Indonesian Journal of Computer Science
ISSN 2549-7286 (*online*)
Jln. Khatib Sulaiman Dalam No. 1, Padang, Indonesia
Website: ijcs.stmikindonesia.ac.id | E-mail: ijcs@stmikindonesia.ac.id

An Evaluation of DevOps Practices to Prevent Fraud: A Case Study of PT XYZ

Adi Hartanto[1], Rizal Fathoni Aji [2]
adi.hartanto11@ui.ac.id, rizal@cs.ui.ac.id
[1,2]Faculty of Computer Science, Universitas Indonesia

Article Information

Submitted : 14 Dec 2023
Reviewed: 18 Dec 2023
Accepted : 30 Dec 2023

Keywords

DevOps, maturity, practices, evaluation, improvement

Abstract

Fraudulent transactions caused mainly by high level bugs in PT XYZ's production environment have caused significant financial losses for the company. Such issues were attributed to subpar DevOps practices in PT XYZ. Based on the given problem, this research evaluates DevOps practices in PT XYZ based on Bucena's maturity model and proposes recommendations for improvement. This study collects data using interviews with four key figures in PT XYZ's software development team and processed using the thematic analysis method in conjunction with magnitude coding. The results showed that PT XYZ's primary weaknesses in implementing DevOps practices fall mostly within Technology and Process domains. From these findings, a roadmap is designed for a timeframe of 12 months, which consists of 9 primary objectives to fulfill to increase overall DevOps maturity level at PT XYZ up to the Defined level.

A. Introduction

Online payments have become a ubiquitous, commonly used means of transaction internationally [1], [2]. As they become more widespread in use, issues that happen in the process of doing online transactions also become a matter of greater importance. Such issues can include fraud, defined as transactions that are not expected to occur by the payment provider and involve movement of funds that result in financial losses for the payment provider [3]. Fraud in this context is part of a greater breadth of cybersecurity issues that all digital service providers face today [4], and it is estimated that international fraudulent online transactions have led to a cumulative loss of USD 35 billion for digital service providers [3]. Fraud may take multiple forms, such as data falsification [5], voice phishing [6], and bug exploits that lead to illegal transactions [7]. In particular, fraudulent transactions that occur from exploiting bugs often stem from malicious internal actors with advance knowledge of the bug [7].

The resilience of an online service against cybersecurity threats correlates to its development cycle time and operational scale, especially if the service's development time is highly accelerated as with most implementations of DevOps practices [8]. DevOps, in this context, is defined as a methodology for software development that focuses on continuous integration and continuous deployment to reduce development time and provide a faster product release [9]. However, in prioritizing development speed, DevOps practitioners commonly undermine good security practices and implementations, among which includes poorly written code that results in security exploits [7].

Given the above, the premise of this paper is a case study into the implementation of DevOps, and evaluation thereof, in an enterprise environment, using PT XYZ as a model organization. PT XYZ is an Indonesian IT company specializing in distribution of digital products and aggregation of payment-point-online-banking (PPOB) services. PT XYZ primarily conducts business-to-business (B2B) transactions, with a clientele consisting primarily of other companies that operate sales of digital products. PT XYZ's internal operational evaluation system includes a set of key performance indicators (KPI), which includes a parameter for fraud frequency and its financial impact the company. PT XYZ's fraud frequency KPI evaluation for the first semester of 2023 outputted a result of 1.5 out of 5, defined as "documented case(s) of fraudulent transactions with a cumulative financial loss of more than IDR 250 million". These fraudulent transactions affect not only PT XYZ itself, but also PT XYZ's partner companies. Beyond the financial repercussions, fraudulent transactions would also lower PT XYZ's trustworthiness with its partner companies. Therefore, it is important that PT XYZ takes steps to find a long-term solution to this problem.

Table 1. Production bugs report for Q1-Q2 2023 from PT XYZ

System ID	Number of reported bugs		
	Low level	Medium level	High level
System A	-	2	-
System B	-	-	1
System C	3	-	-
System D	3	4	1
System E	6	5	-
System F	-	-	1
System G	2	-	-
System H	5	6	-
System I	-	-	1

Based on information gathered from interviews with the company's senior management, the main root cause of fraudulent transactions happening within PT XYZ's systems is system vulnerabilities and exploits caused by high-level bugs in the production environment (see Fig. 1). The issue of a company suffering losses due to faults in the development and operation of a software service has had precedent in prior literature [10]. [10] describes a case study intended to devise an improvement plan that address identified flaws in both the development and operational aspects of the case study's object of research. Both aspects fall within the scope of DevOps theory and practices [7], [8]. Further research by [11] states that properly implemented DevOps practices can serve as a way for a company to rectify issues with their software service products, in terms of both operations and developmental maintenance.

Past studies have covered more generalized topics in a corporate software development environment [10]. As such, DevOps is selected as the main perspective through which this case study will be conducted, with the novelty of a qualitative model-based approach contrasting with quantitative [12] or experimental [13] approaches to studying DevOps. There are two research questions posed by this case study:

1. What is the current state of DevOps implementation in PT XYZ?
2. Based on its current state of DevOps implementation, how can PT XYZ improve its current state of DevOps implementation to mitigate future issues?

The output of this case study is a viable roadmap for the implementation of DevOps in PT XYZ.

B. Literature Review

This section discusses the theoretical basis for this case study, which consists of DevOps and its practices, DevOps maturity models, as well as prior academic work.

1. DevOps overview

[14] defines DevOps as an approach to software engineering in which developers and operational staff work together, communicating closely and

collaboratively working with each other to benefit from an accelerated development cycle and rapid releases. [15] states that even though DevOps as a concept has been variously defined by academicians and IT practitioners alike, a common thread among these definitions is an interlinked presence between developers and operational staff. The concept of DevOps was created as a proposed evolution of the agile software engineering methodology, which itself initially aims to enable greater adaptability of developers in the face of ever-changing needs of the consumer [16], [17].

The implementation of DevOps can serve as a method of improving software engineering processes, as described in [10]. [18], expresses a similar sentiment, finding a link between DevOps and software development output quality. In addition, DevOps implementation and its effect on the quality of the software production process have been studied from several perspectives by previous research. For example, [19] focuses on the quality of code testing, [10] focuses on the process of product development, and [20] focuses on the business performance of companies whose product development processes integrate DevOps practices.

2. DevOps concepts and practices

The implementation of DevOps consists of four core concepts: culture, automation, measurement, and sharing. These four principles can be abbreviated as CAMS. [20], [21]. The first concept, culture, refers to the establishment of collaboration and cooperation between staff members and teams, intended to replace the "closed box" nature of a company's standard work habits with those that favor open communication [16], [22]. As culture change in a company cannot be enforced overnight, the rate at which a company's work culture transitions to this practice generally depends on its size and number of employees [14]. The second concept, automation, describes a general push towards automation of software development processes [14], [16]. At its core, automation in DevOps is driven by continuous integration/continuous delivery (CI/CD), defined as a sustained and automated approach to changing, merging, and deploying code supported by certain tools [14]. The purpose of applying this principle is to minimize the manual handover of builds and releases from the development team to the operational team, which can otherwise slow down the software product development cycle. The third aspect, measurement, refers to the establishment of continual and simultaneous monitoring of development teams and operational teams under a shared, quantifiable key metric [16]. [10] in their research discusses various parameters that can serve as benchmarks for software process improvement (SPI), and notes that implementing DevOps may serve as a viable method of fulfilling said benchmarks. The fourth and final concept, sharing, describes the activity of disseminating information and knowledge between teams. The application of concept is meant to prevent siloing, defined as the restriction of important information so that only one team in the entire organization knows it [23].

There are several practices that underlie the implementation of DevOps on the development side. Previous research ([22], [24]–[28]) generally mention the following as examples of DevOps practices:

1. Agile
2. Continuous Integration and Continuous Delivery (CI/CD)

3. Microservices
4. Infrastructure as Code (IAC)

Agile is a software development methodology that emphasize rapid and iterative product releases in order to get feedback from the product's consumers as soon as possible [25]. [24] proposes four main values that serve to direct Agile practices, namely prioritizing individuals, and interpersonal interactions instead of tools and abstract processes, focusing on releasing adequate code builds rather than aiming for complete documentation, doing more frequent collaboration with consumers rather than formal contract negotiations, and putting more weight in rapid response to change as opposed to strictly following a workplan.

Continuous Integration and Continuous Delivery (CI/CD) is a set of work processes and applications/tools to automate code changes and releases to a production environment [24], [26]. CI/CD implementation as part of DevOps implementation requires several tools including but not limited to [27]: version control software supported by server/cloud version control service providers (e.g. GitHub or Bitbucket), source code builders (e.g. Ant and Gradle), and software testing tools (e.g. JUnit and Selenium).

Microservices are applications that are placed in their own virtual environment (container) with predefined computational resource allocation and execution parameters [22]. [25] states that microservices can be developed independently and used alone or developed together as part of a group of other applications.

Infrastructure as Code (IAC) is a design concept in which software infrastructure management is controlled through direct programming [25]. IAC is based on a foundation of application infrastructure that can be configured through application programming interfaces (APIs) [13].

3. Related work

Based on previous academic literature, five papers were selected to be analyzed based on their similarities to this research, as follows:

Table 2. Previous literature selection

Code	Author	Title
R1	Gunawan & K. Budiardjo [10]	"A Quest of Software Process Improvements in DevOps and Kanban: A Case Study in Small Software Company"
R2	Mäkinen et al. [12]	"Revisiting Continuous Deployment Maturity: A Two-Year Perspective"
R3	Marnewick & Langerman [29]	"DevOps and Organizational Performance: The Fallacy of Chasing Maturity"
R4	Souza et al. [13]	"Infrastructure as Code as a Foundational Technique for Increasing the DevOps Maturity Level: Two Case Studies"
R5	Rafi et al. [30]	"Readiness model for DevOps implementation in software organizations"

R1 [10] is a case study of an Indonesian software development company with a stated research objective of determining software development process maturity

level in the company and designing long-term improvement suggestions. This case study takes a qualitative approach through gap analysis and root cause analysis to determine the state of software development processes, using ISO 29110 as a guideline for evaluation. The results of this case study show that there are 18 aspects in which the company must improve upon to satisfy ISO 29110 standards.

R2 [12] is a case study of Solita Ltd., a Finnish software development company with a stated research objective of determining the specific maturity level of the company's continuous development processes. This case study takes a quantitative approach based on Solita Test, a bespoke maturity model, and uses a questionnaire to collect data over the course of 2 years. The results of this study show that Solita Ltd.'s continuous development maturity level with regards to build and deployment improved over the study's time span, yet test automation and quality remain the same.

R3 [29] is a case study of a South African national bank with a stated research objective of determining the impact of implementing DevOps based on a specific DevOps maturity model towards the company's organizational performance. This case study takes a qualitative approach, analyzing the company's performance based on reports by consulting firms as well as internal company data. The results of this study show that on its own, the use of a DevOps maturity model does not increase organizational performance, and that it must be used in conjunction with various other interventions and strategies.

R4 [13] is a case study of two companies in the United States; one is a telecommunications company, and the other a medical service provider. This research has a stated research objective of analyzing the effects of implementing IAC on the maturity level of DevOps in each company. This case study is an experiment in which for each company studied, an application was built using IAC design principles. After the applications were released, each company's DevOps maturity model was examined using Bucena's maturity model as the guideline. The results of this study show that IAC-based applications confer a positive effect on DevOps maturity levels, specifically with regards to the Technology domain.

R5 [30] is a case study of three different companies whose identities were anonymized, with a stated research objective of determining each company's readiness to implement DevOps using a bespoke model, RMDevOps. This case study qualitatively assesses all three companies through interviews and conference calls. The results of this study show that there are 18 critical challenges and 73 best practices that correlate positively with a company's readiness to implement DevOps.

Based on the above, the authors decided to use Bucena's Maturity Model as a foundation for evaluating DevOps maturity at PT XYZ, with [13] as the reference study. Bucena's Maturity Model is a DevOps maturity model developed by Ineta Bucena and Marite Kirikova in 2017 [31]. This model is different from most DMMs that tend to be structured as one uniform diagram, in that it is instead composed of four models that each cover a distinct domain of DevOps implementation, namely Technology, Process, People, and Culture. Each variable has five levels, respectively from lowest to highest as follows: Initial, Repeatable, Defined, Managed, and Optimized. This model was chosen because its broad scope makes it viable for a holistic evaluation, and its modular nature means that organizations using it can choose which aspects to prioritize independently of each other [32]. In addition, the

chosen reference study [13] had a notably identical research objective in specifically evaluating the research object's DevOps maturity level. Although [13] took a different approach for its research method, which consists of an experiment with IAC implementation, the outline of its research design can still be applied to the context of PT XYZ.

C. Research Method

This research aims to determine the maturity level of DevOps implementation at PT XYZ and provide recommendations for improvement based on the evaluation results. This research is classified as a case study, defined as research that focuses on a particular object or organization related to events or situations that are of academic interest [33]. Because the context of this research (i.e. DevOps maturity level evaluation) is novel to the object of research, it can be considered as exploratory research. Exploratory research is commonly used in certain situations that are still unclear and need to be explored more deeply to understand what is really happening [33]. Qualitative methods in the form of interviews are used to collect data for this research, selected because they generally explore information or information more flexibly and in an in-depth manner [33].

This case study selects four key figures in PT XYZ's hierarchy as interview sources, namely a Project Manager, the Engineering Manager, the Vice President of IT, and the Chief Technology Officer. These people were chosen because only they have the greatest collective oversight of the overall software development processes and activities happening at PT. XYZ. The data collected will be processed using the thematic analysis [34] method in conjunction with the magnitude coding method [10]. These methods are suitable for analysing qualitative data by identifying certain recurring patterns, which are then transformed into quantitative data by categorizing results based on predetermined values representing a certain scale [34]. In this case, the categories used are the DevOps maturity levels depicted in Bucena's maturity model with an ordinal scale of 1-5. In this context, 1 represents the lowest level (Initial) and 5 represents the highest level (Optimized). The results of thematic analysis are mapped to the four domains of Bucena's maturity model, and based on its results, a roadmap of improvements is designed to suit the state of DevOps implementation in PT XYZ.

A notable challenge in data processing is that the answers from multiple sources can be interpreted differently. Notably, [31] in their research did not explain in more detail how to process data converted to quantitative form obtained from the assessment results of several respondents or sources. Therefore, the authors in this study will use the method of averaging points for each answer and then round to the nearest whole number to determine the overall maturity level of each domain, with the following details:

1. An averaged value of 0 to 1.49 is mapped to maturity level 1 (initial).
2. An averaged value of 1.5 to 2.49 is mapped to maturity level 2 (repeatable).
3. An averaged value of 2.5 to 3.49 is mapped to maturity level 3 (defined).
4. An averaged value of 3.5 to 4.49 is mapped to maturity level 4 (managed).
5. An averaged value of 4.5 to 5.0 is mapped to maturity level 5 (optimized).

D. Result and Discussion

This section discusses the results of the interview with sources of PT XYZ, analyzes each variable that requires improvement, and proposes a roadmap of improvement activities based on the result of the analyses.

1. Analysis of DevOps Assessment Interview Results

This subsection discusses the results of interviews with sources from PT XYZ, namely Project Manager (PM), Engineer Manager (EM), Vice President of IT (VP IT), and Chief Technology Officer (CTO). A preliminary assessment with the CTO states an expected baseline level of Defined for each general domain. It should be noted that during interviews, PM only answered questions related to the Process, People, and Culture domains. Table 3 shows the summary of the results for each variable following the criteria outlined in section 3.

Table 3. Summary of DevOps maturity level assessment

ID	Variable	Assessment Result					
		PM	EM	VP IT	CTO	Average	Level
T1	Deployment Environment	-	3	3	3	3	Defined
T2	Testing Scope	-	2	2	2	2	Repeatable
T3	Data Management	-	1	1	1	1	Initial
T4	Deployment Process	-	1	1	1	1	Initial
T5	Code Building	-	1	1	1	1	Initial
T6	Team Collaboration	-	2	2	2	2	Repeatable
T7	Software Configuration	-	1	1	1	1	Initial
T8	System Monitoring	-	1	1	1	1	Initial
T9	Bug Tracking	-	1	1	1	1	Initial
PR1	Delivery Scheduling	1	2	1	2	1.5	Repeatable
PR2	Development Approach	3	3	3	3	3	Defined
PR3	Testing Setup	2	3	3	2	2.5	Defined
PR4	Project Management	2	3	3	2	2.5	Defined
PR5	System Documentation	1	1	3	2	1.75	Repeatable
PR6	Standardization	3	2	2	2	2.25	Repeatable
P1	Team Organizing	3	3	3	3	3	Defined
P2	Team Learning	2	2	4	2	2.5	Defined
P3	Skill Development	2	2	3	3	2.5	Defined
C1	Team Communication	3	4	5	4	4	Managed
C2	Requirements Understanding	3	4	4	4	3.75	Managed
C3	Company Culture	2	4	4	4	3.5	Managed
C4	Company Collaboration	3	4	4	3	3.5	Managed
C5	Company Innovations	2	2	3	3	2.5	Defined

a. Deployment Environment (T1) is the first variable in the Technology domain. It concerns the provisioning of deployment environments in the organization. The three interviewees, namely EM, VP IT, and CTO, share the same view regarding the current condition of PT XYZ's deployment environment management. Currently, PT XYZ provides virtual deployment servers in the form of virtual

machines (VMs). These include VMs for development and testing, VMs for back-end, VMs for front-end, and other purposes. Each running service has its own virtual environment ("venv") to adjust with the version of the programming language or modules used. However, VM management, management of permissions, and venv setup processes are still being done manually and have not yet used integrated configuration management.

b. Testing Scope (T2) is the second variable in the Technology domain. It concerns how the organization integrates automation and assistive tools into software testing. Based on the interview results, PT XYZ uses several tools for automated testing, such as Katalon for testing web applications, Repeato for testing mobile applications, and Postman for testing application programming interfaces (APIs). Testers generally design a testing scenario in each of these tools first, and then run the test scenario as a whole. Even though PT XYZ's developers have made use of testing tools, there is still a significant type of automated testing that has not been carried out, namely unit testing.
c. Data Migration (T3) is the third variable in the Technology domain. It concerns the process of data migration during development. Currently, in PT XYZ's deployment process, each application's database is prepared by their respective developer manually. Scripts for database updates have not been managed with version control.
d. Deployment Process (T4) is the fourth variable in the Technology domain. It concerns the organization's workflow for application deployment. Based on interview results, deployment at PT XYZ is performed manually through pull requests on the VM and directory of each application. Although PT XYZ already employs Bitbucket for code versioning and peer review, it has not yet been utilized optimally to implement a CI/CD pipeline.
e. Code Building (T5) is the fifth variable in the Technology domain. It concerns how source code is built during the development process. At PT XYZ, code building is still done manually by developers on local computers. The build results are then uploaded to the VM for deployment.
f. Team Collaboration (T6) is the sixth variable in the Technology domain. It concerns the use of tools to support collaboration. At PT XYZ, teams use Jira and Confluence to assist collaboration. Jira is used to plan projects, assign tasks, and write kanban/sprint boards. Meanwhile, Confluence is used to share API specifications between teams. Currently, collaboration at PT XYZ does not use an integrated toolset due to lack of continuous integration (CI).
g. Software Configuration (T7) is the seventh variable in the Technology domain. It concerns how software configuration management is performed along with application deployment. Based on interview results, at PT XYZ the development team creates configuration files to manage the environment and database connections for each deployed application. However, configuration files made this way are not standardized, and their structure depends on each developer. To try to standardize this process, PT XYZ is currently experimenting with Docker as a software configuration management (SCM) tool.
h. System Monitoring (T8) is the eighth variable in the Technology domain. It concerns how the organization monitors system performance and data flows. Currently, PT XYZ operates two tools for monitoring, namely Grafana for log

monitoring and Proxmox for monitoring networks and VMs. According to the EM, VP IT, and CTO, monitoring on the whole is still done very minimally. Monitoring tools tend to only be checked when there is a system error report. Furthermore, some of these tools have not been fully utilized, especially with regards to features related to warnings and reporting. In addition, PT XYZ still does not have any tools for monitoring applications.

i. Bug Tracking (T9) is the ninth variable in the Technology domain. It concerns how the organization handles reports of bugs and other errors. Based on interview results, PT XYZ currently operates a tool called Grafana for issue tracking. Grafana is a big data-based logging system that collects logs from various services in real time. Currently, there are 16 services and applications in PT XYZ that are monitored through Grafana. However, the staff tends to only check logs from Grafana whenever an error is reported. There are features in Grafana that PT XYZ's staff have not utilized, such as error alerts and customizable dashboards.
j. Delivery Scheduling (PR1) is the first variable in the Process domain. It concerns how developers schedule product deliveries. According to the EM and CTO, the delivery process that runs at PT XYZ already has an established schedule and workflow. The general workflow is for a developer to sign a report, begin deployment, conduct operation testing, then release to market. However, according to PM and VP IT, this workflow is not always consistently executed across projects.
k. Development Approach (PR2) is the second variable in the Process domain. It concerns how software development is generally conducted at the organization. Based on interview results, PT XYZ's approach to development generally uses agile methodologies. This was the consensus among the four interviewees. However, the PM and CTO are of the opinion that the agile implementation at PT XYZ cannot yet also be called lean.
l. Testing Setup (PR3) is the third variable in the Process domain. It concerns how developers at the organization prepare and perform software testing. Presently, developers at PT XYZ typically conduct requirement-based testing. The PM describes integration testing at PT XYZ as something that is only sparsely done. On the other hand, the EM and VP IT opine that integrated tests are generally carried out, but only minimally or partially documented.
m. Project Management (PR4) is the fourth variable in the Process domain. It concerns how management oversees projects in the organization. PT XYZ presently manages business requirements along with general project management. The EM and VP perceives project management at PT XYZ as an activity that is integrated with project management toolset. However, the CTO and PM disagree with this perception, as they believed the tools currently employed by PT XYZ have not been optimally used.
n. System Documentation (PR5) is the fifth variable in the Process domain. It concerns the breadth and depth of documentation for systems developed in the organization. The EM and PM opine that at PT XYZ, system documentation is generally minimal. This contrasts with the CTO's perception, who mentioned that documentation tends to be up-to-date especially with regards to configuration files. This viewpoint is supported by the VP IT, who stated that PT

XYZ's developers generally make use of Confluence for API documentation and Google Drive for deployment documentation. However, system architecture documentation was perceived by the VP IT to be generally insufficient.

o. Standardization (PR6) is the sixth variable in the Process domain. It concerns institutionalization of company work practices and habits. According to all four interviewees, PT XYZ has generally implemented repeatable, standardized system development processes. With regards to the deployment process, the VP IT thinks that its processes are not yet standardized. However, according to the PM, there are already standardized processes for deployment, though they are not yet formally defined through standard operating procedures (SOPs). A similar opinion was also expressed by EM.
p. Team Organizing (P1) is the first variable in the People domain. It concerns how the organization assigns people into teams. Presently, teams within PT XYZ are usually formed based on how each person may contribute to each project to be carried out, as stated by the four interviewees.
q. Team Learning (P2) is the second variable in the People domain. It concerns the learning processes conducted by team members in the organization. Based on interview results, employee learning at PT XYZ is carried out through weekly meetings to discuss each team's project continuity and peform group problem-solving. In this scenario, the VP of IT argues that cross-process learning has also been implemented by assigning individuals with minimal to rudimentary knowledge of a certain technology into development projects that use said technologies.
r. Skill Development (P3) is the third variable in the People domain. It concerns how the organization enables staff growth and competency upgrading. Based on interview results, PT XYZ provides an e-learning system accessible to employees, as well as special training to employees who are due for promotion. The VP IT and CTO stated that in their experience, teaching/mentoring is also carried out to each employee in conjunction with e-learning system.
s. Team Communication (C1) is the first variable in the Culture domain. It concerns how teams communicate with each other and with their superiors. Currently, communication within PT XYZ is generally carried out frequently between teams, as described by the EM and CTO. The VP of IT further states that communication between teams at PT XYZ accomodates rapid feedback, as there are no bureaucratic practices that can hinder communication, but this view is not supported by the CTO who feels there are still issues in delivering feedback in communication.
t. Requirements Understanding (C2) is the second variable in the Culture domain. It concerns how organization staff perceive business requirements. Based on the interview results, at PT XYZ business requirements are generally clearly defined and can be understood by project teams. All four interviewees also stated that the staff understand PT XYZ's product line requirements clearly, as reflected in their work objectives. A shortcoming related to this aspect was conveyed by the VP of IT, who mentioned that the overall organizational requirements are still poorly understood amongst the staff.
u. Company Culture (C3) is the third variable in the Culture domain. It concerns the perception of corporate culture in the organization and how it impacts

employees. The EM, VP IT, and CTO stated that corporate culture at PT XYZ is seen as a shared asset that needs to be maintained. The culture in question, as explained by EM, is one of kinship, mutual assistance, and constant communication. However, PM opines that such a culture is not believed to help PT XYZ's business strategically. PM also mentioned that many employees are not clearly aware of PT XYZ's long-term business strategies, in addition to its corporate culture.

v. Company Collaboration (C4) is the fourth variable in the Culture domain. It concerns how teams and individuals collaborate within the organization. Based on the interview results, all four interviewees agree that collaboration at PT XYZ is actively practiced, barriers to collaboration are generally known and avoided, and issue escalation tends to be properly directed to the correct stakeholder. EM states that these practices are supported by the establishment of a clear person-in-charge (PIC) for each team.
w. Company Innovations (C5) is the fifth variable in the Culture domain. It concerns how novel approaches are studied and implemented for the organization's benefit. Generally, Innovation at PT XYZ is seen as something that is done only when needed, as stated by PM and EM. On the other hand, VP IT and CTO stated that innovation is done actively with concepts and technologies that can prove useful to the company's business goals.

From the assessment results, taking into account the expectations of the CTO of PT XYZ, namely the Defined level for each domain of DevOps implementation, the authors conclude that the implementation of DevOps at PT XYZ underperforms at the Technology and Process domains. The Technology domain stands at the Initial level on average, with 8 variables that do not meet the expected standards, as follows:

1. Testing Scope (T2)
2. Data Migration (T3)
3. Deployment Process (T4)
4. Code Building (T5)
5. Team Collaboration (T6)
6. Software Configuration (T7)
7. System Monitoring (T8)
8. Bug Tracking (T9)

Meanwhile, the Process domain stands at the Repeatable level on average, with 3 variables that do not meet the expected standards, namely:

1. Delivery Scheduling (PR1)
2. System Documentation (PR5)
3. Standardization (PR6)

2. Analysis of DevOps Implementation Improvement for Technology Domain

In the Technology domain, the first variable with an insufficient maturity level is Testing Scope (T2), which is at the Repeatable level. At PT XYZ, several tools have been used for automated testing such as Postman, Katalon, and Repeato. However, automated unit testing has not yet been implemented. Regarding this, PT XYZ mainly

uses two main programming languages: Angular for front-end programming and Python for back-end programming. Both languages already provide modules for unit testing, namely karma (Angular) and pytest (Python). To improve this variable to the Defined level, triggered automated tests must be implemented [31], in which unit tests will be automatically run by the pipeline every time the developers call a git push to Bitbucket. As such, it is necessary to utilize pipeline and unit testing features both at the front-end and back-end of each system.

The second variable with an insufficient maturity level is Data Migration (T3). To increase the level of this variable to Defined level, the development team needs to prepare automated database modification scripts for each application version [31]. This takes advantage of several tools used for continuous integration (CI), build automation, and version and source control [18]. An example of a tool that can be used to implement continuous integration and build automation is Jenkins. Other tools to support data migration currently used by PT XYZ are Jira for integrated deployment planning, and Bitbucket for version and source. With proper implementation, changes to the database structure will be automatically executed along with the CI process. Accordingly, Jenkins will run the database structure changes based on the script and configuration that has been prepared.

The third variable with an insufficient maturity level is Deployment Process (T4). Improvements to this variable can be made by implementing DevOps practices in the form of build automation, continuous integration, and continuous deployment [31]. Tools that can be used to support these practices are Bitbucket pipeline and Jenkins. A properly implemented Bitbucket pipeline will ensure that the source code in branch testing passes unit testing, while Jenkins will automatically build and deploy code to the correct environment. This deployment workflow can be seen in Figure 1, which illustrates the process.

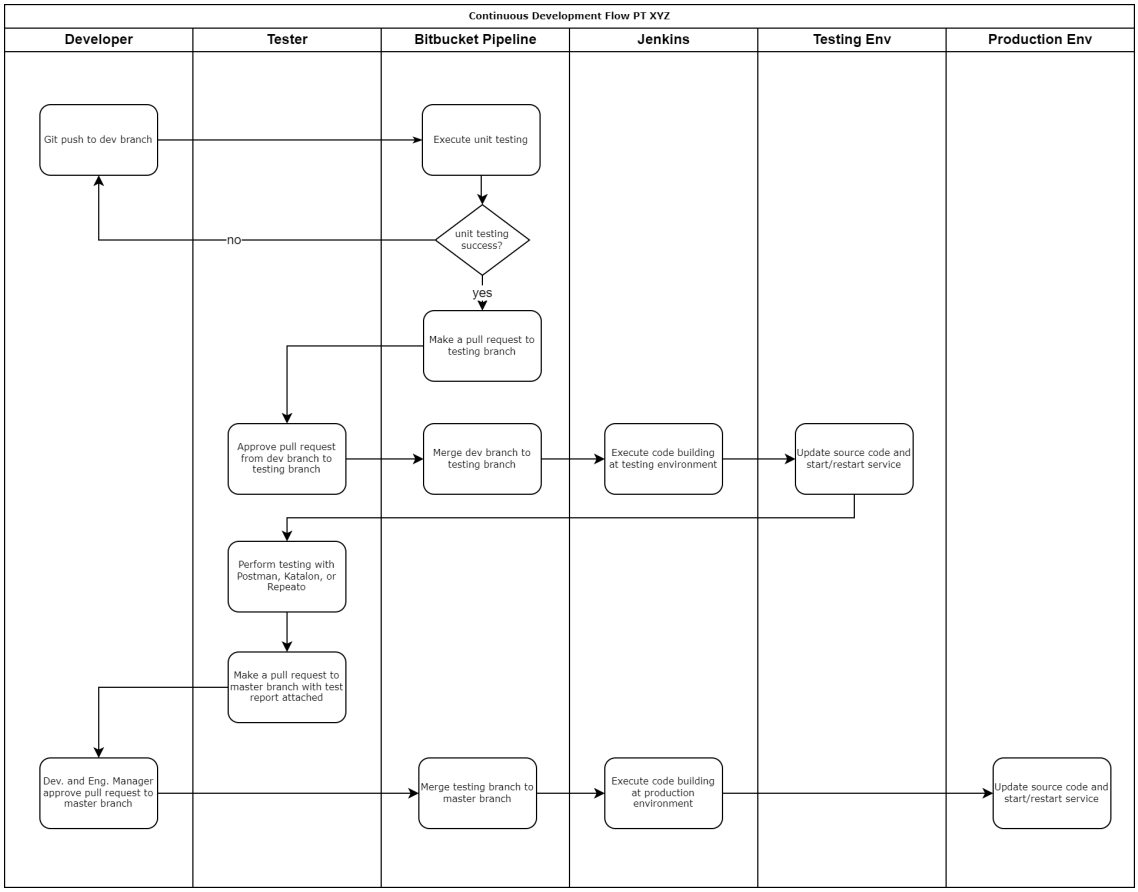

Figure1. Deployment Process improvement flowchart

This process begins when a developer pushes code to the dev branch. After a successful push, unit tests on the code are run by the Bitbucket pipeline. If the unit test fails, the developer must git push the revised code again. If instead the test is successful, the Bitbucket pipeline will automatically create a pull request to the testing branch. In this step, a tester approves the pull request from dev to testing. The dev branch is then merged with the testing branch, and code is built in the testing environment using Jenkins. After the code building is complete, the source code in the testing environment is updated and the corresponding service in the same environment is started or restarted. The tester then performs testing on the code with the help of tools such as Postman, Katalon, or Repeato. When the testing is complete, the tester makes a pull request to the branch master by including a test report. The developer and Engineer Manager (EM) then approve the pull request that has been submitted by the tester, and the testing branch is merged into the master branch. Code building is then done again in the production environment. When code building is complete, the source code in the production environment is updated and the corresponding service in the same environment is started or restarted.

The fourth variable in the Technology domain with an insufficient maturity level is Code Building (T5). Improving this variable to the Defined level primarily requires implementing build automation. [31]. Improvements to the deployment process (i.e. variable T4) will naturally also include improvements to code building by way of using Jenkins as a build automation tool.

The fifth variable in the Technology domain with an insufficient maturity level is Team Collaboration (T6). At the moment, PT XYZ uses Jira and Confluence as collaboration tools. However, the use of these tools alone is not enough to place this variable at the Defined level. Implementing CI/CD will increase this variable to the Defined level, as it encourages stronger collaboration between developers, testers, and operational teams and thus necessitate a highly integrated usage of team collaboration toolsets [31].

The sixth variable in the Technology domain with an insufficient maturity level is Software Configuration (T7). In order to upgrade this variable to the Defined level, the implementation of software configuration management tools is required in a manner that facilitates automated software configuration file delivery with each instance of code handover [31]. Docker can be utilized for this purpose, as Docker instances have a standardized software configuration file.

The seventh variable in the Technology domain with an insufficient maturity level is System Monitoring (T8). To improve this variable to the Defined level, there must exist an integrated tool for monitoring multiple services and systems [31]. Zabbix is a viable tool to implement for this purpose, with the primary consideration being that it is a mature open-source tool that supports PT XYZ's needs.

The last variable in the Technology domain whose maturity level has not met the expected standard is Bug Tracking (T9). As it stands, PT XYZ currently stores all logs from various services to Grafana. These logs are used to track the incidence of bugs when an error occurs. To improve this variable to the Defined level, PT XYZ needs to automate error alerts and bug report status (e.g. open, work in progress, resolved) [31]. These can be achieved by making extensive use of Grafana's error alert and custom issue dashboard features in conjunction with Jira.

3. Analysis of DevOps Implementation Improvement for the Process Domain

In the Process domain, the first shortcoming of PT XYZ's DevOps implementation lies in the Delivery Scheduling (PR1) variable. Delivery Scheduling is strongly related to the practice of continuous delivery (CD) [31]. CD encompasses the automation and simplification of the software product output process, especially in the deployment stage [18]. CD implementation is required to achieve the Defined level for this variable, with the prerequisites being knowledge and use of several types of tools to support continuous integration (CI), build automation, version and source control, and automated testing and validation [18], [31]. PT XYZ has implemented the use of tools for version and source control in the form of Bitbucket, as well as Postman, Katalon, and Repeato for automated testing and validation. These tools can be supplemented with Jenkins for the purpose of build automation.

The second shortcoming of PT XYZ in the Process domain lies in the System Documentation (PR5) variable. System documentation can be split into technical and user documentation [35]. Technical documentation for a system typically includes architecture design, interface structure, and developer-facing system functionality. Meanwhile, user documentation includes the workings of each system feature provided to users [35]. For this variable, [31] defines the Defined level as a state where the latest documentation is always validated regularly, and a description of configuration files relevant to the system is also provided. In order to

achieve this level, there are three common methods suitable for the agile methodology: source code-based documentation with tools such as Javadocs and Docstring, wiki-based documentation with tools such as XSDoc and sprintDoc, and recording user stories as a minimum guide to a feature [36].

The last shortcoming of PT XYZ in the Process domain lies in the Standardization variable. The Defined level for this variable is achieved when the processes commonly carried out in software development in an organization have been codified into formal guidelines that must be followed by all software development project teams in the company [31], for example in the form of a SOP. As there are already unwritten standards for development in PT XYZ, it follows that these standards should be formalized as a SOP and socialized to all employees. This SOP ideally should also includes IT security best practices [37]. The creation and socialization of development SOPs that incorporate IT security practices is academically supported by research such as [38], which states that the convenience of examining written procedures and organizational policies related to IT security positively contributes to the quality of IT security practices.

4. Roadmap for Improvement of DevOps Implementation

There are nine primary activities that PT XYZ needs to fulfill to improve their implementation of DevOps, broadly encompassing the variables analyzed in subsection 4. These activities are organized into a roadmap with an estimated time span allocation of 12 months (see Table 4).

Table 4. Timetable for proposed improvements

No.	Activity Name	Month											
		1	2	3	4	5	6	7	8	9	10	11	12
1	Unit testing implementation	v											
2	Bitbucket pipeline implementation	v	v										
3	Implementation of data migration scripts	v	v										
4	Jenkins usage experiments		v	v									
5	Implementation of Grafana alerts and dashboards			v	v								
6	General CI/CD implementation			v	v	v	v						
7	Docker implementation						v						
8	Zabbix usage experiments							v	v				
9	Zabbix implementation									v	v	v	v

The first step is the implementation of unit testing in the first month. Unit testing is related to the implementation of the Bitbucket pipeline, as discussed in the analysis of variables T2 and T4. Therefore, the second step is to implement the pipeline together with the implementation of unit testing. Work on the Bitbucket pipeline implementation is expected to take two months. The third step is implementing data migration scripts, which should also be done in conjunction with the previous two steps. Data migration scripts can be done with the help of Jira, Bitbucket, and Jenkins. Similar to the timeline for Bitbucket pipeline implementation, it is expected that this activity will be completed in two months.

After these three steps are completed, the next stage is experimentation using Jenkins. This activity aims to familiarize PT XYZ's development staff with Jenkins during the second and third month, as well as improve aspects of the deployment process as discussed in the analysis of variable T4. There are also two other activities slated to be carried out concurrently in the third month, namely the implementation of the Grafana alert/dashboard feature and CI/CD. The implementation of Grafana's alert and dashboard features is expected to take two months, while the CI/CD implementation is expected to take four.

The second half of the roadmap includes the last three stages, in order: implementing Docker, experimenting with Zabbix, and formally implementing Zabbix. The implementation of Docker is set to take place in the sixth month, coinciding with the end of the CI/CD implementation process. Afterwards, experimentation with Zabbix is given a timeframe of two months. Finally, the implementation of Zabbix is set to be done during the last four months. In general, it is expected that the relatively longer CI/CD and Zabbix implementation time compared to other stages will also provide sufficient opportunities for the company's staff to familiarize themselves with associated processes and tools.

In addition, it is also expected that during the execution of the entire roadmap, the establishment of widespread system documentation and standardization of development processes at PT XYZ will be improved. System documentation should be validated regularly, and standardization of development processes should be applied thoroughly to every development project conducted at PT XYZ. As discussed earlier in the analysis of variable PR6, the codification of standard development practices should also include guidelines of IT security practices both in developing a system and in operating it [37], [38]. For example, the guideline may include directives on applying the principle of least privilege with regards to regulating access to sensitive information in the system, as well as linking the practice of securing personal data (e.g. using hard-to-guess passwords and two-factor authentication) with the practices of securing organizational data [39].

E. Conclusions

This study answers the research questions written in section 1, namely:

1. What is the current state of DevOps implementation in PT XYZ?
2. Based on its current state of DevOps implementation, how can PT XYZ improve its current state of DevOps implementation, in order to mitigate future issues?

The research process carried out to answer these two questions began with a literature study of previous research to find the DevOps maturity model that is most suitable as an assesment guideline, which culminates with the choice of using Bucena's Maturity Model. Subsequently, a research instrument in the form of interview questions was compiled using the parameters in each domain of Bucena's Maturity Model. After conducting interviews with four sources at PT XYZ, the answers to research question 1 are as follows:

1. PT XYZ expects a baseline maturity level of Defined (3) for all domains.
2. PT XYZ's current state of DevOps implementation for Technology and Process domains failed to meet this baseline, respectively being at the Initial (1) and Repeatable (2) levels.

3. PT XYZ's current state of DevOps implementation for the People and Culture domains successfully met this baseline, respectively being at the Defined (3) and Managed (4) levels.

To answer research question 2, each variable in the Technology and Process domains that did not meet the baseline level was analyzed, and recommendations for improvement were proposed. All improvement recommendations were organized and summarized into a 12-month roadmap, which consists of the following activities:

1. Implementation of unit testing
2. Bitbucket pipeline implementation
3. Data migration script implementation
4. Implementation of Grafana's alert and dashboard features
5. CI/CD implementation
6. Docker implementation
7. Zabbix implementation
8. Periodic validation of system documents
9. Creation of SOPs related to development, deployment, and operations

Further research that can be built upon this study primarily involves a large scale of the research subject. This includes larger organizations as a research subject, or a longer period of research in order to include data comparison before and after the implementation of proposed DevOps improvements.

F. Acknowledgment

We express our gratitude to the instructors and teaching assistants involved in the Software Engineering Team Dynamics course at the Faculty of Computer Science, Universitas Indonesia for their valuable guidance on DevOps theories and their contributions to this research.

G. References

[1] P. K. Choudhary, S. Routray, P. Upadhyay, and A. K. Pani, "Adoption of enterprise mobile systems – An alternative theoretical perspective," *Int J Inf Manage*, vol. 67, Dec. 2022, doi: 10.1016/j.ijinfomgt.2022.102539.

[2] P. Vallejo-Correa, J. Monsalve-Pulido, and M. Tabares-Betancur, "Systematic mapping review of context-aware analysis and its approach to mobile learning and ubiquitous learning processe," *Computer Science Review*, vol. 39. Elsevier Ireland Ltd, Feb. 01, 2021. doi: 10.1016/j.cosrev.2020.100335.

[3] O. Kolodiziev, A. Mints, P. Sidelov, I. Pleskun, and O. Lozynska, "Automatic Machine Learning Algorithms For Fraud Detection In Digital Payment Systems," *Eastern-European Journal of Enterprise Technologies*, vol. 5, no. 107, pp. 14–26, 2020, doi: 10.15587/1729-4061.2020.212830.

[4] I. Roussou, E. Stiakakis, and A. Sifaleras, "An empirical study on the commercial adoption of digital currencies," *Information Systems and e-Business Management*, vol. 17, no. 2–4, pp. 223–259, Dec. 2019, doi: 10.1007/s10257-019-00426-7.

[5] V. Chang, L. M. T. Doan, A. Di Stefano, Z. Sun, and G. Fortino, "Digital payment fraud detection methods in digital ages and Industry 4.0," *Computers and*

Electrical Engineering, vol. 100, May 2022, doi: 10.1016/j.compeleceng.2022.107734.

[6] C. Na, "Proactive crime prevention through problem-oriented governance: A case study of South Korea's recent efforts to tackle new types of fraud," *Policing: A Journal of Policy and Practice*, vol. 17, Jan. 2023, doi: 10.1093/police/paac080.

[7] S. Mansfield-Devine, "DevOps: finding room for security," *Network Security*, vol. 2018, no. 7, pp. 15–20, Jul. 2018, doi: 10.1016/S1353-4858(18)30070-9.

[8] D. S. Battina, "Best Practices For Ensuring Security In DevOps: A Case Study Approach," *International Journal of Innovations In Engineering Research and Technology*, vol. 4, no. 11, pp. 38–45, 2017.

[9] 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.

[10] F. Gunawan and E. K. Budiardjo, "A Quest of Software Process Improvements in DevOps and Kanban:: A Case Study in Small Software Company," in *ACM International Conference Proceeding Series*, 2021, pp. 39 – 45. doi: 10.1145/3451471.3451478.

[11] A. B. Albuquerque and V. L. Cruz, "Implementing DevOps in legacy systems," in *Advances in Intelligent Systems and Computing*, Springer Verlag, 2019, pp. 143–161. doi: 10.1007/978-3-030-00184-1_14.

[12] S. Mäkinen, T. Lehtonen, T. Kilamo, M. Puonti, T. Mikkonen, and T. Männistö, "Revisiting Continuous Deployment Maturity: A Two-Year Perspective," in *Proceedings of the 34th ACM/SIGAPP Symposium on Applied Computing*, in SAC '19. New York, NY, USA: Association for Computing Machinery, 2019, pp. 1810–1817. doi: 10.1145/3297280.3297458.

[13] I. S. E. Souza, D. P. Franco, and J. P. S. G. Silva, "Infrastructure as Code as a Foundational Technique for Increasing the DevOps Maturity Level: Two Case Studies," *IEEE Softw*, vol. 40, no. 1, pp. 63–68, Jan. 2023, doi: 10.1109/MS.2022.3213228.

[14] L. E. Lwakatare *et al.*, "DevOps in practice: A multiple case study of five companies," *Inf Softw Technol*, vol. 114, pp. 217–230, 2019, doi: https://doi.org/10.1016/j.infsof.2019.06.010.

[15] M. Alawneh and I. M. Abbadi, "Expanding DevSecOps Practices and Clarifying the Concepts within Kubernetes Ecosystem," in *2022 9th International Conference on Software Defined Systems, SDS 2022*, 2022. doi: 10.1109/SDS57574.2022.10062874.

[16] A. Trigo, J. Varajão, and L. Sousa, "DevOps adoption: Insights from a large European Telco," *Cogent Eng*, vol. 9, no. 1, 2022, doi: 10.1080/23311916.2022.2083474.

[17] A. Hemon, B. Lyonnet, F. Rowe, and B. Fitzgerald, "Conceptualizing the transition from agile to DevOps: A maturity model for a smarter is function," *IFIP Adv Inf Commun Technol*, vol. 533, pp. 209 – 223, 2019, doi: 10.1007/978-3-030-04315-5_15.

[18] A. Mishra and Z. Otaiwi, "DevOps and software quality: A systematic mapping," *Comput Sci Rev*, vol. 38, p. 100308, 2020, doi: https://doi.org/10.1016/j.cosrev.2020.100308.

[19] K. Charan, N. Karthikeya, P. B. Narasimharao, S. A. Devi, V. Abhilash, and P. V. V. S. Srinivas, "Effective Code Testing Strategies in DevOps: A Comprehensive Study of Techniques and Tools for Ensuring Code Quality and Reliability," in *2023 4th International Conference on Electronics and Sustainable Communication Systems (ICESC)*, 2023, pp. 302–309. doi: 10.1109/ICESC57686.2023.10193275.

[20] T. Mukange and J. Langerman, "The Alignment between DevOps and Business Outcomes," in *2022 International Conference on Engineering and Emerging Technologies (ICEET)*, 2022, pp. 1–6. doi: 10.1109/ICEET56468.2022.10007221.

[21] M. Muñoz and M. N. Rodríguez, "A guidance to implement or reinforce a DevOps approach in organizations: A case study," *Journal of Software: Evolution and Process*, 2021, doi: 10.1002/smr.2342.

[22] M. Alawneh and I. M. Abbadi, "Expanding DevOps Principles and Best Practices Based on Practical View," in *2022 International Arab Conference on Information Technology (ACIT)*, 2022, pp. 1–6. doi: 10.1109/ACIT57182.2022.9994216.

[23] J. Díaz, J. E. Perez, A. Yague, A. Villegas, and A. de Antona, "DevOps in Practice – A Preliminary Analysis of Two Multinational Companies," *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, vol. 11915 LNCS, pp. 323 – 330, 2019, doi: 10.1007/978-3-030-35333-9_23.

[24] S. Al-Saqqa, S. Sawalha, and H. Abdelnabi, "Agile software development: Methodologies and trends," *International Journal of Interactive Mobile Technologies*, vol. 14, no. 11, pp. 246–270, 2020, doi: 10.3991/ijim.v14i11.13269.

[25] Y. Chen, "DevOps Practices in Digital Library Development," in *2022 ACM/IEEE Joint Conference on Digital Libraries (JCDL)*, 2022, pp. 1–4.

[26] W. P. Luz, G. Pinto, and R. Bonifácio, "Adopting DevOps in the real world: A theory, a model, and a case study," *Journal of Systems and Software*, vol. 157, p. 110384, 2019, doi: https://doi.org/10.1016/j.jss.2019.07.083.

[27] S. Ferdian, T. Kandaga, A. Widjaja, H. Toba, R. Joshua, and J. Narabel, "Continuous Integration and Continuous Delivery Platform Development of Software Engineering and Software Project Management in Higher Education," *Jurnal Teknik Informatika dan Sistem Informasi*, vol. 7, no. 1, Apr. 2021, doi: 10.28932/jutisi.v7i1.3254.

[28] S. S. Sravan, C. Sai Ganesh, K. V. D. Kiran, T. Aakash Chandra, K. Aparna, and T. Vignesh, "Significant Challenges to espouse DevOps Culture in Software Organisations By AWS : A methodical Review," in *2023 9th International Conference on Advanced Computing and Communication Systems (ICACCS)*, 2023, pp. 395–401. doi: 10.1109/ICACCS57279.2023.10113021.

[29] C. Marnewick and J. Langerman, "DevOps and Organizational Performance: The Fallacy of Chasing Maturity," *IEEE Softw*, vol. 38, no. 5, pp. 48 – 55, 2021, doi: 10.1109/MS.2020.3023298.

[30] S. Rafi, W. Yu, M. A. Akbar, S. Mahmood, A. Alsanad, and A. Gumaei, "Readiness model for DevOps implementation in software organizations," *Journal of*

Software: Evolution and Process, vol. 33, no. 4, Apr. 2021, doi: 10.1002/smr.2323.

[31] I. Bucena and M. Kirikova, "Simplifying the DevOps Adoption Process," 2017.

[32] L. De Aguiar Monteiro, D. S. M. Pessoa Monteiro, W. H. Carvalho Almeida, A. Cavalcanti De Lima, and I. S. Sette, "Methods of Implementation, Maturity Models and Definition of Roles in DevOps Frameworks: A Systematic Mapping," in *Proceedings - 2020 International Conference on Computational Science and Computational Intelligence, CSCI 2020*, Institute of Electrical and Electronics Engineers Inc., Dec. 2020, pp. 1766–1773. doi: 10.1109/CSCI51800.2020.00327.

[33] U. Sekaran and R. Bougie, *Research Methods for Business: A Skill-Building Approach*, 7th ed. John Wiley & Sons Ltd, 2016.

[34] M. E. Kiger and L. Varpio, "Thematic analysis of qualitative data: AMEE Guide No. 131," *Med Teach*, vol. 42, no. 8, pp. 846–854, Aug. 2020, doi: 10.1080/0142159X.2020.1755030.

[35] E. Aghajani *et al.*, "Software Documentation Issues Unveiled," in *Proceedings - International Conference on Software Engineering*, IEEE Computer Society, May 2019, pp. 1199–1210. doi: 10.1109/ICSE.2019.00122.

[36] M. A. Islam, R. Hasan, and N. U. Eisty, "Documentation Practices in Agile Software Development: A Systematic Literature Review," in *21st IEEE/ACIS International Conference on Software Engineering, Management and Applications (SERA 2023)*, Apr. 2023. [Online]. Available: http://arxiv.org/abs/2304.07482

[37] 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, pp. 1–8. doi: 10.1109/CyberSecPODS.2019.8884935.

[38] B. Uchendu, J. R. C. Nurse, M. Bada, and S. Furnell, "Developing a cyber security culture: Current practices and future needs," *Comput Secur*, vol. 109, Oct. 2021, doi: 10.1016/j.cose.2021.102387.

[39] W. He and Z. Zhang, "Enterprise cybersecurity training and awareness programs: Recommendations for success," *Journal of Organizational Computing and Electronic Commerce*, vol. 29, no. 4, pp. 249–257, Oct. 2019, doi: 10.1080/10919392.2019.1611528.