retrieved January 16, 2023 from https://www.infoq.com/articles/devops-not-enough-scaling-tech-driven-organizations/

[73] sophieclark. 2019. What Is The Future Of DevOps?. Increasing adoption of microservices architecture. *TechRepublic*. Retrieved January 17, 2023 from https://www.techrepublic.com/forums/discussions/what-is-the-future-of-devops/

[74] Dominik Strube. 2018. BizDevOps for digital services. *Bosch ConnectedWorld Blog*. Retrieved January 16, 2023 from https://blog.bosch-si.com/business-models/bizdevops-for-digital-services/

[75] TechTarget. 2016. Why -- and how -- BizDevOps is going to change everything. *TechTarget*. Retrieved January 16, 2023 from https://www.techtarget.com/searchsoftwarequality/essentialguide/Why-and-how-BizDevOps-is-going-to-change-everything

[76] t\_go\_rust\_flutter. 2022. DevOps is dead, long live DevOps : devops. Retrieved January 16, 2023 from https://www.reddit.com/r/devops/comments/u0yo1h/devops\_is\_dead\_long\_live\_devops/

[77] wageof. 2022. Starting career in DevOps with no professional tech exp?. DevOps is a skill and a craft. *r/devops*. Retrieved January 17, 2023 from www.reddit.com/r/devops/comments/tah72r/starting\_career\_in\_devops\_with\_no\_professional/

[78] Phil Wainewright. 2016. How IT leads disruptive innovation at banking giant ING. *diginomica*. Retrieved January 17, 2023 from https://diginomica.com/how-it-leads-disruptive-innovation-at-banking-giant-ing

[79] Paul Wilkinson. 2019. Is Your Organization ‘Fit for the Future’? *DevOps.com*. Retrieved January 17, 2023 from https://devops.com/is-your-organization-fit-for-the-future/

[80] Myrvin Yap. 2020. User experience done right with BizDevOps. *Frontier Enterprise*. Retrieved January 17, 2023 from https://www.frontier-enterprise.com/user-experience-done-right-with-bizdevops/