DevOps Capabilities and Metrics

Ricardo Manuel Duarte Amaro
Instituto Superior Técnico
Universidade de Lisboa
Lisbon, Portugal
ricardo.d.amaro@tecnico.ulisboa.pt

***Abstract*—Nowadays, IT organizations face ever-changing consumer demands, competitiveness, regulatory environments, and sophisticated external threats. As a result, they seek a competitive advantage by using DevOps and its capabilities, such as improving user experience, increasing productivity, and team collaboration.**

However, DevOps adoption remains inconsistent, emphasizing the need to provide relevant data and insights to management in their decision-making process to enhance efficiency while applying DevOps capabilities. Unfortunately, there is a lack of systematization between the effective use of these capabilities and the ideal metrics for each one.

Therefore, Design Science Research is done with two Multivocal Literature Reviews to elicit the main DevOps metrics and capabilities and semi-structured interviews to build an outcome-based capability evaluation matrix, focusing on metrics (KPIs) for promoting DevOps adoption. A definition of DevOps capability and another for metrics is identified, along with 37 DevOps capabilities and 24 main metrics are defined and categorized.

It is concluded that cultural capabilities have the highest overall impact. Empowering teams to make decisions and organizational culture are top categories. Capabilities are, dynamic and have been growing and changing over the years, being defined by the ability of an organization to perform practices. The five top metrics that an organization should start by measuring are, in order, M03-Deployment Frequency (DF), M01-Mean Time To Recover/Restore (MTTR), M42-Team happiness, M02-Mean Lead-time for Changes (MLT) and M04-Change Failure Rate (CFR).

***Keywords:* DevOps; Capabilities; Competencies; Metrics; Performance; Adoption.**

I. INTRODUCTION

In an organization where software development is part of the core business, having a competitive advantage [1] by doing things better than competitors is a requirement for any IT organization [2]. However, the need for frequent software delivery, without sustained builds, proper testing and release automation, generates burnout and pain in the engineers doing operations [3], decaying software delivery performance [4] and leading to poor reliability.

In response to this broken process, we see the emergence of Developer(Dev) and Operations(Ops) (DevOps), an organizational approach that emphasizes empathy and encourages greater collaboration between engineering teams involved in software delivery [5], in order to reduce development time, improve deployment rates, increase stability, optimize Mean Time to Recover, and lower deployment and implementation costs [6]. Management should have a clear vision of the steps to take ahead based on information and metrics in order to increase efficiency [1], thus the success of applying DevOps capabilities should be measured with existing DevOps metrics. An effective way to control and assess these levels of maturity and adoption would be to use very specific metrics that assess existing capabilities, informing, for example, whether a given software delivery process within the pipeline is performing optimally or could be improved.

To the best of the author's knowledge, no study has been conducted that compares various DevOps capabilities or practices with existing DevOps metrics [7], so the problem of this thesis is identified as follows:

Problem: The assessment of the desired success of each DevOps capability or practice is irregular due to a lack of systematization between the successful application of these capabilities or practices and the ideal metrics for each one.

While DevOps adoption success is irregular, as a few impediments exist, shown by Smeds et al. [8] and measuring its maturity can be hard, relating different DevOps capabilities with existing DevOps metrics will support management decision process towards increasing performance in the software development life cycle within the organization.

However, since there exists a lack of systematization of the different DevOps capabilities or practices in existing DevOps metrics, the purpose of this research is to explore a relationship between the metrics and the capabilities/practices, to elicit the main DevOps metrics for each DevOps capability/practice, align the relations impacting positively each found metric and create an evaluation matrix of the DevOps capabilities/practices. For that end, the following research questions are used towards the mentioned goal:

RQ1. What are the main DevOps capabilities or practices?
RQ2. Where are capabilities and practices mentioned?
RQ3. How authors distinguish capabilities from practices?
RQ4. What are the main DevOps metrics?
RQ5. What is the purpose of each metric?
RQ6. Why is each metric important?
RQ7. How are DevOps capabilities categorized?
RQ8. How are the main metrics categorized?
RQ9. What DevOps capabilities have a positive impact in which main metrics?

II. RESEARCH METHODOLOGY

In an initial report, a Literature Review has been conducted to identify the problem of this research. Here it is used the activities of Design Science Research (DSR) proposed by Hevner et al. [9] as the main research methodology. For the build phase [10] two Multivocal Literature Review (MLR) [11] are conducted as mentioned in Section II-B to help identify the artifact. After having the initial artifact, it is conducted the evaluation phase [12] using Semi-structured Interviews mentioned in Section II-C, in order to support the findings of the initial research and improve the artifact.

A. Design Science Research

This DSR is divided in two processes [9, 10, 13], build (process of construction of an artifact) supported by two MLR in Section III, plus Section IV, resulting in a proposal in Section V and evaluate (to determine how well the artifact behaves) in Section VI.

Baskerville et al. [14] underlines the dual mandate of the DSR, which is to use acquired knowledge to solve problems, create changes, or improve existing solutions; and at the same time to generate knowledge, perceptions and theoretical explanations. Hevner [13] calls the attention to the pragmatism of DSR in cycles of relevance and rigor, in the creation of artifacts. The DSR methodology process in the field of Information Systems, shown in Figure 1, is based on Develop/build and justify/evaluate [9].

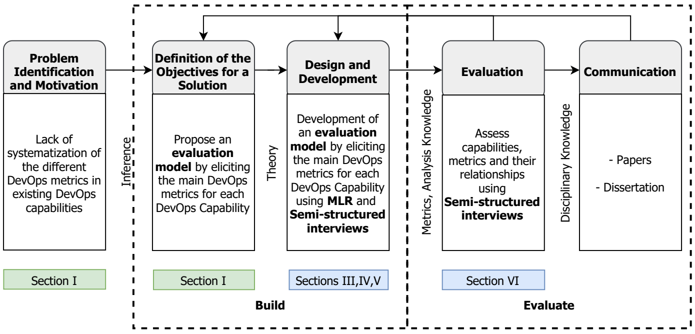

Figure 1. Adapted phases of the DSR process model [9]

B. Multivocal Literature Review

A MLR is a type of Systematic Literature Review (SLR), which aims to incorporate gray literature like blogs, videos, web-pages and white papers, which are constantly produced by Software Engineering (SE) practitioners outside academic forums, notwithstanding the published (peer-reviewed) writing like journal articles and conference papers. Therefore, MLR is important for the expansion of the research by including literature that normally wouldn't be taken due to its "gray" nature [11, 15], as show in the Figure 2 on page 2.

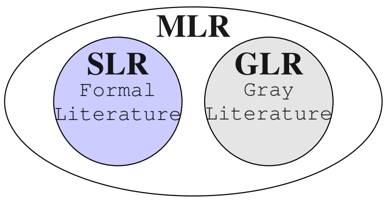

Figure 2. The relationship of SLR, GLR and MLR Studies [11]

Some examples of successful DevOps research, in the same area, using MLR already exist [16] [17] [18], thus corroborating the practical usefulness of this method for the proposed research, expanding the diversity of sources that are available in a variety of forms, reflecting different purposes and perspectives [19].

C. Semi-structured Interviews

During these semi-structured interviews, an interview protocol is used and participants are informed about the research goals, while asking for their consent to be interviewed. Interviews with a semi-structured format are a common approach in development research. It is common for people to give vital information that the researcher had not before [20].

These kinds of interviews are a common method for qualitative research [21]. The researchers and participants engage in a formal interview, using a developed "interview guide" with a list of questions and topics. However, the interviewer can also follow other topics in the conversation that can guide when he or she feels this is appropriate [22, 23]. The DevOps capabilities confirmed in the literature, together with the related metrics to be found, are used to form an interview protocol in which the targeted evaluation model acts as a guide to elicit metrics from capabilities.

III. DevOps Capabilities Multivocal Literature Review

In the build phase of DSRs, the process of construction of the artifact is undertaken by using MLR.

A. Planning, Conducting and Reporting the MLR

This section corresponds to the process summary of the first MLR. The full process is shown in 3, which exposes the planning, conducting and reporting as proposed by Garousi et al. [11].

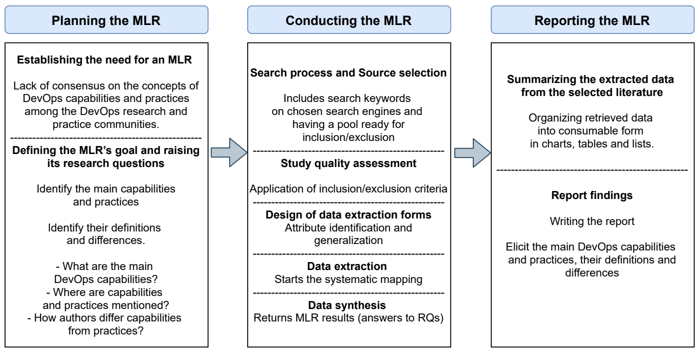

Figure 3. DevOps capabilities Multivocal Literature Review (MLR) Steps [11]

Based on the main purpose of this research, an investigation for scientific and 'gray' related work was done that addresses or explores the area of capabilities and practices, which can be translated into the previously defined RQ1, RQ2, RQ3. The search string used to perform the search to retrieve the maximum number of studies and the chosen datasets are listed here:

- **Search String**: `(devops AND (practices OR capabilities))`.
- **Datasets**: The search engines used were, Google search, Scopus, Web of Science, IEEE, ACM and EBSCO.

Using a review protocol, the first set of papers is obtained. After the search is complete and snowballing is done, inclusion and exclusion criteria is applied for refining the search results. Written in English, identified author, published in and after 2013, with publication date, full-text accessible, no advertisement or job post and Mentions DevOps capabilities or practices. The results arise from the evaluation of the full text of the 93 publications eligible for extraction of any relevant information for this research.

B. RQ1 - What are the main DevOps capabilities or practices?

In this section, the MLR provides a list of 37 capabilities based on the literature review of the 93 publications.

C01. *Cross team collaboration and communication*, mentioned in 81 publications. In order to enable cross-functional collaboration between application teams, operations and security teams [24–35] the organization has to identify the stakeholders, including customers [36], of every project so that they join, have insights about various project phases and processes, and start making valuable contributions.

C02. *Continuous integration (CI)*, mentioned in 80 publications. Continuous integration takes tasks like testing and building, and automates them [37, 38], driving teams to produce high quality software, to reduce the cost of ongoing software development and maintenance [25, 39–45], and to increase the productivity of the teams. [28, 40]. The CI process creates canonical builds and packages that are ultimately deployed and released [46].

C03. *Continuous delivery and deployment automation*, mentioned in 78 publications. While continuous delivery automates the entire software release process with a manual step, continuous deployment automates that step, deploying smaller changes [47] to production as soon as they are released from acceptance testing [48, 49], without manual intervention [46], releasing faster and more frequent, reducing the risk of production deployments and providing faster feedback to the teams [30, 50].

C04. *Proactive monitoring, observability and autoscaling*, mentioned in 74 publications. It is critical to monitor the infrastructure [47, 48, 51–53], whether it is in the cloud or in a local data center. Combining proactive monitoring with autoscaling can automatically solve capacity issues [32, 54–56] and reduces the need to scale the system manually.

C05. *Test automation and environments*, mentioned in 62 publications. Getting quick feedback on the impact of changes across the software delivery life cycle is fundamental to integrate quality into software [52, 57]. It is important to have automated and correctly provisioned test environments along the pipeline [28, 55, 58, 59], reducing long lead times [38, 49, 60–62].

C06. *Continuous improvement of processes and workflows*, mentioned in 46 publications. Continuous improvement is enabled through a combination of continuous integration, deployment, testing, workflows and monitoring [31], like, implementing branch-naming consistency, where all work originates from the same source while developing on a branch referencing a ticket [63], or applying consistent patterns across multiple applications [62].

C07. *Version control system*, mentioned in 45 publications. Version control and automation are tightly intertwined [37, 39, 64, 65] enabling efficiency and productivity [57, 66, 67]. Version control extends versioning to all production artifacts [52], such as application code, configurations, system settings, and scripts for automating build and environment setup [46, 61].

C08. *Support learning culture and experimentation*, mentioned in 44 publications. Organizations that develop a learning culture [68–70] and comprehend its impact on organizational performance encourage engineers to have the ability to work alone and experiment [39, 58] to test business concepts and new ideas, to write and update requirements during development [60, 62].

C09. *Trust/empower teams to make decisions and changes*, mentioned in 42 publications. Trust is essential in every relationship, but it is especially critical for DevOps [63] to improve software delivery performance and job satisfaction empowering them with the ability to make educated decisions about the tools and technologies they employ [34, 44, 51, 54, 60, 68]. This helps to create greater outcomes [28, 55, 71].

C10. *Focus on people, process and technology*, mentioned in 32 publications. People, process and technology are the three pillars of a software development project. There must be a feeling of community, sharing a common goal, and contributing to the common cause [44]. Improving the culture is an ongoing journey [72, 73]. DevOps unifies people, processes, and technology: when all three are aligned toward the same business goals, innovation can be implemented more quickly [62, 74–76].

C11. *Configuration management*, mentioned in 30 publications. Is practiced in one form or another as part of any software engineering project. A Software Configuration Management (SCM) is a system for managing the evolution of software products [77, 78], automating the configuration, monitoring, managing, and maintenance of all entities of infrastructure and systems like servers, applications, storage, networks, and all managed services [42].

C12. *Cloud infrastructure and cloud native*, mentioned in 30 publications. The US National Institute of Standards and Technologies (NIST) defines five essential characteristics of cloud computing [79]: On demand self-service, broad network access, resource pooling, rapid elasticity and measured service [55, 80]. Each service may be deployed individually [81], with flexibility, tool sets, and scalability for applications. Serverless architectures on clouds can dramatically reduce DevOps effort [64]. In a pipeline—for example for worker nodes, deploying artifacts to test or even production environments [82].

C13. *Artifacts versioning and registry*, mentioned in 28 publications. Enable organizations to centrally store artifacts and build dependencies as part of the software delivery process [34, 37, 83]. It is important to version these artifacts in a repository manager [84–86], either they are promoted containers along the pipeline, bundles, charts, packages or any other kind to make the changes visible, reliable and repeatable [5, 86] for all production artifacts [6, 33, 70].

C14. *Loosely coupled architecture/ microservices*, mentioned in 27 publications. Microservices are an architecture design for building a single application with smaller services that run independently, usually communicating via API calls [87]. Improves agility and helps organizations to easily grow their product at a cheaper cost and in a shorter time [26]. Each service may be deployed separately and decentralized. Produced and delivered using automated tools and automated procedures [30].

C15. *Database change management/ release alignment*, mentioned in 25 publications. Database change management [37, 88] and release alignment change management processes [43, 89, 90], when well implemented, help developers and IT professionals to easily manage database updates, system configurations, deploy new code quickly and fix incidents faster.

C16. *Infrastructure as code*, mentioned in 25 publications. With infrastructure as code [47, 71, 91–93], it is possible to express procedures in code [94] rather than setup infrastructure or software manually. Using technologies like Chef, Puppet, Ansible, or Salt [95, 96]. That way it is possible to use version control to keep track of all infrastructure modifications in a repeatable and more efficient manner [97].

C17. *Emergency response/ proactive failure notification*, mentioned in 24 publications. Proactive failure notification [37] focus on actionable notifications based on the values being monitored and that have known failure thresholds, instead of a reactive system to alert when it has already failed [27, 28], improving emergency response efficiency [98] and reducing risk of customer impact [51].

C18. *Containerization*, mentioned in 24 publications. Containers are efficient for app development and hosting [64]. They allow DevOps, developers, and system administrators to swiftly, securely, and effectively test, build, deploy, and manage applications [48]. This capability has become a new standard in DevOps pipelines, clusters, and applications [66, 87]. C19. *Open* *source software adoption*, mentioned in 22 publications. Open source adoption correlates with DevOps success [38, 55, 99, 100], and the knowledge of open source solutions for testing and deployment is a must for a DevOps engineer [64]. This model is well represented in the DevOps tool set [67, 101] with impact in early DevOps emergence [68]. Organizations assemble and contribute open source parts, which has become a reliant software supply chain [52, 60].

C20. *Shift left on security*, mentioned in 20 publications. Integrating security into the design and testing phases of the software development process is key to driving IT performance. Including security reviews of applications, including the infosec team [39, 46, 58]. Shift left on security is related to DevSecOps [34] concept and emphasizes automating as much as possible security policies in order to accelerate processes, decrease human error and aiding in quality improvement [102] and audits [62, 103].

C21. *Transformational leadership*, mentioned in 20 publications. Is a style in which leaders inspire and encourage teams to attain better levels of performance [30, 33]. Transformational leaders focus on the growth and performance of their followers and organization [52, 61]. Effective leaders [62] impact software delivery performance by pushing the use of technical and product management capabilities [104].

C22. *Trunk based development*, mentioned in 19 publications. In trunk based development, each developer works in small batches, merges that work into trunk at least once (and potentially several times) a day [5, 90], consistent with commonly accepted continuous integration practices [33, 105, 106].

C23. *Monitor systems to inform business decisions*, mentioned in 19 publications. Inform business decisions using visual dashboards [28, 53, 62, 65, 107] allows organization to track configuration changes made to servers along with databases and deployments [74] that have taken place, along with various metrics, logs, and graphs [108, 109] to give a holistic view of changes happening in the system.

C24. *Performance/Westrum organizational culture*, mentioned in 19 publications. Ron Westrum developed a typology of organizational cultures that includes three types of organizations [110]. *Pathological organizations* are characterized by low cooperation across groups and a culture of blame. *Bureaucratic cultures* are preoccupied with rules and positions, and responsibilities are compartmentalized by department. *Generative organizations* are performance oriented [58], with good information flow, high cooperation and trust, bridging between teams [52, 111].

C25. *Working in small batches*, mentioned in 16 publications. Working in small batches with a lightweight approval process helps ensure work can get through the system quickly [53] with shorter lead times [37]. It enables fast flow through the development pipeline, fixing errors as they are discovered vs. at the end [112] also allowing to deliver of MVPs, features, and bug fixes sooner, which also helps enable the customer feedback loop above [34].

C26. *Centralized log management*, mentioned in 15 publications. Centralized logs for applications with multiple servers facilitate debugging [55, 113] when sent to a common service that enables easy centralization, rotation, and deletion [35, 52, 114].

C27. *Lightweight/streamlining change approval*, mentioned in 15 publications. Replace heavyweight change-approval systems with peer review [104]. Lead times and release frequency improve considerably [33] with negligible impact on system stability [54, 104].

C28. *Visibility of work in the value stream*, mentioned in 14 publications. Understand and visualize the flow of work [33, 38] from idea to customer outcome in order to drive higher performance. Make the value flow visible for everyone to understand where their piece fits into the whole flow [53, 104] from the business all the way through to customers [39, 111].

C29. *Work in progress limits* or Work in process limits, mentioned in 14 publications. Prioritize work, limit the number of things that people are working on [61, 115], and focus on getting a few high-priority tasks done [104]. Work in Progress (WIP) limits [53] are identified and enforced. Flow is defined [34]. Overload is limited [30].

C30. *Customer/user feedback*, mentioned in 13 publications. Drive better organizational outcomes by gathering customer feedback [111, 116] and incorporating it into product and feature design [53].

C31. *Blameless postmortems/reduced fear of failure*, mentioned in 12 publications. By removing blame, fear is reduced, and by reducing fear, teams are empowered to surface and solve problems more efficiently. Mistakes occur. Holding blameless postmortems [6, 52, 54, 55] is an effective technique of learning from mistakes [104, 117].

C32. *Data-driven approach for improvements*, mentioned in 12 publications. Analyzing factual data [93] can help an organization achieve performance. Sharing application graphs [113, 118], usage patterns with team members to get everyone aligned. Include scalability, testing, and deployment to simplify the entire process [35].

C33. *Job satisfaction*, mentioned in 11 publications. Job satisfaction is the top predictor of organizational performance [24, 54]. DevOps adoption also help avert burnout [61], a common reason why technical people leave jobs [119].

C34. *Test data management*, mentioned in 10 publications. Managing test data can be challenging [53]. Define the right strategies for managing test data effectively along with approaches to provide fast, secure data access for testing [102] like adequate data to run a test suite, acquiring data on demand, conditioning and limiting the amount of test data needed in the pipeline [46].

C35. *Chaos engineering*, mentioned in 9 publications. Contemporary and distributed software programs must be capable of dealing with unexpectedly tumultuous environments [26, 52, 53, 76, 82]. As a result, such systems must be built from the start to withstand unanticipated problems and shortcomings in production contexts [120].

C36. *Code maintainability*, mentioned in 9 publications. Code maintainability [37, 60] is essential for making it simple for developers to identify, reuse, and alter code, as well as keep dependencies up to date [121].

C37. *Visual management capabilities*, mentioned in 9 publications. Improves a company's capacity to assess progress toward goals, manage change and improvements [122]. Visual management of work and pulling it through the system [123] is a critical part of the First Way of DevOps [52].

C. RQ2 - Where are capabilities and practices mentioned?

Capabilities have been mentioned interchangeably as practices in 66 publications Table I and eight publications even distinguish practices from capabilities [27–29, 43, 61, 73, 74, 124]. The two terms are described across several types of white and gray literature as seen in Figure 4.

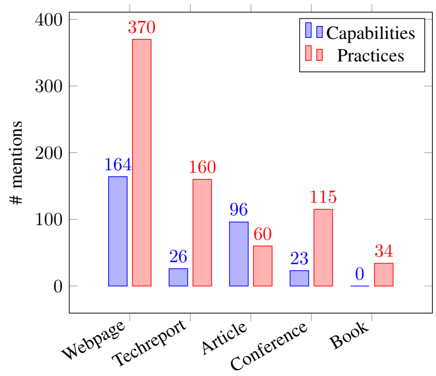

Figure 4. Number of publications mentioning capabilities and practices among sources.

It can be observed, that the webpages, overwhelmingly created by practitioners, have the most mentions as practices, but also include a substantial number of mentions as capabilities. The same happens in Techreport, Conference and Book, which are closer to the gray literature.

Research has shown that DevOps practitioners are more focused on DevOps as a concept, while the scientific community tries to organize capabilities in a way that abstracts more generic concepts applicable to building skills and enablers. Nevertheless, the concepts are the same, only at different stages of the process. In Table I it is mentioned that one publication indicates a DevOps practice definition [74] to be a subset implementation of a capability.

Table I
SIX PUBLICATION PROPERTIES IDENTIFIED FROM THE MLR.

Property	Publications	Total
Interchangeably mentions capabilities and practices	[6, 24–26, 30, 31, 33, 35, 36, 38, 42, 44, 45, 48–52, 54, 55, 57, 59, 62–70, 72, 75, 76, 80, 82, 94, 100, 107, 111, 117, 125–149]	66
Mentions capabilities directly	[27–29, 34, 38, 39, 43, 45, 47, 50, 58, 60, 61, 73, 74, 88, 99, 124, 150]	19
Presents different or reorganized capabilities compared to Senapathi et al. [28]	[27, 34, 39, 43, 45, 47, 50, 58, 60, 73, 74, 88, 99, 124]	14
Distinguishes practices from capabilities	[27–29, 43, 61, 73, 74, 124]	8
Indicates a definition for capability	[27, 28, 33, 47, 61, 74]	6
Indicates a definition for practice	[74]	1

Six publications indicate various capabilities definitions like a higher level categorization of practices [47], are important to enhance software company's profitability, productivity and market share [61], capabilities are the core DevOps aspect which comprises capabilities such as "continuous planning, collaborative and continuous deployment, continuous integration and testing, continuous release and deployment, continuous infrastructure monitoring and optimization, continuous user behavior monitoring and feedback and service failure recovery without delays" [27]. The previously mentioned research done by Senapathi et al. [28] and 2017 State of DevOps report [33] defines them as a combination set that can change over time. These set of capabilities have been changing over the years.

D. RQ3 - How authors distinguish capabilities from practices?

The differentiation of why some authors mention capabilities and other practices is discussed here and defined in Section III-E. It is largely observed that the word "capability" is used when the observation is external or at a high-level overview. It is then a matter of perspective. When talking about a capability, we see a third-party assessment of something that is being looked at from the outside, while observing a group to see what they are capable of doing.

Whereas, a practice is seen from the standpoint of the internal team or group, realizing "I am doing these things". That ability converted to an action is then mentioned with the term "practice". Therefore, authors will speak about capabilities from an evaluation standpoint, and practices from a hands-on approach perspective. The capability definition points to an organization's "ability" to perform or achieve a certain process, whereas a practice is referred to more at the level of DevOps practitioners and thus more observed in the gray literature publications.

Clear examples of this more formal research concept were presented earlier by Smeds et al. [8], Senapathi et al. [28] and more recently in the book Accelerate [46], in DORA [37, 50, 60] and in several journal articles or proceedings [27, 27, 29, 30, 47, 51, 61, 74]. A capability is also mentioned as a "construct" [46, 99] and that there are "capabilities we are building" [52] in order to enable the organization for a certain practice. Organizations should target developing capabilities and habits in their people [151] as an enabler for continuous improvement and functional skills.

The usage of capabilities or practices is not consensual, and because this study is not about achieving that consensus, it is chosen to use the term **capabilities**; nevertheless, others might also use practices. A consensus between the two should be further researched in the future.

E. DevOps Capabilities Synthesis

The list of the most studied and approached capabilities was collected highlighting *cross team collaboration and communication*. In this collaboration, developers have access to a self-service platform that provides a foundation for automation, standardization, and team autonomy to enable the other three most mentioned capabilities: *continuous integration*; *continuous delivery and deployment automation*; *proactive monitoring, observability and autoscaling*.

Table II
DEFINITION OF DEVOPS CAPABILITY.

A **DevOps capability** is here defined as the ability to do something [152] or by the quality, or state of being capable [153]. On the other hand, technological capabilities are defined as the information and skills - technical, managerial and institutional - that enable productive enterprises [30, 104, 151] to utilize equipment and technology efficiently [154]. It consists of the combined skills accumulated and developed by its members over time.

A **DevOps practice**, differently from the definition of capability given in Table II, is defined as the use of the mentioned capability by an individual or a group such as the engineering team. As a result, authors will discuss capabilities from the perspective of evaluation, but refer to practices from the perspective of a hands-on approach. Therefore, this research will use the term "capability" from now on, since that is applicable to evaluating DevOps adoption.

IV. DevOps Metrics Multivocal Literature Review

The process continues to the second MLR that gathers the DevOps metrics needed for the research proposal.

A. Planning, Conducting and Reporting the MLR

A summary of the second MLR is given in 5 with planning, conducting and reporting as proposed by Garousi et al. [11].

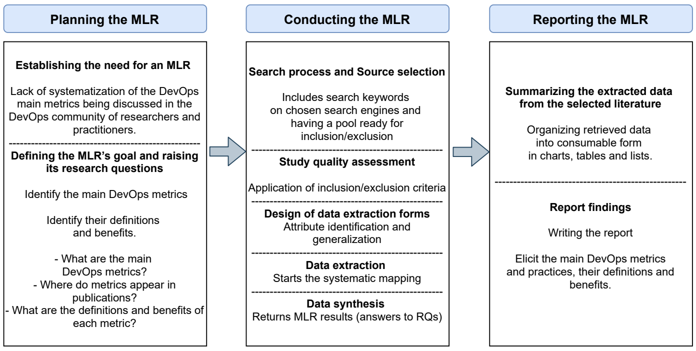

Figure 5. DevOps metrics Multivocal Literature Review (MLR) Steps [11]

A MLR is used to search for scientific and 'gray' literature that discusses or investigates DevOps metrics, which can then be captured into RQ4, RQ5, and RQ6 research questions. A search using various keywords is conducted to identify studies that may provide answers to the specified research topics.

- **Search String**: `(devops AND (metrics OR measures OR kpi OR indicator))`.
- **Datasets**: The search engines used were, Google search, Scopus, Web of Science, IEEE, ACM and EBSCO.

The criteria for inclusion and exclusion for this MLR is literature written in English, with identified author, published in and after 2010, with publication date, full-text accessible, no advertisement or job post and mentions DevOps metrics. As a result, 114 articles are left for full-text document review.

B. RQ4 - What are the main DevOps metrics?

From all the publications, selected in the Multivocal Literature Review, there were 22 main DevOps metrics found:

M01. Mean Time To Recover/Restore (MTTR)
M02. Mean Lead-time for Changes (MLT)
M03. Deployment Frequency (DF)
M04. Change Failure Rate (CFR)
M05. Service Availability and Uptime
M06. Deployment duration time
M07. Mean Time To Detection (MTTD)
M08. Application response time
M09. Defect escape rate
M10. Cycle Time Value (CTV)
M11. Service Level Agreements (SLAs) and Service level Objectives (SLOs)
M12. Deployment size
M13. Production Error and Incident rate
M14. Customer tickets Volume and Feedback
M15. Mean time to failure (MTTF)
M16. Customer Usage and traffic
M17. Pipeline automated tests success/fail rate
M18. Westrum organizational culture measures
M19. Automated Test Code Coverage
M20. Work in Progress (WIP) /Load
M21. Unplanned Work Rate (UWR)
M22. Wait Time

C. RQ5 - What is the purpose of each metric?

The list of the main DevOps metrics including their purpose description, the references that mention each specific metric and the total of references discovered is shown in Table III.

Table III. Purpose and references for each main DevOps metric.

ID	Metric	Purpose	References	Total
M01	Mean Time To Recover/Restore (MTTR)	Measures the mean of the time required to recover or restore service from a failure in production.	[5–7, 33, 38, 46, 51, 52, 54, 56, 57, 60, 62, 91–93, 95, 96, 99, 102, 104, 109, 111, 115, 116, 119, 120, 155–222]	96
M02	Mean Lead-time for Changes (MLT)	Indicates how long it takes for a change to go from code committed to code successfully running in production.	[5, 6, 33, 38, 46, 52, 54, 56, 57, 60, 62, 91–93, 95, 96, 99, 104, 108, 111, 115, 116, 119, 120, 155–162, 165–168, 170–174, 176–178, 180–191, 193, 194, 196–202, 205, 206, 208–218, 220, 223–233]	91
M03	Deployment Frequency (DF)	Checks how often changes are deployed to production.	[5, 6, 33, 38, 46, 51, 52, 54, 56, 57, 60, 62, 91–93, 95, 99, 102, 104, 108, 109, 111, 115, 116, 119, 120, 155–162, 164–174, 176–194, 198, 200, 202, 205–216, 218, 219, 222, 225, 227, 230, 232, 234]	88
M04	Change Failure Rate (CFR)	Informs how often a change in production fails and must be immediately remedied.	[6, 33, 38, 46, 52, 54, 57, 60, 62, 91, 93, 95, 99, 102, 111, 115, 116, 120, 155–159, 162, 164, 166–168, 170–174, 176–178, 180–185, 187–192, 194, 198, 200, 202, 206–219, 222, 223, 227, 230, 233]	72
M05	Service Availability and Uptime	Shows the percentage a service is available during a period of time.	[5, 35, 46, 51, 52, 60, 92, 95, 96, 99, 104, 116, 157, 160, 167, 170, 179, 183, 189, 191–193, 197, 200, 201, 205, 212, 220, 227, 234, 235]	31
M06	Deployment duration time	Informs on how long it takes to deploy a set of changes.	[91, 92, 96, 102, 109, 116, 157, 158, 160, 167, 170, 177, 181, 183, 191–193, 198, 199, 205, 207, 209, 215, 219, 220, 222, 225, 230, 232, 236]	30
M07	Mean Time To Detection (MTTD)	Measures the mean of the time required to detect a failure in production.	[52, 56, 92, 95, 116, 157, 158, 160, 162, 164, 167, 169, 170, 179, 183, 186, 189, 193, 198, 200, 201, 203, 205, 206, 220, 233]	26
M08	Application response time	How an application responds to increases or decreases in user traffic and activity.	[38, 46, 52, 92, 108, 109, 119, 157, 158, 160, 167, 179, 189, 193, 196, 198, 201, 205, 211, 212, 216, 221, 227]	23
M09	Defect escape rate	Indicates the number of defects discovered in production versus the number of defects found during development.	[56, 92, 102, 108, 109, 157, 158, 160, 162, 167, 168, 170, 177, 183, 192, 193, 195, 205, 219, 228]	21
M10	Cycle Time Value (CTV)	Provides information on the full Cycle Time Value, beginning with the idea and finishing with user feedback. .	[5, 46, 51, 52, 57, 102, 104, 116, 118, 119, 163, 164, 183, 192, 195, 199, 204, 210, 214, 224, 226]	21
M11	Service Level Agreements (SLAs) and Objectives (SLOs)	Sets customer expectations for service availability with SLA and internal teams with SLO.	[5, 35, 46, 52, 60, 92, 104, 108, 116, 118, 160, 170, 183, 189, 193, 200, 201, 205, 220, 225, 230]	21

Continued on next page

Table III – *Continued from previous page*

ID	Metric	Purpose	References	Total
M12	Deployment size	Shows the number of changes incorporated in each production release.	[92, 96, 116, 119, 157, 158, 160, 167, 168, 177, 183, 192, 193, 198, 205, 212, 216, 227–229]	20
M13	Production Error and Incident rate	Measures the frequency of faults and incidents in production following a deployment.	[92, 96, 109, 160, 167, 169, 171, 177, 186, 192, 193, 195–197, 200, 201, 205, 225, 233]	19
M14	Customer tickets Volume and Feedback	Indicates the level of satisfaction of customers using their feedback.	[92, 96, 116, 158, 160, 162, 167, 168, 170, 183, 193, 205, 206, 212, 218, 225, 228, 230]	19
M15	Mean time to failure (MTTF)	Exposes the average time a flawed deployment into a system will manage to run until it fails.	[92, 95, 116, 158, 160, 165, 169, 170, 183, 186, 193, 198, 200, 203, 205, 218, 234]	17
M16	Customer Usage and traffic	Measures usage and traffic of customer-facing applications when there are defined business goals to increase.	[35, 52, 56, 92, 102, 160, 170, 193, 196, 199, 201, 205, 211, 212, 227, 237]	16
M17	Pipeline automated tests success/fail rate	Shows the rate of success/failure of Pipeline automated test jobs.	[46, 52, 104, 158, 160, 167, 175, 177, 192, 193, 199, 205, 227, 229]	14
M18	Westrum organizational culture measures	Result of the Westrum cultural assessment [110]	[7, 27, 35, 46, 52, 60, 96, 99, 104, 111, 207, 209, 210, 220]	14
M19	Automated Test Code Coverage	Measures how many lines, statements, or blocks of code are tested using the suite of automated tests.	[5, 52, 92, 102, 170, 192, 199, 204, 219, 220, 229, 234, 235]	13
M20	Work in Progress (WIP) /Load	Presents the number of open issues of each type (story, defect, task).	[46, 52, 104, 199, 201, 220, 223, 224, 226, 232, 234, 237]	12
M21	Unplanned Work Rate (UWR)	Indicates the amount of time spent on tasks that weren't in the initial plan. Shouldn't be over 25%.	[6, 33, 35, 46, 104, 158, 171, 183, 200, 218, 233]	11
M22	Wait Time	Measures the amount of time spent waiting for the next step to add value.	[52, 104, 198, 204, 218, 220, 223, 224, 226, 237]	10

Concluded

The 22 key metrics presented here are among the initial 58 identified in the MLR. They were referenced ten times or more and are relevant in organizational and mission goals like operational efficiency, customer happiness, profitability, productivity, product or service quality [174].

D. RQ6 - Why is each metric important?

Here the MLR goes over *why* these DevOps metrics are important.

M01. *Mean Time To Recover/Restore (MTTR)* handling unexpected outages should be as quick as possible and a focus for DevOps KPI monitoring, as it contributes to greater customer satisfaction, faster application delivery, and better cost control [51, 155, 160, 161].

M02. *Mean Lead-time for Changes (MLT)* is the hardest to measure of the top four key DevOps metrics. It is important to know how long it takes to deploy a change, and understand delaying problems like technical debt [38, 57, 174, 214].

M03. *Deployment Frequency (DF)* happens many times per day for elite industry performers [60, 93], where there is continuous development, testing, and integration of small changes continuously improving applications. Important to respond quickly to business requests for new features and to critical issues [205, 215].

M04. *Change Failure Rate (CFR)* should be as low as possible in DevOps. It is a critical metric that businesses must monitor, since unsuccessful deployments result in revenue losses and dissatisfied consumers [183, 189].

M05. *Service Availability and Uptime* should be more than 99.999% of the time for users of the system, with a ratio based on availability, reliability and uptime. Achieving 100% availability is unrealistic, once planned downtime for maintenance is accounted for. Therefore, it is important to track and distinguish from unplanned downtime [160, 167].

M06. *Deployment duration time* allows to track the progress of the deployment. It can help identify potential problems and allow a dramatic increase in revenue by using that extra time to develop more value-added services [198, 205, 207] .

M07. *Mean Time To Detection (MTTD)* demonstrates the effectiveness of monitoring technologies and intelligent alerting techniques, assessing whether current response efforts are appropriate. High detection times may result in bottlenecks [92, 183].

M08. *Application response time* that are long may indicate bottlenecks that require attention, since it degrades user experience and satisfaction. The cause might be code, data access, protocol problems, or a variety of other factors [109, 160].

M09. *Defect escape rate* is an important DevOps metric to track how often defects make it to production. Abnormally high defect rates could be the first sign of problems in testing, qualification or in team performance. [160, 170, 193].

M10. *Cycle Time Value (CTV)* is the time it takes to go from idea to production, spanning all the steps of build, test, stage, and push to production. Slowing down the cycle with manual testing or assessments creates friction for developers and the business [183, 195].

M11. *Service Level Agreements (SLAs) and Objectives (SLOs)* serve to define external and internal availability goals as commitments between providers, clients and internal teams, defining how fast releasing is possible, while measuring that performance with Service level Indicators (SLIs) [60, 116, 118, 183, 238, 239].

M12. *Deployment size* is important to keep an eye on the deployment artifacts that are shipped to production with each release and track the amount of the bug fixes and feature requests delivered [92, 183, 193].

M13. *Production Error and Incident rate* tells the DevOps team how often new bugs appear in running applications. It is important to capture spikes in the error rate because these can indicate that something is not right. Not all errors are equally impactful for customer's trust [160, 169, 196].

M14. *Customer tickets Volume and Feedback* is a good assessment of a successful DevOps adoption. Reduce customer tickets by preventing bugs from reaching production, and repair them as fast as possible if they do, improving quality. Customer satisfaction leads to a competitive advantage [28, 92, 160, 162].

M15. *Mean time to failure (MTTF)* is an indication of how long on average the system or a component can run before failing after deployment. It can indicate problems with the deployment or quality of the software. For example, maybe there are not enough tests covering different scenarios that might contain bugs [95, 169, 186].

M16. *Customer Usage and traffic* metrics allow tracking user engagement with application features. Increased engagement after

Table IV
DEFINITION OF DEVOPS METRICS.

A **DevOps metric** is defined as a quantifiable, business-relevant, trustworthy, actionable and traceable [51, 92, 171, 183, 204] indicator that aids organizations in making data-driven decisions to continuously improve their software delivery process [7, 93, 158–160, 162, 184, 188, 196, 206, 212, 213, 235].

an update may indicate users are pleased with the updates. If traffic reports indicate too much or no activity, there might be an issue, suggesting a malfunctioning component is generating the anomaly [92, 193, 205].

M17. *Pipeline automated tests success/fail rate* is another contributor to the speed of the DevOps process. Automated tests are faster than humans, but it should deliver the right results. To increase velocity, the team needs to make extensive usage of unit and functional testing. It is important to know how often changes are causing tests to break [160, 193, 205].

M18. *Westrum organizational culture measures* results are key to fostering a performance-oriented organizational environment stating Westrum et al. 2004 [110]. DevOps teams must be supported by transformational leadership [34, 240] to enable this culture, thus empowering strategic alignment and reducing conflict [99, 111, 207, 241].

M19. *Automated Test Code Coverage* is a DevOps best practice for measuring the percentage of unit or integrity tests coverage [92, 192].

M20. *Work in Progress (WIP) /Load* is a lean manufacturing principle shown in the Toyota Production System [151] that enhances teams overall throughput by limiting work in progress (partially completed work). This increases total velocity [223, 234, 237].

M21. *Unplanned Work Rate (UWR)* tracks the amount of time spent on unplanned work. A high UWR may indicate that efforts wasted on unexpected errors that were not identified early in the workflow. The difference between acting on warning signs or having an unexpected outage [242] is defined as unplanned work [6, 183].

M22. *Wait Time* (queuing or waste) is an estimate of the time that the work item spends idle in a non-productive state during its processing by the value stream. Wait time is in opposition to touch time when value is created [52, 198, 223].

E. DevOps Metrics Synthesis

From the extensive MLR done, it is understood that DevOps metrics should aim to quantify the right elements in order to understand if DevOps is working [119, 175, 188]. There are 69 publications that try to organize and explains each stated metric and 49 publications that also try to define what are DevOps metrics. Following those dispersed definitions this MLR is now able to propose a unified definition of DevOps metrics in Table IV.

The DevOps metrics typically measure either throughput, stability or quality [171], while quantifying a faster cadence (efficiency) and value addition (effectiveness) [224].

V. RESEARCH PROPOSAL

This section focuses on the development of a proposed assessment model by eliciting the main DevOps metrics for each DevOps capability using the two MLR done and a set of semi-structured interviews [20] as mentioned in Section II.

A. Interviews with Practitioners

For the construction of the proposed artifact, this research used the results from the two MLR done in Section III and in Section IV and a total of 21 interviews were conducted in this section to build the research proposal. The practitioners interviewed were: three individual contributors, six team leads and two principal engineers. The companies they work for are very diversified: Google, Noesis, Newdecision, Siemens, Azores Governement, Inuits.eu, IBM, amazee.io, Acquia, Drupal Association, Manifold, Bloomidea, Deeper Insights, Facebook, Dropsolid and some freelancers.

The majority of 7 practitioners is between 35-44 years of age, six are between 25-34 years, seven are between 45-54 years and one has more than 55 years. Their locations are Belgium, India, Portugal, Spain, Switzerland, USA, and United Kingdom.

The resulting answers to research questions are here summarized.

B. RQ7 - How are DevOps capabilities categorized?

During the interviews the alignment of capabilities into categories was discussed, as well as their relevance and categorization leading to a more refined list as seen in Table V.

With this resulting table, a matrix for DevOps capabilities and metrics was obtained that includes Cultural, Measurement, Process, and Technical categories to be used in Section V-E.

C. RQ8 - How are the main metrics categorized?

The categorization of the main metrics is shown in Table VI, with Business, Change, Cultural and Operating types of Key Performance Indicators (KPIs).

D. RQ9 - What DevOps capabilities have a positive impact in which main metrics?

The metrics that are impacted positively by capabilities are represented in Table VII.

E. Proposed Capability Evaluation Matrix

The proposed DevOps capability evaluation matrix in Table VII is an artifact that aims to be used for future strategic planning.

VI. EVALUATION

The evaluation of the artifact follows the process in the design cycle mentioned in Section II-A, based on the framework for evaluation in design science (FEDS) [243] driven by the work of Pries-Heje, Baaskerville and Venable [244–246].

A. Semi-structured Interviews Iterations

Ten semi-structured interviews were done with practitioners with various skills, to perform several assessment iterations, following an outline, with the objective of refining and validating the artifact proposed in Section V-E.

A total of 10 interviews were conducted in this section to evaluate the research proposal. The practitioners interviewed were: Seven individual contributors, one architect, one VP and one product owner. The companies they work for are: Acquia, Boxboat and Liip. Their locations are India, Romania, Switzerland and USA.

Table V
CATEGORIZATION OF DEVOPS CAPABILITIES.

Category	ID	DevOps Capability
Cultural	C01	Cross team collaboration and communication
Technical	C02	Continuous Integration
Technical	C03	Continuous Delivery and Deployment automation
Measurement	C04	Proactive Monitoring, Observability and autoscaling
Technical	C05	Test Automation and environments (Continuous testing)
Process	C06	Continuous Improvement of processes and workflows
Technical	C07	Version Control System
Cultural	C08	Support learning culture and experimentation
Technical	C09	Trust/empower teams to make decisions and changes
Process	C10	Focus on people, process and technology
Technical	C11	Configuration Management
Technical	C12	Cloud infrastructure and cloud native
Technical	C13	Artifacts versioning and registry
Technical	C14	Loosely coupled architecture/ microservices
Technical	C15	Database change management/ release alignment
Technical	C16	Infrastructure as Code
Measurement	C17	Emergency response/ proactive failure notification
Technical	C18	Containerization
Cultural	C19	Open source software adoption
Technical	C20	Shift left on security
Cultural	C21	Transformational leadership
Technical	C22	Trunk based development
Measurement	C23	Monitor systems to inform business decisions
Cultural	C24	Performance/Westrum organizational culture
Process	C25	Working in small batches
Technical	C26	Centralized log management
Process	C27	Lightweight/streamlining change approval
Process	C28	Visibility of work in the value stream
Measurement	C29	Working in progress limits
Process	C30	Customer/user feedback
Cultural	C31	Blameless Postmortems/reduced fear of failure
Process	C32	Data-driven approach for improvements
Cultural	C33	Job satisfaction
Technical	C34	Test data management
Technical	C35	Chaos Engineering
Technical	C36	Code maintainability
Measurement	C37	Visual management capabilities

Table VI
CATEGORIZATION OF MAIN DEVOPS METRICS.

Proposed Category	ID	Main Metric
Operating KPI	M01	Mean Time To Recover/Restore (MTTR)
Change KPI	M02	Mean Lead-time for Changes (MLT)
Change KPI	M03	Deployment Frequency (DF)
Change KPI	M04	Change Failure Rate (CFR)
Operating KPI	M05	Service Availability and Uptime
Change KPI	M06	Deployment duration time
Operating KPI	M07	Mean Time To Detection (MTTD)
Operating KPI	M08	Application response time
Change KPI	M09	Defect escape rate
Change KPI	M10	Cycle Time Value (CTV)
Operating KPI	M11	SLAs and SLOs
Change KPI	M12	Deployment size
Operating KPI	M13	Production Error and Incident rate
Business KPI	M14	Customer tickets Volume and Feedback
Change KPI	M15	Mean time to failure (MTTF)
Business KPI	M16	Customer Usage and traffic
Change KPI	M17	Pipeline automated tests success/fail rate
Cultural KPI	M18	Westrum organizational culture measures
Change KPI	M19	Automated Test Code Coverage
Operating KPI	M20	Work in Progress (WIP) /Load
Operating KPI	M21	Unplanned Work Rate (UWR)
Operating KPI	M22	Wait Time
Cultural KPI	M42	Team Happiness
Cultural KPI	M59	Talent retention

B. Evaluation Results

Primarily the additions or updates to the matrix have become residual, which confirms the data saturation already observed in Section V. A summary of all updated relations can be found in Table VIII. There were residual relations suggested between capabilities and metrics, but there were no suggestions for changing capabilities, metrics or categories, therefore not expressed in the table. This indicates that the evaluation was successful and supporting the proposal's design [247].

C. Validated Artifact

Using the validated artifact here presented, business executives and organization leaders will be able to evaluate the various DevOps capabilities in relation to the outcomes, identifying the capabilities that require the most significant long-term enhancements and highlighting the capabilities that should become the primary focus of future IT investments as seen in Table VIII.

As a few participants mentioned, this will also allow the organization to stimulate discussions about how each capability impacts DevOps adoption, allowing them to figure out where they can get the most value from. Like what capabilities, if included from the start, would have the most influence on a variety of KPIs.

This capability matrix is outcome-based, focusing on important outcomes (KPIs) and how capabilities promote change in those outcomes. This gives clear guidance and strategy on high-level goals (with an emphasis on capabilities to enhance key outcomes) to technical leadership. It also allows team leaders and individual contributors to define progress targets for the current time period based on the capabilities their team is working on.

VII. CONCLUSION

In this chapter, we conclude the research done with communication, general conclusions, limitations and future work.

A. Research Conclusions

This study has brought important contributions to both academia and industry on the DevOps topic. In summary, a Design Science research was done that includes two MLRs on DevOps capabilities, in DevOps metrics and 21 semi-structured interviews in the build phase. To find literature, Google search, Scopus, Web of Science, IEEE, ACM, and EBSCO were utilized, and after applying the inclusion and exclusion criteria and snowballing, 207 papers were identified as relevant to these study topics. In order to evaluate the research proposal, 10 semi-structured interviews were done, resulting in a validated Capability Evaluation Matrix.

Table VII
PROPOSED ARTIFACT SHOWING CATEGORIZED DEVOPS CAPABILITIES INFLUENCING MAIN METRICS.

		Change KPI										Operating KPI									Cultural KPI			Business KPI	
		M02	M03	M04	M06	M09	M10	M12	M15	M17	M19	M01	M05	M07	M08	M11	M13	M20	M21	M22	M18	M42	M59	M14	M16
Cultural	C01	✓	✓	✓	✓	✓	✓					✓	✓	✓		✓					✓	✓	✓	✓	
	C08		✓				✓			✓					✓		✓				✓	✓	✓		
	C19	✓	✓				✓				✓									✓		✓	✓		
	C21	✓	✓	✓			✓					✓	✓								✓	✓	✓	✓	
	C24	✓	✓	✓					✓			✓		✓						✓	✓	✓	✓		
	C31		✓	✓	✓					✓			✓			✓	✓	✓	✓	✓		✓	✓	✓	✓
	C33		✓	✓								✓									✓	✓	✓		
Technical	C02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓													
	C03	✓	✓	✓	✓		✓		✓			✓								✓	✓				
	C05	✓	✓	✓			✓			✓	✓						✓					✓		✓	✓
	C07	✓	✓								✓	✓					✓			✓		✓	✓		
	C09	✓	✓	✓	✓		✓		✓	✓		✓	✓				✓	✓		✓	✓	✓	✓	✓	
	C11	✓		✓							✓											✓	✓		
	C12	✓	✓		✓		✓					✓	✓		✓	✓			✓	✓		✓			
	C13		✓					✓				✓		✓					✓	✓					
	C14	✓	✓	✓				✓	✓			✓			✓	✓		✓		✓				✓	
	C15		✓	✓		✓			✓			✓	✓				✓		✓			✓		✓	
	C16	✓			✓		✓	✓				✓	✓	✓				✓	✓	✓		✓			
	C18		✓	✓	✓			✓		✓	✓	✓	✓		✓	✓			✓			✓			
	C20	✓	✓	✓		✓						✓	✓	✓	✓			✓	✓			✓	✓	✓	
	C22	✓	✓	✓		✓	✓	✓				✓					✓		✓		✓	✓			
	C26			✓								✓	✓	✓			✓		✓	✓		✓		✓	
	C34		✓	✓						✓										✓					
	C35			✓		✓			✓	✓		✓	✓	✓		✓	✓		✓		✓		✓	✓	
	C36	✓																	✓	✓		✓			
Measurement	C04											✓	✓	✓	✓	✓	✓		✓						✓
	C17											✓	✓	✓	✓	✓		✓	✓	✓				✓	
	C23			✓						✓		✓	✓		✓		✓								
	C29	✓	✓									✓						✓			✓	✓	✓		
	C37	✓	✓	✓	✓		✓					✓				✓		✓	✓			✓			
Process	C06	✓	✓				✓					✓						✓	✓	✓	✓				
	C10	✓	✓																		✓	✓	✓	✓	
	C25	✓	✓		✓			✓								✓	✓	✓	✓	✓	✓			✓	
	C27	✓	✓				✓					✓									✓	✓			
	C28		✓	✓								✓					✓		✓			✓		✓	
	C30			✓		✓						✓					✓							✓	✓
	C32	✓	✓	✓								✓				✓						✓			

Legend:

M01 - Mean Time To Recover; M02 - Mean Lead-time for Changes; M03 - Deployment Frequency; M04 - Change Failure Rate; M05 - Service Availability and Uptime; M06 - Deployment duration time; M07 - Mean Time To Detection; M08 - Application response time; M09 - Defect escape rate; M10 - Cycle Time Value; M11 - SLAs and SLOs; M12 - Deployment size; M13 - Production Error and Incident rate; M14 - Customer tickets Volume and Feedback; M15 - Mean time to failure; M16 - Customer Usage and traffic; M17 - Pipeline automated tests success/fail rate; M18 - Westrum organizational culture measures; M19 - Automated Test Code Coverage; M20 - Work in Progress/Load; M21 - Unplanned Work Rate; M22 - Wait Time; M42 - Team Happiness; M59 - Talent retention.

C01 - Cross team collaboration and communication; C02 - Continuous Integration; C03 - Continuous Delivery and Deployment automation; C04 - Proactive Monitoring; Observability and autoscaling; C05 - Test Automation and environments; C06 - Continuous Improvement of processes and workflows; C07 - Version Control System; C08 - Support learning culture and experimentation; C09 - Empower teams to make decisions and changes; C10 - Focus on people; process and technology; C11 - Configuration Management; C12 - Cloud infrastructure and cloud native; C13 - Artifacts versioning and registry; C14 - Loosely coupled architecture; C15 - Database change management; C16 - Infrastructure as Code; C17 - Emergency response; C18 - Containerization; C19 - Open source software adoption; C20 - Shift left on security; C21 - Transformational leadership; C22 - Trunk based development; C23 - Monitor systems to inform business decisions; C24 - Westrum organizational culture; C25 - Working in small batches; C26 - Centralized log management; C27 - Lightweight change approval; C28 - Visibility of work in the value stream; C29 - Working in progress limits; C30 - Customer feedback; C31 - Blameless Postmortems; C32 - Data-driven approach for improvements; C33 - Job satisfaction; C34 - Test data management; C35 - Chaos Engineering; C36 - Code maintainability; C37 - Visual management capabilities.

Table VIII
VALIDATED ARTIFACT WITH CATEGORIZED DEVOPS CAPABILITIES INFLUENCING MAIN METRICS.

		Change KPI										Operating KPI									Cultural KPI			Business KPI	
		M02	M03	M04	M06	M09	M10	M12	M15	M17	M19	M01	M05	M07	M08	M11	M13	M20	M21	M22	M18	M42	M59	M14	M16
Cultural	C01	✓	✓	✓	✓	✓	✓					✓	✓	✓		✓					✓	✓	✓	✓	
	C08	✓	✓		✓		✓			✓					✓		✓	✓			✓	✓	✓		
	C19	✓	✓				✓				✓									✓		✓	✓		
	C21	✓	✓	✓			✓					✓	✓								✓	✓	✓	✓	✓
	C24	✓	✓	✓	✓			✓	✓	✓		✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	
	C31		✓	✓	✓					✓			✓			✓	✓	✓	✓	✓		✓	✓	✓	✓
	C33		✓	✓								✓									✓	✓	✓		
Technical	C02	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓												✓	
	C03	✓	✓	✓	✓		✓		✓			✓								✓	✓			✓	
	C07	✓	✓								✓	✓					✓			✓		✓	✓		
	C09	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓	✓			✓	✓	✓	✓	✓	✓	✓	✓	✓	
	C11	✓		✓							✓											✓	✓		
	C12	✓	✓		✓		✓		✓			✓	✓		✓	✓	✓	✓	✓	✓		✓			✓
	C13		✓					✓				✓		✓					✓	✓					
	C14	✓	✓	✓	✓			✓	✓			✓	✓	✓	✓	✓		✓		✓		✓	✓	✓	✓
	C15		✓	✓		✓			✓			✓	✓	✓		✓	✓		✓	✓		✓	✓	✓	
	C16	✓			✓		✓	✓				✓	✓	✓				✓	✓	✓		✓			
	C18		✓	✓	✓			✓	✓	✓	✓	✓	✓		✓	✓			✓			✓			
	C20	✓	✓	✓		✓	✓	✓		✓		✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	
	C22	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓					✓		✓	✓	✓	✓			
	C26			✓								✓	✓	✓			✓		✓	✓		✓		✓	
	C34		✓	✓						✓										✓					
	C35			✓		✓			✓	✓		✓	✓	✓		✓	✓		✓		✓		✓	✓	
	C36	✓																	✓	✓		✓			
Measurement	C04											✓	✓	✓	✓	✓	✓		✓						✓
	C17											✓	✓	✓	✓	✓		✓	✓	✓				✓	
	C23			✓					✓	✓		✓	✓	✓	✓	✓	✓			✓				✓	
	C29	✓	✓									✓						✓			✓	✓	✓		
	C37	✓	✓	✓	✓		✓					✓	✓			✓		✓	✓			✓			
Process	C06	✓	✓		✓		✓					✓	✓			✓		✓	✓	✓	✓	✓	✓	✓	
	C10	✓	✓																		✓	✓	✓	✓	
	C25	✓	✓		✓			✓								✓	✓	✓	✓	✓	✓			✓	
	C27	✓	✓				✓					✓								✓	✓	✓			
	C28		✓	✓			✓					✓					✓		✓	✓		✓		✓	
	C30			✓		✓						✓					✓							✓	✓
	C32	✓	✓	✓								✓				✓						✓			

Legend:

✓ : Updated relation.

M01-M59, C01-C37: Same Capabilities and Metrics already listed in Table VII.

- This study proposes a consensus definition distinguishing capabilities from practices, based on academic and industry literature review.
- A thorough investigation was conducted in order to find a consensus definition of DevOps metrics across academics and practitioners.
- It was investigated and exposed where the capabilities and practices are mentioned in literature (RQ2) and what are their differences seen (RQ3).
- The purpose of each metric is identified in detail (RQ5) and the reason why each metric is important is analyzed (RQ6).
- From all the literature review done in this research, 37 DevOps capabilities (RQ1) and 24 validated DevOps metrics (RQ4) were identified.
- DevOps capabilities have been researched, explained and after a careful evaluation categorized (RQ7). The same rigorous work was conducted for DevOps metrics (RQ8).
- In order to develop a core study proposal, this investigation drafted the DevOps capabilities that have a beneficial influence in each of the key metrics (RQ9).
- The major outcome of this research is a Capability Evaluation Matrix presented in Table VIII, which has been proposed, debated, and validated by DevOps practitioners.

A set of interesting conclusions arise when identifying the analysis vector that connects the capability categories to the KPI categories based on the validated artifact in Table VIII. In Table IX the most evident conclusion is that cultural capabilities have a strong impact for improving the cultural KPIs and the highest overall impact.

Table IX
CAPABILITY CATEGORIES WEIGHT ON KPI CATEGORIES.

	Change KPIs	Operating KPIs	Cultural KPIs	Business KPIs	Total AVG
Cultural capabilities	45.71%	39.68%	90.48%	42.86%	54.68%
Technical capabilities	49.44%	61.90%	46.30%	33.33%	47.74%
Measurement capabilities	20.00%	64.44%	26.67%	30.00%	35.28%
Process capabilities	30.00%	33.33%	52.38%	42.86%	39.64%

Nevertheless, operating KPIs are mostly impacted by technical and measurement capabilities. On the other hand, process capabilities are the second most impactful on cultural KPIs and when looking at what are the main drivers of change those are technical, cultural and process capabilities. As a result, organizations should prioritize not only technology, but also cultural and process improvements. Capabilities are, dynamic and have been growing and changing over the years. The most relations to metrics are seen in C09-Empower teams to make decisions and changes and C24-Performance organizational culture.

The five top metrics with most of the relations in the model, that an organization should start by measuring are, in order, M03-DF, M01-MTTR, M42-Team happiness, M02-MLT and M04-CFR. Metrics have been extended to 24, divided into four KPI categories: change, operating, cultural and business.

References

[1] J. P. L. . K. C. Laudon, *Management Information Systems: Managing the Digital Firm, Global Edition*, 15th ed. Pearson Education, 2017.

[2] A. Dyck, R. Penners, and H. Lichter, "Towards definitions for release engineering and DevOps," in *Proceedings - 3rd International Workshop on Release Engineering, RELENG 2015*. IEEE Press, may 2015, p. 3.

[3] J. Wettinger, U. Breitenbücher, and F. Leymann, "DevOpSlang – Bridging the Gap between Development and Operations," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. IFIP, 2014, vol. 8745 LNCS, pp. 108–122. [Online]. Available: http://link.springer.com/10.1007/978-3-662-44879-3_8

[4] N. Forsgren and J. Humble, "The Role of Continuous Delivery in it and Organizational Performance," *SSRN Electronic Journal*, pp. 1–15, 2015. [Online]. Available: http://www.ssrn.com/abstract=2681909

[5] J. Humble and D. Farley, *Continuous Delivery: Reliable Software Releases Through Build, Test, and Deployment Automation*. Addison-Wesley Professional, dec 2010.

[6] Puppet Labs, "2015 State of DevOps Report," Puppet Labs, Tech. Rep. 877, 2015. [Online]. Available: http://puppetlabs.com/2015-devops-report

[7] N. Forsgren and M. Kersten, "DevOps Metrics: Your biggest mistake might be collecting the wrong data." *Queue*, vol. 15, no. 6, pp. 19–34, dec 2017.

[8] J. Smeds, K. Nybom, and I. Porres, "DevOps: A Definition and Perceived Adoption Impediments," in *Lecture Notes in Business Information Processing*. Springer, 2015, vol. 212, pp. 166–177.

[9] A. R. Hevner, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *MIS Quarterly: Management Information Systems*, vol. 28, no. 1, pp. 75–105, 2004.

[10] S. T. March and G. F. Smith, "Design and natural science research on information technology," *Decision Support Systems*, vol. 15, no. 4, pp. 251–266, dec 1995.

[11] V. Garousi, M. Felderer, and M. V. Mäntylä, "Guidelines for including grey literature and conducting multivocal literature reviews in software engineering," *Information and Software Technology*, vol. 106, no. September 2018, pp. 101–121, feb 2019.

[12] C. Sonnenberg and J. vom Brocke, "Evaluation Patterns for Design Science Research Artefacts," in *Practical Aspects of Design Science*, 2012, vol. 286, no. Chapter 7, pp. 71–83.

[13] A. R. Hevner, "A Three Cycle View of Design Science Research," *Scandinavian Journal of Information Systems*, vol. 19, no. 2, pp. 87–92, 2007.

[14] R. L. Baskerville, M. Kaul, and V. C. Storey, "Genres of Inquiry in Design-Science Research: Justification and Evaluation of Knowledge Production," *MIS Quarterly*, vol. 39, no. 3, pp. 541–564, mar 2015.

[15] V. Garousi, M. Felderer, and M. V. Mäntylä, "The need for multivocal literature reviews in software engineering," in *Proceedings of the 20th International Conference on Evaluation and Assessment in Software Engineering - EASE '16*. New York, New York, USA: ACM Press, 2016, pp. 1–6.

[16] M. Sánchez-Gordón and R. Colomo-Palacios, "A Multivocal Literature Review on the use of DevOps for e-learning systems," *ACM International Conference Proceeding Series*, pp. 883–888, 2018.

[17] B. B. Nicolau de França, H. Jeronimo, and G. H. Travassos, "Characterizing DevOps by hearing multiple voices," in *ACM International Conference Proceeding Series*, ser. SBES '16, E. DeAlmeida, Ed., Unicesumar; Colivre; Espweb; Tasa Eventos. New York, USA: Assoc Computing Machinery, 2016, Proceedings Paper, pp. 53–62.

[18] L. Prates, J. Faustino, M. Silva, and R. Pereira, "DevSecOps Metrics," in *Lecture Notes in Business Information Processing*, M. J. Wrycza S., Ed. Instituto Universitário de Lisboa (ISCTE-IUL), Lisbon, Portugal: Springer, 2019, vol. 359, pp. 77–90.

[19] R. T. Ogawa and B. Malen, "Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method," *Review of Educational Research*, vol. 61, no. 3, pp. 265–286, sep 1991.

[20] K. Raworth, C. Sweetman, S. Narayan, J. Rowlands, and A. Hopkins, *Conducting semi-structured Interviews*. Oxfam, 2012.

[21] W. C. Adams, "Conducting Semi-Structured Interviews," in *Handbook of Practical Program Evaluation: Fourth Edition*. Hoboken, NJ, USA: John Wiley and Sons, Inc., oct 2015, pp. 492–505.

[22] D. Cohen and B. Crabtree, "Semi-structured Interviews," 2006. [Online]. Available: http://www.qualres.org/HomeSemi-3629.html

[23] C. M. Conway and K. Roulston, "Conducting and Analyzing Individual Interviews," *The Oxford Handbook of Qualitative Research in American Music Education*, no. June, pp. 1–24, 2014.

[24] PuppetGuideCIOs, "The 5 Stages of DevOps Evolution: A Guide for CIOs," 2018. [Online]. Available: https://www.thinkahead.com/wp-content/uploads/2018/12/puppet-5-stages-devops-evolution-cio-guide.pdf

[25] O. Mikhalchuk, "5 core DevOps principles for a painless culture shift | Forte Group," 2020. [Online]. Available: https://fortegrp.com/what-are-the-core-devops-principles/

[26] Infopulse, "6 DevOps Best Practices to Launch Enterprise-Wide Transformations | Continuous Delivery, Microservices Architecture, Collaboration | Infopulse," 2018. [Online]. Available: https://www.infopulse.com/blog/6-devops-best-practices-to-launch-enterprise-wide-transformations/

[27] S. Badshah, A. A. Khan, and B. Khan, "Towards Process Improvement in DevOps: A Systematic Literature Review," in *ACM International Conference Proceeding Series*, ser. EASE '20. New York, NY, USA: Association for Computing Machinery, apr 2020, pp. 427–433.

[28] M. Senapathi, J. Buchan, and H. Osman, "DevOps Capabilities, Practices, and Challenges," in *Proceedings of the 22nd International Conference on Evaluation and Assessment in Software Engineering 2018 - EASE'18*, ser. EASE'18, no. June. New York, USA: ACM Press, jun 2018, pp. 57–67.

[29] R. Mao, H. Zhang, Q. Dai, H. Huang, G. Rong, H. Shen, L. Chen, and K. Lu, "Preliminary Findings about DevSecOps from Grey Literature," in *2020 IEEE 20th International Conference on Software Quality, Reliability and Security (QRS)*. IEEE, dec 2020, pp. 450–457.

[30] P. Rodríguez, M. Mäntylä, M. Oivo, L. E. Lwakatare, P. Seppänen, and P. Kuvaja, "Advances in Using Agile and Lean Processes for Software Development," in *Advances in Computers*, ser. Advances in Computers, A. Memon, Ed. Faculty of Information Technology and Electrical Engineering, University of Oulu, Finland: Academic Press Inc., 2019, vol. 113, pp. 135–224.

[31] M. M. A. Ibrahim, S. M. Syed-Mohamad, and M. H. Husin, "Managing Quality Assurance Challenges of DevOps through Analytics," in *Proceedings of the 2019 8th International Conference on Software and Computer Applications*, ser. ICSCA '19, vol. Part F1479. New York, NY, USA: ACM, feb 2019, pp. 194–198.

[32] M. D. Z. Imam, "What are the Best Practices For Successful Implementation Of DevOps?" 2018. [Online]. Available: https://www.knowledgehut.com/blog/devops/best-practices-for-successful-implementation-of-devops

[33] Puppet Labs, "2017 State of DevOps Report," Puppet Labs, Tech. Rep., 2017. [Online]. Available: https://puppetlabs.com/solutions/devops/

[34] E. DeBoer, "Accelerate: A Principle-based DevOps Framework," 2019. [Online]. Available: https://blog.sonatype.com/principle-based-devops-frameworks-accelerate

[35] Puppet Labs, "2019 State of DevOps Report," Puppet Labs, Tech. Rep., 2019.

[36] S. Team, "Understanding 6 Essential DevOps Principles - SKILLOGIC Official Blog," 2018. [Online]. Available: https://skillogic.com/blog/understanding-6-essential-devops-principles/

[37] Google, "DevOps capabilities | DORA - Google Cloud," 2020. [Online]. Available: https://cloud.google.com/solutions/devops/capabilities

[38] Puppet Labs, "2020 State of DevOps Report," Puppet Labs, Tech. Rep., 2020. [Online]. Available: https://puppet.com/resources/report/2020-state-of-devops-report/

[39] I. T. Revolution, "24 Key Capabilities to Drive Improvement in Software Delivery - IT Revolution," 2020. [Online]. Available: https://itrevolution.com/24-key-capabilities-to-drive-improvement-in-software-delivery/

[40] H. Ketterer, "Leaner, Faster, and Better with DevOps," 2017. [Online]. Available: https://www.bcg.com/publications/2017/technology-digital-leaner-faster-better-devops

[41] Netapp, "What Is DevOps? - Practices and Benefits Explained | NetApp," pp. 1–12, 2019. [Online]. Available: https://www.netapp.com/devops-solutions/what-is-devops/

[42] J. Wade, "Devops Best Practices Checklist · GitHub," pp. 1–8, 2017. [Online]. Available: https://gist.github.com/jpswade/4135841363e72ece8086146bd7bb5d91

[43] R. de Feijter, S. Overbeek, R. van Vliet, E. Jagroep, and S. Brinkkemper, "DevOps competences and maturity for software producing organizations," *Lecture Notes in Business Information Processing*, vol. 318, pp. 244–259, 2018.

[44] Vilmate, "7 DevOps Best Practices for a Project Success | Vilmate," 2020. [Online]. Available: https://vilmate.com/blog/devops-best-practices/

[45] RedHat, "What is DevSecOps?" 2018. [Online]. Available: https://www.redhat.com/en/topics/devops/what-is-devsecops

[46] N. Forsgren, J. Humble, and G. Kim, *Accelerate: The science of lean software and devops: Building and scaling high performing technology organizations*. IT Revolution, 2018.

[47] D. Teixeira, R. Pereira, T. A. Henriques, M. Silva, and J. Faustino, "A Systematic Literature Review on DevOps Capabilities and Areas," *International Journal of Human Capital and Information Technology Professionals*, vol. 11, no. 2, pp. 1–22, apr 2020.

[48] I. Karamitsos, S. Albarhami, and C. Apostolopoulos, "Applying devops practices of continuous automation for machine learning," *Information (Switzerland)*, vol. 11, no. 7, pp. 1–15, jul 2020.

[49] J. Faustino, "DevOps Practices in Incident Management Process," Thesis, ISCTE-IUL, 2018. [Online]. Available: https://repositorio.iscte-iul.pt/bitstream/10071/18294/1/Master_Joao_Carvalho_Faustino.pdf

[50] DORA, "DORA research program," 2020. [Online]. Available: https://www.devops-research.com/research.html

[51] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Computer Science Review*, vol. 38, no. 1, p. 100308, nov 2020.

[52] G. Kim, J. Humble, P. Debois, and J. Willis, *The DevOps Handbook : How to Create World-Class Agility, Reliability, and Security in Technology Organizations*. IT Revolution Press, 2016.

[53] N. Ceresani, "The 4 Core Capabilities of DevOps," 2017. [Online]. Available: https://dzone.com/articles/the-4-core-capabilities-of-devops

[54] Puppet Labs, "2014 State of DevOps Report," Puppet Labs, Tech. Rep., 2014. [Online]. Available: http://puppetlabs.com/2014-devops-report

[55] L. Bass, Len; Weber, Ingo; Zhu, *DevOps: A Software Architect's Perspective*, 1st ed. Addison-Wesley Professional, 2015.

[56] I. Sacolick, "15 KPIs to track devops transformation," 2018. [Online]. Available: https://www.infoworld.com/article/3297041/15-kpis-to-track-devops-transformation.html

[57] Puppet Labs, "2013 State of DevOps Report," Puppet Labs, Tech. Rep., 2013.

[58] H. T. C. G. Services, "24 DevOps Capabilities - DevOps Efficiency Matrix," 2020. [Online]. Available: https://devopsefficiency.com/list.html

[59] C. A. Technologies, "How Can DevOps Practices Help You Increase - Broadcomdocs.broadcom.com," 2017. [Online]. Available: https://docs.broadcom.com/doc/how-can-devops-practices-help-you-increase-innovation-velocity-and-business-agility-on-the-mainframe

[60] DevOps Research and Assessment (DORA), D. Research, and A. (DORA), "State of DevOps 2019 - DORA," DORA, Tech. Rep., 2019. [Online]. Available: https://services.google.com/fh/files/misc/state-of-devops-2019.pdf

[61] K. Mikko, "DevOps Capability Assessment in a Software Development Team," *Science*, vol. 135, pp. 408–415, 2018. [Online]. Available: https://www.theseus.fi/bitstream/handle/10024/334710/MikkoKurkela_DevOpsCapabilityAssessmentInASoftwareDevelopmentTeam.pdf

[62] Puppet Labs, "2018 State of DevOps Report," Puppet Labs, Tech. Rep., 2018. [Online]. Available: https://media.webteam.puppet.com/uploads/2019/11/Puppet-State-of-DevOps-Report-2018_update.pdf

[63] B. I. Staff, "10 DevOps Best Practices To Know | Built In," pp. 1–7, 2020. [Online]. Available: https://builtin.com/software-engineering-perspectives/devops-best-practices

[64] I. Pavlenko, "DevOps: Principles, Practices, and DevOps Engineer Role," 2021. [Online]. Available: https://www.altexsoft.com/blog/engineering/devops-principles-practices-and-devops-engineer-role/

[65] C. Us and StarAgile, "Top 7 DevOps practices for Successful Implementation of DevOps," pp. 1–8, 2020. [Online]. Available: https://staragile.com/blog/devops-best-practices

[66] G. B. Ghantous and A. Q. Gill, "DevOps: Concepts, practices, tools, benefits and challenges," in *Proceedings ot the 21st Pacific Asia Conference on Information Systems: "Societal Transformation Through IS/IT", PACIS 2017*. Association for Information Systems, 2017, p. 12.

[67] K. Eby, "The Tools and Technology of DevOps | Smartsheet," pp. 1–26, 2017. [Online]. Available: https://www.smartsheet.com/devops-tools

[68] C. Crowley, L. McQuillan, and C. O'Brien, "Understanding DevOps: Exploring the origins, composition, merits, and perils of a DevOps Capability," in *Proceedings of the 4th International Conference on Production Economics and Project Evaluation, ICOPEV 2018, Guimarães, Portugal*, ser. ICOPEV International Conference on Project Economic Evaluation, M. Araujo, Ed., Univ Minho, Sch Engn, Ind Engn and Management, Algoritmi Res Ctr. Guimaraes: Universade do Minho, 2018, Proceedings Paper, pp. 29–37.

[69] Scaledagile, "DevOps - Scaled Agile Framework," 2021. [Online]. Available: https://www.scaledagileframework.com/devops/

[70] Wikipedia, "DevOps - Wikipedia," 2021. [Online]. Available: https://en.wikipedia.org/wiki/DevOps

[71] E. Mueller, "What Is DevOps? | the agile admin," 2019. [Online]. Available: https://theagileadmin.com/what-is-devops/

[72] T. Hall, "DevOps Best Practices | Atlassian," 2020. [Online]. Available: https://www.atlassian.com/devops/what-is-devops/devops-best-practices

[73] Veritis, "DevOps Capabilities: 6-point Principles for Business Success," 2018. [Online]. Available: https://www.veritis.com/blog/devops-capabilities-a-6-point-principle-that-drives-business-success/

[74] D. Teixeira, R. Pereira, T. Henriques, M. M. D. Silva, J. Faustino, O. Faustino, and M. Silva, "A maturity model for DevOps," *International Journal of Agile Systems and Management*, vol. 13, no. 4, p. 464, 2020.

[75] M. Zulfahmi Toh, S. Sahibuddin, and M. N. Mahrin, "Adoption issues in DevOps from the perspective of continuous delivery pipeline," in *ACM International Conference Proceeding Series*, ser. ICSCA '19, vol. Part F1479. New York, NY, USA: Association for Computing Machinery, 2019, pp. 173–177.

[76] A. Lichtenberger, "Blog: Agile: Dead End? Taking the next step by applying DevOps Practices effectively - impact matters Blog," 2019. [Online]. Available: https://www.impactmatters.ch/blog/agiledevops-deadend/

[77] Department of Energy Quality Managers: Software Quality Assurance Subcommittee, "Software Configuration Management (SCM) A Practical Guide," United States Department of Energy, Tech. Rep., 2000. [Online]. Available: http://energy.gov/sites/prod/files/cioprod/documents/scmguide.pdf

[78] ANSI/IEEE, "IEEE Std 1042-1987 Guide to Software Configuration Management," 1987. [Online]. Available: http://ieeexplore.ieee.org/articleDetails.jsp?arnumber=89631

[79] P. M. Mell and T. Grance, "The NIST definition of cloud computing," National Institute of Standards and Technology, Gaithersburg, MD, Tech. Rep., 2011. [Online]. Available: https://nvlpubs.nist.gov/nistpubs/Legacy/SP/nistspecialpublication800-145.pdf

[80] D. Stahl, T. Martensson, and J. Bosch, "Continuous practices and devops: beyond the buzz, what does it all mean?" in *2017 43rd Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, aug 2017, pp. 440–448.

[81] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture," *IEEE Software*, vol. 33, no. 3, pp. 42–52, may 2016.

[82] T. F. Düllmann, C. Paule, and A. Van Hoorn, "Exploiting devops practices for dependable and secure continuous delivery pipelines," in *Proceedings - International Conference on Software Engineering*, IEEE Comp Soc; Assoc Comp Machinery; SIGSOFT; IEEE Tech Council Software Engn. New York, NY, USA: Association for Computing Machinery, 2018, Proceedings Paper, pp. 27–30.

[83] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Migrating to Cloud-Native Architectures Using Microservices: An Experience Report," *Communications in Computer and Information Science*, vol. 567, pp. 201–215, jul 2015.

[84] M. Stillwell and J. G. F. Coutinho, "A DevOps approach to integration of software components in an EU research project," in *Proceedings of the 1st International Workshop on Quality-Aware DevOps*, Imperial College London United Kingdom. New York, NY, USA: ACM, sep 2015, pp. 1–6.

[85] R. Souza, F. Silva, L. Rocha, and I. Machado, "Investigating agile practices in software startups," *ACM International Conference Proceeding Series*, pp. 317–321, 2019. [Online]. Available: http://doi.org/10.1145/3350768.3350786 https://doi.org/10.1145/3350768.3350786

[86] A. Steffens, H. Lichter, and J. S. Döring, "Designing a next-generation continuous software delivery system: Concepts and architecture," in *Proceedings - International Conference on Software Engineering*, ser. RCoSE '18. New York, NY, USA: Association for Computing Machinery, 2018, pp. 1–7.

[87] BoxBoat, "What is DevOps? Exploring DevOps Principles & Benefits | BoxBoat," 2018. [Online]. Available: https://boxboat.com/2018/12/26/what-is-devops/

[88] M. E. Pasten, M. Siyam, and D. Academy, "Role-based DevOps Capability Model," pp. 1–12, 2020. [Online]. Available: https://devonacademy.com/enterprise-devops-capability-model/

[89] C. Marnewick and J. Langerman, "DevOps and Organisational Performance: The Fallacy of Chasing Maturity," *IEEE Software*, pp. 0–0, 2020. [Online]. Available: https://ieeexplore.ieee.org/document/9190017/

[90] R. d. F. R. v. V. E. J. S. O. S. Brinkkemper, "Towards the adoption of DevOps in software product organizations: A maturity model approach Rico," *Journal of Chemical Information and Modeling*, vol. 53, 2019.

[91] A. Valdes, "5 DevOps Metrics and KPIs that CTOs Must Monitor," pp. 1–9, 2020. [Online]. Available: https://www.clickittech.com/devops/devops-metrics-and-kpis/

[92] V. Fedak, "DevOps metrics: what to track, how and why do it." pp. 1–11, 2020. [Online]. Available: https://medium.com/@FedakV/devops-metrics-what-to-track-how-and-why-do-it-e08dc6864eab

[93] E. Lock, "Measure DevOps Metrics That Matter," 2020. [Online]. Available: https://www.devopsdigest.com/measure-devops-metrics-that-matter

[94] D. Sato, "Practices for DevOps and Continuous Delivery," 2015. [Online]. Available: https://www.infoq.com/articles/book-DevOps-continuous-delivery/

[95] Devopedia, "DevOps Metrics," 2019. [Online]. Available: https://devopedia.org/devops-metrics

[96] G. Motroc, "Key DevOps metrics that matter: How well does your team sleep?" pp. 1–13, 2018. [Online]. Available: https://jaxenter.com/devops-influencers-interview-series-4-142312.html

[97] AWS, "What is DevOps? - Amazon Web Services (AWS)," 2021. [Online]. Available: https://aws.amazon.com/devops/what-is-devops/

[98] CMMI Product Team, "CMMI for Development, Version 1.3," *Software Engineering Process Management Program*, 2010.

[99] N. Forsgren, J. Humble, G. Kim, A. Brown, N. Kersten, Dr. Nicole Forsgren, Jez Humble, and Gene Kim, "Accelerate: State of DevOps 2018 Strategies for a New Economy," *Report. DevOps Research & Assessment (DORA)*, p. 78, 2018. [Online]. Available: https://cloudplatformonline.com/rs/248-TPC-286/images/DORA-State of DevOps.pdf

[100] L. Yin and V. Filkov, "Team Discussions and Dynamics during DevOps Tool Adoptions in OSS Projects," in *Proceedings - 2020 35th IEEE/ACM International Conference on Automated Software Engineering, ASE 2020*, ser. ASE '20. New York, NY, USA: Association for Computing Machinery, 2020, pp. 697–708.

[101] Intellipaat, "What is DevOps - Introduction to DevOps Architecture & Benefits," 2016. [Online]. Available: https://intellipaat.com/blog/what-is-devops/

[102] H. Packard, "Measuring DevOps success," 2016. [Online]. Available: http://www.baldrover.com/wp-content/uploads/Measuring-DevOps-Success.pdf

[103] H. Myrbakken and R. Colomo-Palacios, "DevSecOps: A multivocal literature review," *Communications in Computer and Information Science*, vol. 770, no. 1, pp. 17–29, 2017.

[104] J. Humble and B. O'Reilly, *Lean enterprise : adopting continouos delivery, devops, and lean startup at scale*. O'Reilly Media, Inc., 2014.

[105] L. E. Lwakatare, T. Kilamo, T. Karvonen, T. Sauvola, V. Heikkilä, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, C. Lassenius, V. Heikkila, J. Itkonen, P. Kuvaja, T. Mikkonen, M. Oivo, and C. Lassenius, "DevOps in practice: A multiple case study of five companies," *Information and Software Technology*, vol. 114, no. March 2017, pp. 217–230, oct 2019.

[106] M. Shahin, M. Ali Babar, and L. Zhu, "Continuous Integration, Delivery and Deployment: A Systematic Review on Approaches, Tools, Challenges and Practices," *IEEE Access*, vol. 5, pp. 3909–3943, 2017. [Online]. Available: http://ieeexplore.ieee.org/document/7884954/

[107] A. R. Chowdhury, "DevOps Best Practices: A Complete Guide," 2019. [Online]. Available: https://stackify.com/devops-best-practices-a-complete-guide/

[108] K. Wakayama, "How to Ensure the Success of DevOps in Your Organization," pp. 1–10, 2020. [Online]. Available: https://codersociety.com/blog/articles/devops-success-in-organization

[109] Ubiq, "Top DevOps Metrics and KPIs To Monitor Regularly - Ubiq BI Blog," pp. 1–12, 2020. [Online]. Available: http://ubiq.co/analytics-blog/top-devops-metrics-kpis-to-monitor-regularly/

[110] R. Westrum, "A typology of organisational cultures," *Quality and Safety in Health Care*, vol. 13, no. SUPPL. 2, pp. 22–27, 2004. [Online]. Available: www.qshc.com

[111] Puppet Labs, "2016 State of DevOps Report," Puppet Labs, Tech. Rep. 7, 2016. [Online]. Available: https://puppetlabs.com/solutions/devops/

[112] A. Wiedemann, N. Forsgren, M. Wiesche, H. Gewald, and H. Krcmar, "Research for practice: The Devops phenomenon," *Communications of the ACM*, vol. 62, no. 8, pp. 44–49, 2019.

[113] M. Outlaw, "The DevOps Handbook – The Technical Practices of Flow," pp. 1–7, 2020. [Online]. Available: https://www.codingblocks.net/podcast/the-devops-handbook-the-technical-practices-of-feedback/

[114] D. Cukier, "DevOps patterns to scale web applications using cloud services," in *SPLASH 2013 - Proceedings of the 2013 Companion Publication for Conference on Systems, Programming, and Applications: Software for Humanity*. New York, New York, USA: ACM Press, 2013, pp. 143–152. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2508075.2508432

[115] N. Forsgren, M. C. Tremblay, D. VanderMeer, and J. Humble, "DORA Platform: DevOps Assessment and Benchmarking," in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*. Springer, 2017, pp. 436–440.

[116] Veritis, C. Services, D. D. T. Services, D. D. T. Services, A. Management, M. Services, T. Advisory, F. Services, R. Estate, C. Studies, W. Papers, and Veritis, "Measuring DevOps: Key 'Metrics' and 'KPIs' That Drive Success!" pp. 1–10, 2020. [Online]. Available: https://www.veritis.com/blog/measuring-devops-key-metrics-and-kpis-that-drive-success/

[117] N. Tomas, J. Li, and H. Huang, "An empirical study on culture, automation, measurement, and sharing of DevSecOps," in *2019 International Conference on Cyber Security and Protection of Digital Services, Cyber Security 2019*, jun 2019, pp. 1–8.

[118] S. I. Mohamed, "DevOps Maturity Calculator DOMC -Value oriented approach," *International Journal of Engineering Research and Science*, vol. 2, no. 2, pp. 2395–6992, 2016.

[119] P. Waterhouse, "DevOps Practitioner Series - Metrics That Matter," 2015. [Online]. Available: https://docs.broadcom.com/doc/devops-practitioner-series-metrics-that-matter-developing-and-tracking-key-indicators

[120] S. Smith, "High performing DevOps metrics," pp. 1–11, 2020. [Online]. Available: https://samlearnsazure.blog/2020/04/30/high-performing-devops-metrics/

[121] V. Gupta, P. K. Kapur, and D. Kumar, "Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling," *Information and Software Technology*, vol. 92, no. 1, pp. 75–91, 2017. [Online]. Available: http://search.ebscohost.com/login.aspx?direct=true&db=eoah&AN=42845256&lang=pt-pt&site=ehost-live&scope=site

[122] U. S. Bititci, P. Garengo, A. Ates, and S. S. Nudurupati, "Value of maturity models in performance measurement," *International Journal of Production Research*, vol. 53, no. 10, pp. 3062–3085, may 2015.

[123] G. Kim, K. Behr, K. Spafford, and G. Spafford, *The phoenix project: A novel about IT, DevOps, and helping your business win*. IT Revolution, 2014. [Online]. Available: https://books.google.pt/books?id=H6x-DwAAQBA

[124] RDC Partner, "6 core capabilities of DevOps practice," pp. 2–6, 2018. [Online]. Available: https://rdcpartner.nl/6-core-capabilities-of-devops-practice/

[125] C. Rudder, "10 ways DevOps helps digital transformation | The Enterprisers Project," 2019. [Online]. Available: https://enterprisersproject.com/article/2019/8/devops-role-digital-transformation

[126] S. Yusuf, "Ebbs and Flows Of DevOps Debugging PART 1 - Security Boulevard," pp. 1–32, 2021. [Online]. Available: https://securityboulevard.com/2021/02/ebbs-and-flows-of-devops-debugging-part-1/

[127] Puppet, "5 Foundational DevOps Practices: How to Establish & Build on Them | Puppet," 2018. [Online]. Available: https://puppet.com/resources/whitepaper/5-foundational-devops-practices-how-establish-build-them/

[128] T. Jachja, "CI/CD Patterns and Practices — DevOps Institute," 2020. [Online]. Available: https://devopsinstitute.com/ci-cd-patterns-and-practices/

[129] C. Patel, "DevOps Best Practices - DZone DevOpsdzone.com," 2020. [Online]. Available: https://dzone.com/articles/devops-best-practices

[130] C. S. Tov, "DevOps Best Practices: Take Your Pipeline to the Next Level - Codemotion," 2020. [Online]. Available: https://www.codemotion.com/magazine/dev-hub/devops-engineer/devops-best-practices/

[131] R. Powell, "Essential DevOps Principles for 2021 | CircleCIcircleci.com," 2020. [Online]. Available: https://circleci.com/blog/essential-devops-principles/

[132] M. Patil, "Maximizing DevOps ROI with Modern Application Practices | HCL Blogs," 2020. [Online]. Available: https://www.hcltech.com/blogs/maximizing-devops-roi-modern-application-practices

[133] BMC, "State of DevOps 2020: A Report Roundup," *BMC*, 2020. [Online]. Available: https://www.bmc.com/blogs/state-of-devops/

[134] D. Online, "The Practice of DevOps," 2020. [Online]. Available: https://www.dqindia.com/the-practice-of-devops/

[135] C. Solutions, "8 Best Practices for Successful Implementation of DevOps," 2019. [Online]. Available: https://dev.to/credencys/8-best-practices-for-successful-implementation-of-devops-in-your-enterprise-2k93

[136] G. Pousseo, "DevOps Practices: Agility without DevOps is pointless," 2019. [Online]. Available: https://www.pentalog.com/blog/agility-without-devops-is-pointless

[137] A. D. Rayome, "5 foundational DevOps practices your enterprise needs to succeed - TechRepublic," 2018. [Online]. Available: https://www.techrepublic.com/article/5-foundational-devops-practices-your-enterprise-needs-to-succeed/

[138] K. Kuusinen, V. Balakumar, S. C. Jepsen, S. H. Larsen, T. A. Lemqvist, A. Muric, A. Ø. Nielsen, and O. Vestergaard, "A large agile organization on its journey towards DevOps," in *Proceedings - 44th Euromicro Conference on Software Engineering and Advanced Applications, SEAA 2018*. Institute of Electrical and Electronics Engineers Inc., aug 2018, pp. 60–63.

[139] P. CIO, "Building a Strong DevOps Foundation | CIO," 2018. [Online]. Available: https://www.cio.com/article/3319077/building-a-strong-devops-foundation.html

[140] N. Stamenkovic, "DevOps - principles, practices and why should you care? - Ingsoftware Blog," pp. 1–11, 2018. [Online]. Available: https://www.ingsoftware.com/devops-principles-and-practices

[141] Pinkelephant, "DevOps: Why The 15 Essential Practices Are Key To Your Organization's Survival | Pink Elephant Blog," 2018. [Online]. Available: https://blog.pinkelephant.com/blog/devops-why-the-15-essential-practices-are-key-to-your-organizations-surviva

[142] PuppetSplunk, "The 5 foundational devops practices," 2018. [Online]. Available: https://www.1sttechguide.com/wp-content/uploads/2019/09/The-5-Foundational-DevOps-Practices.pdf

[143] A. Crouch, A. Gaspari, and A. Crouch, "6 Steps to a Successful DevOps Adoption | AgileConnection," 2017. [Online]. Available: https://www.agileconnection.com/article/6-steps-successful-devops-adoption

[144] C. Pang and A. Hindle, "Continuous maintenance," in *Proceedings - 2016 IEEE International Conference on Software Maintenance and Evolution, ICSME 2016*. Institute of Electrical and Electronics Engineers Inc., oct 2017, pp. 458–462.

[145] A. Wadhera, "10 Key DevOps Practices to Improve IT Efficiency," 2016. [Online]. Available: https://www.tothenew.com/blog/10-key-devops-practices-to-improve-it-efficiency/

[146] L. Zhu, L. Bass, and G. Champlin-Scharff, "DevOps and Its Practices," *IEEE Software*, vol. 33, no. 3, pp. 32–34, may 2016.

[147] D. Linthicum, “DevOps tools best practices: A 7-step guide,” 2016. [Online]. Available: https://techbeacon.com/devops/7-steps-choosing-right-devops-tools
[148] K. Horvath, “DevOps Part 2: Methods, Practices and Tools,” 2015. [Online]. Available: https://content.intland.com/blog/agile/devops/devops-part-2-methods-practices-and-tools
[149] M. Croker, “DevOps: innovative engineering practices for continuous software delivery,” 2015. [Online]. Available: https://www.accenture.com/_acnmedia/PDF-18/Accenture-DevOps-brochure-new.pdf
[150] M. A. Silva, J. P. Faustino, R. Pereira, and M. Mira da Silva, “Productivity gains of DevOps adoption in an IT team: A case study,” in *Proceedings of the 27th International Conference on Information Systems Development: Designing Digitalization, ISD 2018*, B. C. L. M. L. H. S. C. Andersson B. Johansson B., Ed. Association for Information Systems, jul 2018, p. 12. [Online]. Available: https://repositorio.iscte-iul.pt/handle/10071/16388
[151] M. Rother, *Toyota Kata: Managing people for improvement, adaptiveness and superior results*. MGH, New York, 2019.
[152] Cambridge, “CAPABILITY | meaning in the Cambridge English Dictionary,” 2021. [Online]. Available: https://dictionary.cambridge.org/dictionary/english/capability
[153] Merriam-Webster, “Capability | Definition of Capability by Merriam-Webster,” 2021. [Online]. Available: https://www.merriam-webster.com/dictionary/capability
[154] T. Biggs, M. Shah, and P. Srivastava, *Technological capabilities and learning in African enterprises*. The World Bank, 1995.
[155] Flosum, “Keys to Improve Salesforce DevOps Efficiency - Flosum - Continuous Integration, release management,” 2021. [Online]. Available: https://flosum.com/keys-to-improve-salesforce-devops-efficiency/
[156] R. Edwards, “Research highlights challenges of Salesforce DevOps in 2020 -,” 2021. [Online]. Available: https://www.enterprisetimes.co.uk/2021/02/16/research-highlights-challenges-of-salesforce-devops-in-2020/
[157] Digital.ai, “10 DevOps Metrics You Should Know,” 2020. [Online]. Available: https://digital.ai/resources/infographic/10-devops-metrics-you-should-know
[158] S. Miteva, “13 DevOps Metrics for Increased Productivity,” pp. 1–9, 2020. [Online]. Available: https://dev.to/microtica/13-devops-metrics-for-increased-productivity-5084
[159] D. Kodjamanova, “4 DevOps Metrics To Maximize Success - MentorMate,” pp. 1–20, 2020. [Online]. Available: https://mentormate.com/blog/how-devops-metrics-pave-the-way-to-better-performance/
[160] T. Development, “A Guide To Measuring DevOps Success an Proving ROI,” 2020. [Online]. Available: https://www.tiempodev.com/blog/measuring-devops/
[161] N. Forsgren, M. A. Rothenberger, J. Humble, J. B. Thatcher, and D. Smith, “A taxonomy of software delivery performance profiles: Investigating the effects of devops practices,” *26th Americas Conference on Information Systems, AMCIS 2020*, pp. 0–5, 2020. [Online]. Available: https://www.scopus.com/inward/record.uri?eid=2-s2.0-85097711091&partnerID=40&md5=7c2dbc3859bd41e964feac16170bd302
[162] R. Honig, P. Page, K. Base, P. Page, K. Base, O. P. Debugger, O. Visual, S. Extension, P. Page, K. Base, P. Page, K. Base, O. P. Debugger, O. Visual, S. Extension, O. P. Debugger, O. Visual, S. Extension, O. P. Debugger, O. Visual, S. Extension, and R. Honig, “Bridge the DevOps Development Chasm to Boost DevOps KPIs - Ozcode,” pp. 1–14, 2020. [Online]. Available: https://oz-code.com/blog/devops/bridging-the-devops-development-observability-chasm-to-boost-devops-kpis
[163] Stackexchange, D. S. Exchange, I. Management, C. Management, R. M. To, R. C. Analysis, Stackexchange, and M. Hossain, “What key performance indicators (KPIs) are used to measure DevOps?” pp. 30–32, 2020. [Online]. Available: https://devops.stackexchange.com/questions/738/what-key-performance-indicators-kpis-are-used-to-measure-devops
[164] R. Pruess, “DevOps Best Practices: 5 Key Performance Indicators,” pp. 1–7, 2020. [Online]. Available: https://flexagon.com/devops-best-practices-5-key-performance-indicators/
[165] K. Ahmad, “DevOps KPIs and "Design for Failure",” pp. 1–4, 2020. [Online]. Available: https://www.linkedin.com/pulse/devops-kpis-design-failure-khalil-ahmad
[166] Opsgenie and O. Support, “DevOps Metrics,” pp. 1–15, 2020. [Online]. Available: https://docs.opsgenie.com/docs/devops-metrics-global
[167] T. Nero, “DevOps Metrics : 15 KPIs that Boost Results and RoI - Cuelogic Technologies Pvt. Ltd.” 2020. [Online]. Available: https://www.cuelogic.com/blog/devops-metrics
[168] Appdynamics, “DevOps Metrics and KPIs: How To Measure DevOps?” pp. 1–14, 2020. [Online]. Available: https://www.appdynamics.com/topics/devops-metrics-and-kpis
[169] J. Kernel, “DevOps Metrics: 7 KPIs to Evaluate Your Team’s Maturity,” 2020. [Online]. Available: https://www.xplg.com/devops-metrics-7-kpis/
[170] ReleaseTEAM, “DevOps Metrics: Measure Your DevOps Results,” 2020. [Online]. Available: https://www.releaseteam.com/measure-your-devops-results/
[171] G. Logic, “DevOps Success: What to Measure and Why - Gorilla Logic,” pp. 1–6, 2020. [Online]. Available: https://gorillalogic.com/blog/devops-success-what-to-measure-and-why/
[172] Plutora, “DORA DevOps Metrics - Accelerate your Value Stream - Plutora.com,” 2020. [Online]. Available: https://www.plutora.com/resources/videos/devops-dora-metrics
[173] Gitlab, “Getting Started with Agile/DevOps Metrics | GitLab,” 2020. [Online]. Available: https://about.gitlab.com/handbook/marketing/strategic-marketing/devops-metrics/
[174] J. Riggins, “Google’s Formula for Elite DevOps Performance – The New Stack,” 2020. [Online]. Available: https://thenewstack.io/googles-formula-for-elite-devops-performance/
[175] R. Powell, “How to measure DevOps success: 4 key metrics,” 2020. [Online]. Available: https://circleci.com/blog/how-to-measure-devops-success-4-key-metrics/
[176] G. Guimarães, “On the Four Key DevOps Metrics, and why I measure them differently – SourceLevel,” 2020. [Online]. Available: https://sourcelevel.io/blog/on-the-four-key-devops-metrics-and-why-i-measure-them-differently
[177] A. Ramesh, “Ten Key DevOps Metrics to Accelerate Your Continuous Delivery Pipeline,” pp. 1–6, 2020. [Online]. Available: https://www.go2group.com/resources/blog/devops-metrics-to-accelerate-ci-cd/
[178] C. Lawrence, “The Four Key Metrics of DevOps,” pp. 1–16, 2020. [Online]. Available: https://humanitec.com/blog/devops-key-metrics
[179] P. Gallagher, “Tracking Success in DevOps Pipelines,” pp. 1–14, 2020. [Online]. Available: https://blog.goodelearning.com/subject-areas/devops/how-to-measure-success-in-devops/
[180] D. G. Portman, “Using the Four Keys to measure your DevOps performance,” 2020. [Online]. Available: https://cloud.google.com/blog/products/devops-sre/using-the-four-keys-to-measure-your-devops-performance
[181] M. Hossain, “What key performance indicators (KPIs) are used to measure DevOps?” 2020. [Online]. Available: https://www.quora.com/What-key-performance-indicators-KPIs-are-used-to-measure-DevOps
[182] O. Nasser, “What Metrics Should DevOps Teams Be Tracking?” pp. 1–5, 2020. [Online]. Available: https://cto.ai/blog/what-metrics-should-devops-teams-be-tracking/
[183] B. Dobran, “15 DevOps Metrics and Key Performance Indicators (KPIs) To Track,” 2019. [Online]. Available: https://phoenixnap.com/blog/devops-metrics-kpis
[184] M. Bizzantino, “4 fundamental metrics to measure DevOps performances,” 2019. [Online]. Available: https://www.kiratech.it/en/blog/4-fundamental-metrics-to-measure-devops-performances
[185] D. Dingley, V. S. As, A. Coach, V. Solutions, D. Dingley, I. Team, and H.-c. Interaction, “4 Key Metrics for DevOps Success Video,” 2019. [Online]. Available: https://www.veracitysolutions.com/4-key-metrics-for-devops-success
[186] V. Kamani, “7 crucial DevOps metrics that you need to track,” pp. 1–8, 2019. [Online]. Available: https://hub.packtpub.com/7-crucial-devops-metrics-that-you-need-to-track/
[187] L. Cīrule, “Analyze DevOps Metrics With eazyBI,” pp. 1–8, 2019. [Online]. Available: https://eazybi.com/blog/analyze-devops-metrics-with-eazybi
[188] O. Habib, B. Home, C. Delivery, C. Integration, C. C. Management, S. Page, G. Started, and O. Habib, “DevOps Accelerate Metrics | Harness Platform-as-a-Service,” pp. 1–13, 2019. [Online]. Available: https://harness.io/blog/featured/continuous-insights-accelerate-devops-metrics/
[189] R. Cyber, “DevOps KPIs to Measure Success,” 2019. [Online]. Available: https://www.royalcyber.com/blog/devops/devops-kpis-to-measure-success/
[190] M. Baldani, “DORA Metrics - Getting on the Bandwagon,” pp. 1–5, 2019. [Online]. Available: https://www.cloudbees.com/blog/dora-metrics-getting-bandwagon
[191] M. Samuel, “How to Successfully Scale Agile and DevOps – Part 3: Driving Success with Technology,” 2019. [Online]. Available: https://www.hcltech.com/blogs/how-successfully-scale-agile-and-devops-part-3-driving-success-technology
[192] R. Shinde, “Is your DevOps successful?” 2019. [Online]. Available: https://www.accenture.com/us-en/blogs/software-engineering-blog/reshma-shinde-devops-success-metrics
[193] G. Singh, D. Transformation, S. Providers, H. Multi-cloud, E. H. M.-c. Environment, D. Intelligence, E. D. Strategy, E. D. Strategy, D. I. Management, P. Strategy, E. Design, E. Devops, M. Cyber, and S. Solutions, “Measuring DevOps Success with DevOps Metrics,” pp. 1–14, 2019. [Online]. Available: https://www.xenonstack.com/blog/devops-metrics/
[194] A. Hawkins, “Measuring DevOps Success: What, Where, and How - Cloud Academy,” pp. 1–16, 2019. [Online]. Available: https://cloudacademy.com/blog/measuring-devops-success-what-where-and-how/
[195] A. Schurr, “Mobile App DevOps Metrics that Matter - NowSecure,” 2019. [Online]. Available: https://www.nowsecure.com/blog/2019/02/27/mobile-app-devops-metrics-that-matter/

[196] TestEnvMgt, "Top 5 DevOps Metrics – Test Environment Management," pp. 1–8, 2019. [Online]. Available: https://www.testenvironmentmanagement.com/top-5-devops-metrics/
[197] D. Holloran, "Top Metrics for Measuring DevOps Delivery Value," pp. 1–12, 2019. [Online]. Available: https://victorops.com/blog/top-metrics-for-measuring-devops-delivery-value
[198] F. J. Moraes, "DevOps KPI in Practice — Chapter 1 — Deployment Speed, Frequency and Failure," pp. 1–18, 2018. [Online]. Available: https://medium.com/@fabiojose/devops-kpi-in-practice-chapter-1-deployment-speed-frequency-and-failure-2fd0a9303249
[199] T. Gilmore, D. Toolchain, and D. Metrics, "DevOps Metrics - ADAPT Model Community," 2018. [Online]. Available: http://www.adapttransformation.com/devops-toolchain/monitor/devops-metrics/
[200] R. Bobbett, "DevOps Value: How to Measure the Success of DevOps," pp. 1–6, 2018. [Online]. Available: https://www.fpcomplete.com/blog/devops-value-how-to-measure-the-success-of-devops/
[201] N. Relic, "Measuring DevOps," 2018. [Online]. Available: https://newrelic.com/devops/measuring-devops
[202] P. Duvall, "Measuring DevOps Success with Four Key Metrics | Stelligent," pp. 1–5, 2018. [Online]. Available: https://stelligent.com/2018/12/21/measuring-devops-success-with-four-key-metrics/
[203] AlertOps, "MTTD vs MTTF vs MTBF vs MTTR - Resolve Major IT Incidents Quickly," 2018. [Online]. Available: https://alertops.com/mttd-vs-mttf-vs-mtbf-vs-mttr/
[204] H. Vora, "Software Quality Metrics for Agile and DevOps success - QMetry," pp. 1–14, 2018. [Online]. Available: https://www.qmetry.com/blog/software-quality-metrics-for-agile-and-devops-success/
[205] M. Watson, "15 Metrics for DevOps Success," pp. 1–19, 2017. [Online]. Available: https://stackify.com/15-metrics-for-devops-success/
[206] L. Karam, "DevOps metrics you must take into account," 2017. [Online]. Available: https://apiumhub.com/tech-blog-barcelona/devops-metrics/
[207] S. Watts, "DevOps: Metrics and Key Performance Indicators (KPIs)," 2017. [Online]. Available: https://itchronicles.com/devops/devops-metrics-kpis/
[208] DevOpsEnterpriseSummit, "Featured Resource: Metrics for DevOps Initiatives - IT Revolution," 2017. [Online]. Available: https://itrevolution.com/devops-resource-metrics/
[209] N. Forsgren, "How to use metrics, measurement to drive DevOps," 2017. [Online]. Available: https://techbeacon.com/devops/how-use-metrics-measurement-drive-devops
[210] J. A. X. London, "Measuring DevOps: The Key Metrics That Matter - JAX London," 2017. [Online]. Available: https://jaxlondon.com/blog/devops-continuous-delivery/measuring-devops-key-metrics-matter/
[211] R. Technology, "Seven Metrics That Matter When Measuring DevOps Success," 2017. [Online]. Available: https://www.aternity.com/blogs/seven-metrics-matter-measuring-devops-success/
[212] C. Shabe, "Understanding DevOps metrics," pp. 1–5, 2017. [Online]. Available: https://betanews.com/2017/08/24/devops-metrics/
[213] KnowledgeHut, "What are the metrics and why do they matter for DevOps Success?" 2017. [Online]. Available: https://www.knowledgehut.com/blog/agile/metrics-matters-devops-success
[214] T. Hall, "DevOps metrics | Atlassian," 2016. [Online]. Available: https://www.atlassian.com/devops/frameworks/devops-metrics
[215] C. Staff, A. Development, A. Testing, D. Engineering, A. D. Training, D. Training, S. S. T. Training, S. S. T. Training, E. U. Classes, Q. Starts, A. Coveros, O. Team, and C. Staff, "Essential Quantitative DevOps Metrics - Coveros," pp. 1–5, 2016. [Online]. Available: https://www.coveros.com/essential-quantitative-devops-metrics/
[216] Cigniti, "Top 6 DevOps Metrics that Enterprise Dashboards Should Capture," 2016. [Online]. Available: https://www.cigniti.com/blog/6-devops-metrics-for-enterprise-dashboards/
[217] Valtech, "4 metrics for measuring DevOps success," pp. 1–12, 2015. [Online]. Available: https://www.valtech.com/insights/4-metrics-for-measuring-devops-success/
[218] M. Michaelis, "DevOps Metrics - IntelliTect," 2015. [Online]. Available: https://intellitect.com/devops-metrics/
[219] P. Arora, B. Branches, and C. Coverage, "Measuring the Success of DevOps – Prashant Arora's Blog," 2015. [Online]. Available: https://aroraprashant.wordpress.com/2015/04/14/measuring-the-success-of-devops/
[220] N. Forsgren, "Metrics for DevOps Initiatives - nicole forsgren," 2015. [Online]. Available: https://nicolefv.com/s/DOES_forum_metrics_102015.pdf
[221] Pagerduty, "The Best Metrics for Driving Cultural Change in DevOps Teams," 2015. [Online]. Available: https://www.pagerduty.com/blog/best-metrics-devops-culture/
[222] A. Earnshaw, "5 KPIs that Make the Case for DevOps," 2013. [Online]. Available: https://puppet.com/blog/5-kpis-make-case-for-devops/
[223] J. Gelo, "DevOps Metrics Matter: Why, Which Ones, and How - HCL SW Blogs," 2020. [Online]. Available: https://blog.hcltechsw.com/accelerate/devops-metrics-matter-why-which-ones-and-how-2/
[224] A. Khanduri, "DevOps Metrics: Measuring What Matters," 2020. [Online]. Available: https://blog.sonatype.com/devops-metrics-measuring-what-matters
[225] P. Lab, "Metrics for Successful DevOps ?" 2020. [Online]. Available: https://performancelabus.com/successful-devops-metrics/
[226] Digital.ai, "4 DevOps Metrics to Improve Delivery Performance on Vimeo," pp. 1–5, 2019. [Online]. Available: https://vimeo.com/331032185
[227] S. Watts, "DevOps Metrics and KPIs – BMC Blogs," pp. 1–8, 2019. [Online]. Available: https://www.bmc.com/blogs/devops-kpi-metrics/
[228] M. Edwards, "Measuring for Success - Change or Hold - DevOps," 2019. [Online]. Available: https://www.tesm.com/resources-blog-measuring-for-success-should-you-change-or-should-you-hold-devops-pt-5/
[229] R. Osherove, "Ten Devops an Agility Metrics to Check at the Team Level - Pipeline Driven," pp. 1–3, 2018. [Online]. Available: https://pipelinedriven.org/article/ten-ideas-for-things-you-can-measure-as-a-team-on-your-devops-journey
[230] H. Idan, "The Must Have Metrics Any DevOps and SRE Manager Should Measure," pp. 1–8, 2018. [Online]. Available: https://www.overops.com/blog/the-must-have-metrics-any-devops-and-sre-manager-should-measure/
[231] B. Gracely, "The Most Important DevOps Metric to Measure," 2017. [Online]. Available: https://www.openshift.com/blog/important-devops-metric-measure
[232] K. de Boer, "2 Most Important DevOps Metrics Tools," 2016. [Online]. Available: https://labs.sogeti.com/the-two-most-important-metrics-for-devops/
[233] D. Better, S. Faster, and A. Weiss, "Measuring DevOps Flow by otomato," 2016. [Online]. Available: https://devopsflowmetrics.org/
[234] C. C. Ann Marie Fred, "6 proven metrics for DevOps success | TechBeacon," 2020. [Online]. Available: https://techbeacon.com/devops/6-proven-metrics-devops-success
[235] I. Software, "6 Metrics to Measure DevOps Test Automation," 2017. [Online]. Available: https://www.indiumsoftware.com/blog/devops-test-automation-metrics/
[236] T. Palko, "The Missing Metrics of DevOps," 2015. [Online]. Available: https://insights.sei.cmu.edu/devops/2015/05/the-missing-metrics-of-devops.html
[237] Cprime, "DevOps Metrics to Monitor Software Development - Cprime," pp. 1–15, 2021. [Online]. Available: https://www.cprime.com/resources/blog/devops-metrics-to-monitor-software-development/
[238] D. N. Blank-edelman, *Seeking SRE: Conversations About Running Production Systems at Scale*. O'Reilly Media, Inc., 2018.
[239] B. Beyer, C. Jones, J. Petoff, and N. R. Murphy, *Site Reliability Engineering: How Google Runs Production Systems*. O'Reilly Media, Inc., 2016. [Online]. Available: https://landing.google.com/sre/sre-book/toc/
[240] C. J. Kuiper, "Relationship of Transformational Leadership and Organizational Change During Enterprise Agile and DevOps Initiatives In Financial Service Firms," Ph.D. dissertation, Liberty University, School of Business, 2019.
[241] A. Kankanhalli, B. C. Tan, and K. K. Wei, "Contributing knowledge to electronic knowledge repositories: An empirical investigation," *MIS Quarterly: Management Information Systems*, vol. 29, no. 1, pp. 113–143, 2005.
[242] K. Behr, G. Kim, and G. Spafford, *The Visible Ops Handbook: Implementing ITIL in Four Practical and Auditable Steps*. IT Process Institute (ITPI), 2005.
[243] J. Venable, J. Pries-Heje, and R. Baskerville, "FEDS: a Framework for Evaluation in Design Science Research," *European Journal of Information Systems*, vol. 25, no. 1, pp. 77–89, jan 2016.
[244] ——, "A Comprehensive Framework for Evaluation in Design Science Research," in *Proceedings of the 7th International Conference on Design Science Research in Information Systems*, 2012, pp. 423–438.
[245] J. Pries-Heje, R. Baskerville, and J. Venable, "Strategies for design science research evaluation," *16th European Conference on Information Systems, ECIS 2008*, 2008.
[246] J. Pries-Heje and R. Baskerville, "The design theory nexus," *MIS Quarterly: Management Information Systems*, vol. 32, no. 4, pp. 731–755, 2008.
[247] B. Saunders, J. Sim, T. Kingstone, S. Baker, J. Waterfield, B. Bartlam, H. Burroughs, and C. Jinks, "Saturation in qualitative research: exploring its conceptualization and operationalization," *Quality and Quantity*, vol. 52, no. 4, pp. 1893–1907, jul 2018.