Kristianstad University
SE-291 88 Kristianstad
Sweden
+46 44 250 30 00
www.hkr.se

Independent project (degree project), 15 credits, for the degree of Bachelor of Science (180 credits) with a major in Computer Science
Spring Semester 2021
Faculty of Natural Sciences

The core problems of globally distributed work in software development environments, and possible solutions

DevOps environments' opportunities for better adoption of a globally distributed working culture

Nemanja Negovanović
Alex Oacheşu

Authors
Nemanja Negovanović
Alex Oacheşu

Title
The core problems of globally distributed work in software development environments, and possible solutions
DevOps environments' opportunities for better adoption of a globally distributed working culture

Supervisor
Daniel Einarson

Examiner
Kamilla Klonowska

Abstract
Both distributed work and DevOps are on an upward trend. There is a slight resemblance between the problems that DevOps is trying to find answers to, the solutions, and the common problems that geographically distributed work faces. Mainly, they are related to isolated environments that have difficulties in mutual understanding and communication, collaboration. All this leads to inefficiencies and costs that affect the overall efficiency of companies. This report identifies how DevOps engineering principles and implementations provide solutions to common problems in globally distributed work environments. It uses a literature systematic literature search and review to extract the recent and relevant academic data in the scope of the two research questions. Then, a proof-of-concept is implemented for DevOps, which confirms the literature. In parallel, a survey addressed to Swedish companies provides subject-related data from the professional environment, which largely supports the literature and brings extra knowledge. All of this is considered in data analysis and formulation of conclusions, showing DevOps features that can improve and support work in globally distributed environments and outlining the importance of the tailored organizational culture for the modern need of large-scale distributed work.

Keywords
DevOps, Concepts, Engineering Practices, Geographically Distributed Work, Remote Work, CI/CD, Automation

Table of Contents

1 Introduction

1.1 Problem Definition

In the current context and the one that is expected to follow, the geographically distributed work will be an almost obligatory component of the companies' new organizational schemes. Both remote and distributed work is not new. These have been studied over time, but now it is imperative to understand better and streamline these forms of collaboration.

Geographically distributed work (GDW) has its challenges related to many aspects of the organizations' activities. Some relate to organizational culture, others to cultural differences within different teams, others to technical or methodological choices of organizations [1].

DevOps seeks to create an environment that improves collaboration and communication coherently and cohesively, enhancing performance by automating processes traditionally performed manually and changing the organizational paradigm [2,3]. This research investigates how DevOps project management methodology principles can improve efficiency and support globally distributed work environments.

1.1.1 Agile Methodologies

Agile adds business agility to the project management process; it promotes working in 'sprints' by breaking down tasks into smaller parts, leading to faster delivery of product features. What is specific about the agile method is that it appreciates more customer collaboration than contract negotiation. It implements an iterative, incremental approach to software development. A customer can test the product early and more often and give feedback at an early stage allowing developers to make changes at any point of the development. At the end of each 'sprint,' a user evaluates the product and accepts the progress. This report will briefly cover Agile project management methodology because it is considered that DevOps is an extension of it [1].

1.1.2 DevOps Concepts

DevOps is an approach to software development seen as an extension of Agile methodology [1]. It is a paradigm that mainly focuses on better collaboration between developers and operators. DevOps offers scalability, fast delivery of products, and reliability through continuous testing, continuous integration, and continuous delivery [2].

1.1.3 DevOps Environments

One of DevOps' key concepts is Continuous Integration/Continuous Delivery (CI/CD), a set of principles and practices that enables automation of specific processes. Its implementation is known as the CI/CD pipeline. DevOps environment utilizes DevOps concepts and includes all needed tools for the DevOps method of working. It is a set of tools and a collection of resources that developers and operators use during product development. Tools included in this environment depend heavily on the type of project. Such an environment aims to reduce bottlenecks such as a single person, usually a DevOps engineer, responsible for building and maintaining servers that can only be managed by the person who built them. DevOps environment relies on heavy documentation and automation, which, as a result, enables developers and operatives to communicate more. It is not a well-defined project management approach, and it can be

met as a feature in different other existing project management environments that try to transition to DevOps.

1.1.4 Remote Work

Remote work is briefly defined as an organizational, procedural arrangement that involves employees or collaborators performing tasks under their responsibility from different physical locations, which can often be their homes. This mode of operation can generate multiple benefits for employees, collaborators, and organizations. Simultaneously, it is a paradigm that must be considered rigorously and requires a slightly different approach from the one in which the organization's activity takes place at its headquarters [3].

1.1.5 Distributed Work

Distributed work involves carrying out activities as part of a team outside the company's headquarters and involves a new mentality within the organization, which consists of streamlining the activity due to the focus on collaboration and overcoming barriers of a geographical, temporal, cultural nature. While remote work focuses on individuals needs, distributed work focuses on the whole organizational environment [4].

1.2 Aim & Purpose

In addition to the apparent pandemic generated needs, companies understand that GDW brings many benefits. This can consist of cost reductions, access to more human resources and globally distributed talent, encourages dynamism and innovation, and others. All this, however, comes with a price. Some of the factors that create conflicts in these environments are the lack of communication and the difficulty of finding standard interconnection methods between different entities, finding a common path of inclusion, respect, reward, motivation, vision, sharing, access to resources, and others [3].

Given that DevOps responds to the need for efficient communication and skillful interconnection of different teams and that automation avoids many situations that can lead to misunderstandings or friction between the parties, we wonder if all these problems can support and solve parts of GDW problems.

Our online research found no express studies linking DevOps and GDW environments. Separately, however, different studies present GDW software development environments' challenges and the advantages of DevOps environments [5,6].

This degree thesis report investigates how DevOps can participate with principles, engineering practices, and organizational culture to streamline GDW environments' performance.

1.3 Research Questions

The two research questions presented below in this subchapter are the starting point of this report. These are landmarks considered in developing working methodology, literature review, proof of concept, and survey. These are taken into account when analyzing the results and formulating the report's conclusions.

1.3.1 Research Question 1 (RQ1)

How can DevOps contribute to embracing a distributed style of working?

DevOps embraces continuous integration, continuous development, and process automation; this question aims to determine how these aspects relate to adopting or

improving the distributed style of work. The question's scope is limited to the DevOps concepts and engineering practices; how can they solve some of the identified core problems of software development teams that embrace distributed work?

1.3.2 Research Question 2 (RQ2)

Do companies think that DevOps methods can improve the new pandemic context's distributed work conditions? How? What other options do they have?

To gain another perspective apart from the literature review result, we include several companies in our research, regardless of whether they work within a DevOps environment or not. This research question aims to familiarize with other existing solutions that proved in practice to broaden the view of how we look at DevOps concepts, distributed work, and the connection between the two in the context of pandemic constraints.

1.4 Literature Review

The literature review aimed to define the concepts approached and how they can be applied and exist in organizations. The concepts, practices, and models associated with DevOps and distributed work were addressed and used to create the case study, the survey, and data analysis when formulating conclusions.

1.4.1 Research and Implementation

1. Proof-of-Concept

As a proof-of-concept, a CI/CD DevOps server is deployed, and the practical work efficiency in a distributed environment is tested against the literature review conclusions. This part aims to outline if and how DevOps practices can enhance the overall performance in the distributed way of working considering software development teams.

2. Survey

Data is collected by conducting surveys sent out to many software development companies in Sweden to parallel existing environments and the literature findings. This survey focuses on the real issues that companies face regarding the distributed work. The results are compared to the literature findings to conclude if the theory works in practice.

1.4.2 Data Analysis & Discussions

The data analysis summarizes the results obtained from the three sources: literature review, DevOps proof-of-concept, and survey. The data obtained from the literature review cover three different categories: DevOps concepts, DevOps engineering practices, and globally distributed work. The data obtained from the proof of concept and survey are analyzed to confirm and complete the data obtained from the literature. Analyzing all these, the connections between different sources and findings are made to formulate relevant conclusions.

1.4.3 Results & Conclusions

The conclusions of this thesis are based on the analysis of the result. They will overview the dynamics of distributed work in software development environments and how DevOps concepts and practices can bring extra performance in this regard.

1.5 Expected Results

In response to RQ1, we note that DevOps has methods that improve remote distributed work. DevOps automation methods and technologies used to interconnect different teams and improve communication between a distributed work environment can reduce conflicts caused by a possible lack of communication or inconsistent visions whose globally spread work environments may be susceptible. The implementation of our proof of concept and the search literature shows how DevOps can do this.

For RQ2, the result shows that companies believe that DevOps environments can better adapt and integrate distributed parts, given the organizational culture oriented towards integration, communication, and automation. We can also conclude the changes considered by the companies participating in the survey to adapt to this new challenges of distributed work. Some companies intend to switch to DevOps, and others want to stay with the currently used project management methodology.

1.6 Limitations

The identified limitations of this project were due to the following:

- COVID-19 associated restriction (100% distributed work).
- Both co-authors did their best to approach the topic in a relevant and impartial way.
- No access to physical resources in the university for building the proof-of-concept's environment.

1.7 Acknowledgments

This report was made possible with the support and coordination of the university where both students are enrolled. The subject itself is the result of a consultation with the group's supervisor, Daniel Einarson. Also, the content is adjusted according to the feedback received after a first examination done by Kamilla Klonowska.

A significant factor that helped approach the companies participating in the survey is the support received from Mobile Heights through Ola Svedin.

2 Methodology

This research report is based on the conclusions of the literature review analysis, the case study, and the survey.

2.1 Literature Review

- ***DevOps concepts*** (for RQ1)

Extensive literature research about DevOps concepts will be executed to identify DevOps concepts and their principles. The knowledge gained in this part of the study will be presented in the introduction to familiarize the reader about DevOps in general. The findings in this category will also be included in answer to research question 1.

- ***DevOps engineering practices*** (for RQ1, RQ2)

The practical part of the research would be deploying the DevOps CI/CD server with *the essential DevOps tools*. In creating a genuine DevOps environment, findings from systematic literature review (SLR) regarding the DevOps engineering practices will be of great importance.

- ***Distributed work principles and challenges*** - identifying core problems (for RQ2)

By offering solutions for distributed work challenges, the question: "What are the core problems regarding the distributed work?" has to be answered. By conducting the literature research on distributed work, the aim is to identify and isolate the core problems of distributed work style and the suggested or applied solutions.

2.2 DevOps Environment Implementation Proof-of-Concept

As a proof-of-concept, a server with a DevOps environment is deployed on a Linux server virtual machine and includes tools such as Docker and Jenkins. The choice of Jenkins was made due to literature findings stating that CI and CD are the core concepts of DevOps [2]. Jenkins covers the CI part of the project as confirmed by the literature review. The proof-of-concept aims to prove literature findings on the subject of automation of specific processes during development, utilizing core DevOps concepts, and help to realize if this environment can be of help when working on a project with a collaborator in a distributed manner.

2.3 Survey about DevOps environments perception and distributed work

To get a clearer view of the current situation in the present regarding the globally distributed style of working, the use of DevOps in general, and possible solutions to existing problems of globally distributed work, a survey is conducted. The survey includes questions about challenges in GDW, the use of DevOps. It discusses possible solutions to the difficulties in GDW and is sent out to several software development companies based in Sweden.

2.4 Data Collection & Analysis

The study of the literature will focus on research questions. The conclusions about DevOps environments' known benefits are highlighted, subsequently analyzed, simulating the DevOps environment. The interest is focused on how this concept improves efficiency and whether it solves the fundamental problems of distributed work style (if any).

Survey responses are centralized and analyzed to understand the state of affairs within companies using DevOps or other organizational cultures and find out how they view geographically distributed work and its relationship with DevOps concepts.

3 Literature review

The systematic literature search aimed to identify the relevant sources for the research for this paper's topic. As this research is done as a degree thesis at Kristianstad University, the HKR Summon search engine provided access to multiple databases. It offered plenty of reading materials related to the research subject. The queries had the same search options and different keywords. Peer-review articles from sources other than HKR Summon were also used, such as ResearchGate, Google Scholar, Academia.edu, and others. Three distinct searching categories have been created/used to cover the components of this paper. These categories are meant to cover the literature on DevOps concepts, tools, models, engineering practices, and the core problems and possible solutions within distributed work environments.

3.1 Systematic Literature Search

The *content types* that were chosen were: Archival Material, Article, Computer File, Conference Proceeding, Data Set, Dissertation/Thesis, Electronic Resource, Exam, Government Document, Journal / eJournal, Journal Article, Library Holding, Magazine, Magazine Article, Manuscript, Market Research, Paper, Patent, Presentation, Publication, Reference, Report, Standard, Student Thesis, Technical Report, Trade Publication Article, Web Resource.

The search *discipline* options that were considered in this search were: applied sciences, business, computer science, education, engineering, environmental sciences, international relations, law, library & information science, medicine, occupational therapy & rehabilitation, philosophy, physical therapy, political science, psychology, public health, sciences, social sciences, social welfare & social work, sociology & social history, statistics.

The *languages* considered in this literature search were: English, Romanian, Serbian and Swedish.

All of the results were *limited to* items with full text online, open access items only, and peer-reviewed publications with an expansion to include results from outside of Kristianstad University library's collection.

Excepting the literature search for remote and distributed work, the search period was set between the years 2015-2021. The search was split into three categories: 'DevOps concepts', 'DevOps Engineering Practices and environments', and 'Remote and Distributed Work Definition and Challenges'.

3.1.1 DevOps Concepts

This search category covers the Agile methodology basics and DevOps principles. For the category of *DevOps Concepts,* the search query was: *'(Fulltext:(DevOps)) AND (Fulltext:(Concepts))'*.

As a result, this query gave one hundred articles, of which only ten were relevant to the subject of this research.

Out of the ten results, five articles were selected as the most relevant for the research.

3.1.2 DevOps Engineering Practices and Environments

DevOps Engineering Practices and Environments category covers the engineering practices, tools, models, and implementation of the DevOps environment.

The search query used for this category was: *'(Fulltext:(DevOps)) AND (Fulltext:(Engineering)) AND ((Fulltext:(Practices)) OR (Fulltext:(CI/CD))) AND ((Fulltext:(Pipeline)) OR (Fulltext:(DevOps))) AND (Fulltext:(Deployment)) AND (Fulltext:(Automation))'*.

A total of one hundred and thirteen articles were the result of this query, of which only thirty-five were fit for the research paper. Out of the thirty-five relevant results, a total of seventeen articles were selected.

3.1.3 Remote and Distributed Work Definition and Challenges

What remote work and distributed work are, and the differences between the two is just a part of the answer that will be realized in the category *Remote and Distributed Work Definition and Challenges*. This category will identify challenges that exist in the distributed and remoted work styles.

The query that was used for this purpose is: *'((Abstract:("Remote work")) OR ("Distributed work") OR (Globally distributed work")) AND (Abstract:(Challenges)) AND (Fulltext:(software development))'*.

The result of this query was thirty-four articles, of which only nine were relevant to the subject of this research.

A total of five articles were selected from the nine relevant results that fit best the research questions.

3.2 Systematic Literature Review Selection Criteria

The relevant results were chosen by reading the titles, abstracts, introductory parts, and conclusions, taking into account the document's structure. The most relevant articles were filtered by reading the entire content, considering the data's relevance for the three research categories mentioned above. Papers with repetitive or identical content and those with too elaborate, advanced, or irrelevant content related to this paper's purpose have been discarded.

3.2.1 DevOps Concepts

The concept of DevOps, together with the arguments for its utilizations, was the main focus of the article selection. The papers that offered clear and concise definitions and examples were considered relevant for this research category.

3.2.2 DevOps Engineering Practices and Environments

This category seeks to explain different DevOps models, DevOps culture, and working environment. The criteria for selecting the articles regarding this category were based on existing practical examples of DevOps implementations in companies, tools used in DevOps environment creation, and the tools that the environment should include the proven outcomes of adopting the DevOps paradigm. Articles that were considered relevant for this part of the paper have DevOps environment pros and cons and evaluate existing DevOps environments.

3.2.3 Remote and Distributed Work Definition and Challenges

Concerning this category, the article selection bases were to pick out those papers that include explanations of geographically distributed work regarding the software development environments, challenges that occur when working in a distributed work style, and solutions that help overcome the obstacles concerning the distributed type of work. The articles that outline the difference between remote teams and the distributed teams were also considered relevant to this research paper.

3.3 Systematic Literature Search Selection Evaluation

Following this literature search, it was found that there is a recent growing interest in DevOps. Most articles focus on the differences between Agile and DevOps and how

companies can implement DevOps environments or address this approach's components, such as CI or CD, automated testing. There are also many references to how DevOps reduces costs and streamlines processes through automation, accountability. The same is true because DevOps focuses on team culture and streamlines overall cooperation and performance. However, few recent articles directly address distributed work in software development environments using Agile methodologies. No result has been identified that directly links DevOps environments' benefits as potential solutions to distributed work environments' challenges.

3.4 Literature Review Result

3.4.1 Agile Methodology

Agile is a project management methodology introduced in software engineering environments to streamline performance, meet new customer requirements, and make it easier to adapt to new contexts or data (*Figure 1*). This need was due to previous methodologies' inability to cope efficiently with frequent changes (e.g., Waterfall, PMBOK, PRINCE2) [1]. Agile streamlines create software products by breaking down tasks and solving them iteratively into 'sprints' that can take several weeks and lead to faster delivery of components. For this, teams with specific skills and responsibilities that work in parallel are used. Agile changed the operating paradigm from a process-centric to a people-centric one. It can be implemented in various forms, such as SCRUM, XP, RUP, or various other models, with a specific orientation to the organizations' requirements or needs [1,7]. The emphasis is on agility and collaboration between different functional modules (silos).

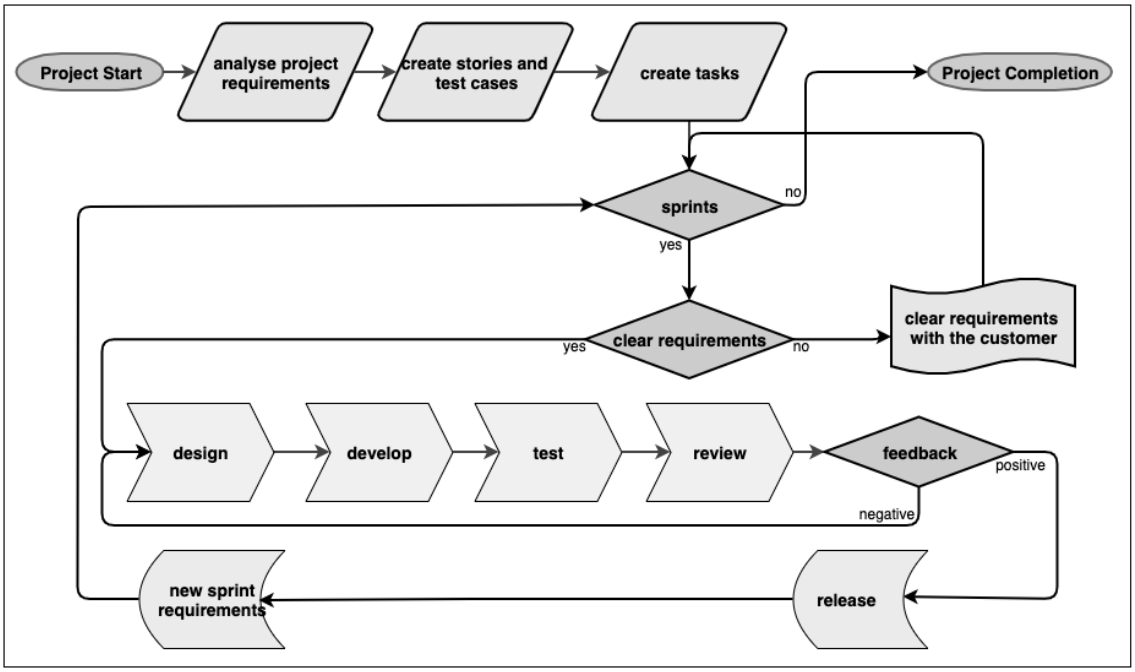


Figure 1. Simplified Diagrammatic Representation of Agile Methodology

The repetitive evaluation of each component at the end of each sprint streamlines the testing, production process. It also offers customers the possibility to contribute to the production process by providing feedback for every component. This improves production speed and customer satisfaction and allows companies to adapt to market requirements, customers in real-time and at reasonable costs.

Agile is a software development project management methodology that has many recognized qualities and some disadvantages, briefly presented in *Table 1*.

Table 1. Agile Methodology Advantages and Drawbacks

Agile advantages
Creates business agility
Client satisfaction orientation
Speed up the software engineering process
Supports individuals and increases the team collaboration
Prioritize the working software over documentation
Enhances the adaptability to new requirements
Reduces organizational costs
Different teams work in parallel, avoiding the 'idle' state
Agile drawbacks
Possible miss-communication between the Development team and Operations team
Components can be not compatible with each other and generate delays
The scope of different functional teams (silos) can be slightly divergent and cause conflicts, delays
The silos can cause or sustain a 'us and them' mentality that can lead to miss-communication or conflictual interest
Before the release, the quality of the product can not be a guarantee

3.4.2 DevOps Concepts

DevOps' beginnings were attempts and practical solutions to eliminate the frictions and problems between Development and Operations. Later, when these solutions proved their effectiveness, they were adopted under the title 'DevOps'. Meanwhile, DevOps has developed into an organizational culture based on beginning principles and new ones. It can be seen as a *process innovation* [8] that addresses issues identified in organizations using Agile methodologies or as an extension of Agile that addresses *organizational concepts* that streamline performance and reduce friction between different Agile silos [9]. DevOps does not directly address new processes or tools but creates a *different paradigm* that integrates the same functions/processes in an environment focused on *better collaboration, transparency*, and *process automation*. The main issues that led to this new concept were the need to reduce problems that traditionally appear between development teams and IT operations teams and provide better quality assurance before the release [10].

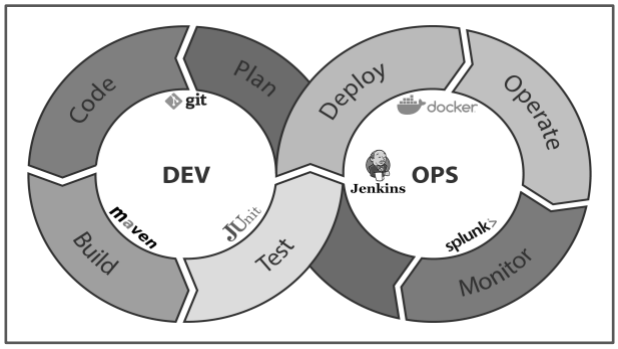


Figure 2. DevOps life-cycle and possible tools setting

DevOps is conceptually described as *organizational culture*, a mindset meant to bridge the gap between different expertise (like development and operations engineers), cultures, or mentalities, replacing them with an organizational culture focused on inclusivity. This is the core of the DevOps principles, and the way this goal is pursued is through massive process automation where possible, using integrated tools. This way, it allows everyone involved to understand and access all the information related to each stage of the DevOps life cycle, which is repetitive and continuous (*Figure 2*).

Researchers define that DevOps identify the importance of organizational culture and the closeness between software development and implementation. The acronym 'CALMS', when associated with DevOps, stands for its main features: *Culture, Automation, Lean, Measurements, Sharing* [11]. The organizational culture is essential in DevOps and supports shared responsibilities and skills that emphasize communication and collaboration within software development and operations environments. To achieve that, DevOps proposes the focus on treating the infrastructure as code and *continuous integration and delivery* [1]. This facilitates process automation and constant and fast testing, validation of each release. All this can be done using different tools meant to *bridge the gap between development and operations*. The two traditional Agile silos, Development and Operations, are replaced by a single multifunctional team that takes over the two's responsibilities. *Testing is done automatically* with specially built tools and is performed when new commits trigger builds. Broken builds are automatically noticed and generate new issues that the DevOps team can handle promptly. So, the information is used promptly and sustains continuous delivery and deployment. DevOps focus is on automating as much as possible of the workflow.

Table 2. DevOps Core Principles

DevOps Core Principles
Improved communication
Collaborative development
Continuous integration, delivery, testing, deployment, monitoring, and feedback
End-to-end control and responsibility
Customer-centric approach
Shared ownership, values, responsibilities
Shared knowledge, competencies
Process automation

The main DevOps principles are continuous integration, delivery, deployment, continuous automated testing, continuous monitoring, end-to-end control, collaborative development, shared responsibilities and competencies, customer-centric (*Table 2*). They also somewhat reflect the DevOps cycle that is repetitive and continuous. The way and extent to which all these goals are achieved differ between different implementation models and existing environments that aim to transition to a DevOps approach [12]. The DevOps features are meant to create a *collaborative environment* based on communication and an objective understanding of the issues discussed [9]. Automation eliminates human error, delays, provides conformity, and maintains transparency in all processes. All this is possible due to the *extensive competencies* that DevOps members must acquire. Even though there are different areas of expertise for each member, they all have common competence regarding the product's entire life-cycle. *Shared values and responsibilities* accompany *shared knowledge* [9].

3.4.3 DevOps Environments and Engineering Practices

DevOps helps development teams with continuous planning, continuous integration, and constant feedback, according to Sushma T N [2]. Findings gained from the literature review on *DevOps Environments and Engineering Practices' context* identify that the standard DevOps environment should include the following parts: CI/CD, Project management, Version control system, build and configuration management.

Continuous integration (CI) includes code compilation, sanity check, and building of deployment packages. This process is automated and is executed as soon as a change is

made to the code [13]. *Continuous delivery* (CD) is a process at which small software releases are developed in short cycles and, together with CI, stands as a critical concept of DevOps [14]. *Continuous deployment* is a step up from CD and an automated process that carries out software builds deployment to a repository instead of end-users. The step-up is that instead of deploying minor releases like in CD, every change in the source code is deployed automatically [13]. *Testing* in the DevOps environment is automated and requires no user engagement. The aim is to reduce human error and speed up the error detection of the builds [13]. According to Jayasri, Srinivas, and Gutta from the Malla Reddy Institute of Technology, *project management* is a complex phenomenon that requires high maturity. Various risks and issues should be identified and resolved as soon as possible, which points to the importance of this part of the DevOps environment [15]. As both CI and CD terms are previously explained, it is worth mentioning that they together form a vital DevOps toolchain (*Figure 3*). *Version control* is a part of that toolchain and is crucial for tracking code modifications by storing them in a special kind of database. A minimal CI/CD toolchain should include, at least, some version control tool besides the tools for automation, tests, and package managers [14]. Build configuration issues can manifest in long building times and multiple and failed build attempts [16]. This outlines the importance of including *build and configuration management* tools in the DevOps environment.

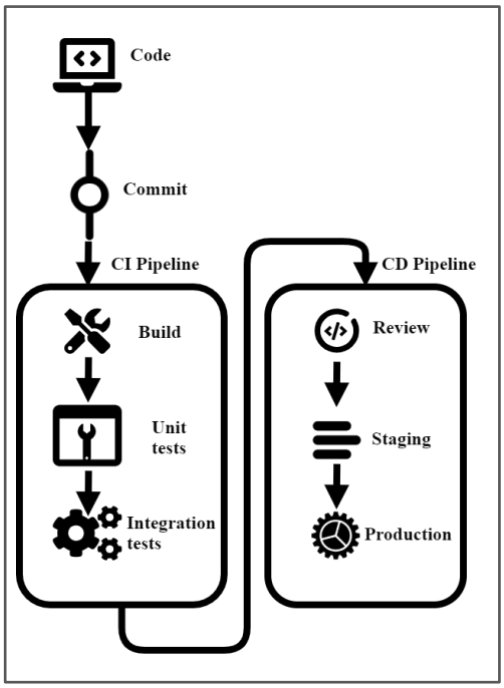


Figure 3. CI/CD pipeline diagram

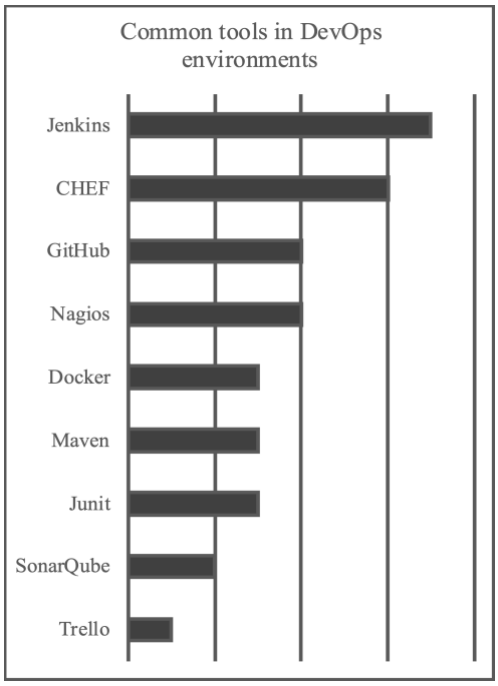


Figure 4. List of common tools in DevOps environments

Georges and Asif researched the subject *DevOps: Concepts, Practices, Tools, Benefits and Challenges* where they identified a set of 20 DevOps practices. The three practices that were adopted the most among DevOps teams were [17]:

- Sandbox deployment is automated through the development pipeline
- A DevOps team carries out the synchronization of critical services such as uptime, performance, deployment schedule, transactions, run-time costs, version control, and project scope
- Release deployment is fully automated across the development pipeline.

These practices point out the fundamental concepts of the DevOps environment, which are automation and synchronization. *Figure 4* shows a list of "common" tools included in DevOps environments. This list was created by examining the selected papers for the

literature review and measuring the frequency of the tool used in different DevOps environments [18,16,19,2,13,17,20,14,21].

3.4.4 Remote and Distributed Work

Remote work arrangements and distributed work are not new issues in IT organizations [22]. Since the '80s, working from home and remote work were defined as an organizational arrangement that allows employees to use technologies that can enable them to work from home regularly to substitute for regular workplace attendance. In many papers of that time, technology is considered support for remote work [23,24,22,25]. It expanded after the 2000s to enable organizations to access globally distributed talent [26,25], which can be seen as a strategic advantage on a competitive market and increase operational efficiency [25,27]. Today, most organizations implement a hybrid working environment [27]. In the new pandemic context and the immediate future one, distributed work seems a necessity. The efficiency of distance work seems necessary to respond to the new problems generated by this new context. Organizations need to find effective ways to support globally distributed work and collaboration, cooperation [3].

While remote work seems to involve the employee working at a different location than the company and concerns individuals, distributed work implies that the employee's work environment, tools, and possibilities do not vary whether they work in the office or remotely and concerns the whole organization methodologies and culture. In both cases, the company for which the employee works must provide the necessary facilities and environment for this work. Besides, companies must provide firm work directions, communication and collaboration opportunities, good human resource management, embrace diversity, and have a clear vision of work politics and conflict management [3]. As an enhanced form of distributed work, GWD can bring different benefits to both companies and employees [22]. It refers to work from home or other remote locations and the global distribution of human resources. It provides a global openness to the workforce and allows organizations to access global talent. However, all this requires additional efforts and new strategies, training, and an adequate framework - both physically, materially, technologically, and as a culture and organizational mentality.

Some of the aspects to consider in the implementation of a geographically distributed work environment are: ensuring a suitable separate work environment, dedicated to working; good internet connection and access to the necessary IT tools and systems; the possibility to interact easily with colleagues; clear and timely guidance from senior colleagues or supervisors [3]. The most common tools used in globally distributed software development environments are concerning project management, collaboration, and code version control [5]. In globally distributed work environments, the software development process is widely known as global software development (GSD) [28].

3.4.5 Distributed Work Challenges

In the new pandemic context, many companies and their employees had to adapt to the new reality and find solutions to continue their activities, considering social distancing and maintaining adequate health security. This sub-subchapter tries to identify the most relevant challenges in a geographically distributed work software development environment.

Of course, solutions have already been implemented in the labor market, such as part-time or full-time work from home or other locations outside the organizations' headquarters (remote work) or geographically distributed work. Also, there were already different project management methodologies that facilitate collaboration, communication, and performance. However, the transition from office to remote work, especially in jobs traditionally performed onsite, proved to have many challenges and requirements. This

situation highlighted the need to *streamline collaboration and communication* in these work environments and the need for an *organizational culture* appropriate to this context [29,27].

Communication is an essential factor in any organization, but it is a vital vector in those whose activity is geographically distributed [30,5]. The lack of face-to-face communication can *diminish the affinity and empathy between colleagues*, which can be a cause of conflict [31]. Efficient collaboration and communication prove to be essential and an advantage in a competitive business environment. In distributed work environments, employees have *fewer opportunities to contact their peers* and create informal communication streams [22]. In the absence of an adequate organizational environment, geographically distributed work can suffer from ineffective communication and collaboration [5]. Challenges are also meat in areas like *efficient control or transfer of software products* [30,32].In environments that focus on globally distributed work, the *organization's culture* is of utmost importance, this being one of the significant factors of successful activity, as Conway's law states. Organizational design has a significant impact on product design [33].

The organizations' considerations must include their employees' particular nature requirements, such as *time zones*, the need for *interpersonal connections, psychological comfort, cultural differences* [5] and specific *holidays, schedule* availability, *stress* due to misunderstood requirements or expectations, and possible *social isolation* [29]. If not considered correctly, all these problems can lead to collaboration problems, tensions, misunderstandings, and conflicts [5,34]. Organizations must also consider the tendency to create groups based on affinities or identities, the so-called *'us vs. them'* attitude [4], which over time, without properly manipulated, can cause conflicts and lead to critical situations. The *lack of face-to-face dialogue* [27]can accentuate these tendencies and lead to situations when individuals feel their identity is threatened. *Identity threats* [30,34].
Not using a common set of tools can lead to *process management differences*, possible *duplication of work*, and *inefficiency in information sharing* [30,28]. All this can generate *delays* and *additional costs* for the final product. Challenges can be *linguistic* in nature as well. Even if they can speak the same language, *different dialects, customs, regionalism*, they can cause separation and erosion of communication and collaborators. Same when *stereotyping people* based on their region or country [5].

Other factors that can create conflicts that affect collaboration and communication efficiency are *knowledge, power, resource asymmetries*. Shared understanding creates bounds and facilitates communication. Not understanding the other's position on a subject can make communication harder or compromise and create coordination issues [34]. Power asymmetry means that not everyone involved has access to all relevant information or technology. It can cause conflict, misunderstandings, cause costs to both companies and employees. Resource asymmetry refers to the possible incapacitation or impediment of teams or individuals from functioning to their maximum capacity by limiting or conditioning direct access to the necessary resources [34].

3.4.6 Distributed Work Solutions

The previous subchapter presented the challenges of distributed work environments. This subchapter offers the solutions suggested by the authors of the studied works.

A *flexible work schedule*, adapted to employees' location and needs, can motivate and maintain a productive and high-quality pace. This can increase employee comfort and *reduce stress* levels [32]. Encouraging and providing *informal conversation opportunities* between employees, like *face-to-face communication, video calls*, and *phone calls*, can allow teams *to raise issues rapidly*, find better and faster solutions, and accelerate product development speed GDW environments, increasing customer satisfaction [32,22]. This

can be supported using *state-of-the-art cloud-based collaboration tools*, such as *communication platforms* and good equipment, to *access and distribute information and technologies* [22,5] easily. These can reduce possible duplicate work cases and help *maintain a clear record of ownership, responsibilities, and developmental or managerial processes*.

Effective collaboration through state-of-the-art cloud-based tools can solve many of the problems of GDW environments. *Improving communication, collaboration*, and *access to information* and *streamlining data, technologies, and experience* [35] are vital elements that can enhance these work environments providing faster delivery of a workable solution and greater client satisfaction.

Integrated systems can help overcome communication challenges in GDW. They can support processes that can solve issues generated by different physical locations, misunderstood responsibilities or requirements, ambiguous roles, *reduced communication effort required between various entities*, and improved *conformity* and *accountability*, increasing *cohesion between the teams*. They also support efficient data sharing, providing a unified, standardized approach for all previously mentioned aspects [32,25,34].

These tools can make GDW teams *not feel isolated* and perceive *symmetry when it comes to power, resources, and knowledge*. This reduces issues such as identity threats, "us vs. them", or the stereotyping of individuals or groups. Access to information and technology to *standard business processes* can create connection points in these environments. It can *highlight common elements* easily identifiable by employees, like *shared competencies, skills, knowledge, responsibilities,* and others [27]. This can overcome prejudices related to the cultural background or other attributes of a personal nature, *improve mutual respect*, a shared sense of purpose, and *encourage different teams to collaborate willingly on distributed processes* [32,5,34].

Process automation using cloud-based technologies can create a *transparent work environment*, where requirements are easy to understand, easy to involve in and provide a firm, depersonalized *feedback*. Individuals can quickly adapt to this environment and can feel *treated and evaluated correctly*. They can feel that their work is appreciated correctly and an active part of the ongoing processes [32,25].

Automation can also support accommodating teams' or individuals' schedules, located worldwide, at *different time zones*. Their work schedules can be adapted to their location's reality, keeping the processes continuity and *prompt feedback*.

Although state-of-the-art cloud-based modern tools can solve many GDW's management problems, there are issues related to human nature and organizational culture that organizations need to consider appropriate. The potential challenges due to these issues can be solved within organizations considering these differences appropriately, *training staff in intercultural interaction and communication*, using the presence and input of *specially trained multicultural brokers* [4] (to manage these situations), emphasizing collaboration and acceptance of differences *by identifying the things that are in common*. They can bring teams and individuals together and improve GDW environments' quality, reducing the possibility of feeling socially isolated [5,29].

4 Proof-of-Concept

4.1 DevOps Environment Design

The typical DevOps cycle (*Figure 2*) consists of eight stages: Plan, Code, Build, Test, Release, Deploy, Operate, and Monitor [6,20]. These parts are considered during the design phase of the minimal DevOps environment needed to realize the proof-of-concept. Operate and Monitor stages are not implemented as they are focused on the application's stability, which is not required for such a small-scale project that will be used as a proof of concept. As mentioned before in chapter 3.4.3 '*the minimal CI/CD toolchain should include, at least, some version control tool besides the tools for automation, tests, and package managers'* and this environment is designed to cover that minimum to test the efficiency of DevOps core concepts regarding the development in a distributed style of work. The proof-of-concept project is a simple web JSP application that has no other purpose but to confirm literature findings regarding the efficiency of DevOps environments regarding adopting a distributed style of working in software development.

4.2 DevOps Environment Implementation

The environment for proof-of-concept is set up on a *Linux server* (20.04.2 LTS) installed on an *Oracle VirtualBox* virtual machine. It includes the *Jenkins automation server*. Its role is to automate different processes in the development of the proof-of-concept project. The platform of choice for version control is *GitHub*. Planning is carried out on *Trello*, and communication between collaborators is done through *Discord*, *GitHub* issues, and *Slack*. *Apache Tomcat* is a web server that is chosen for deploying the JSP proof-of-concept project artifacts.

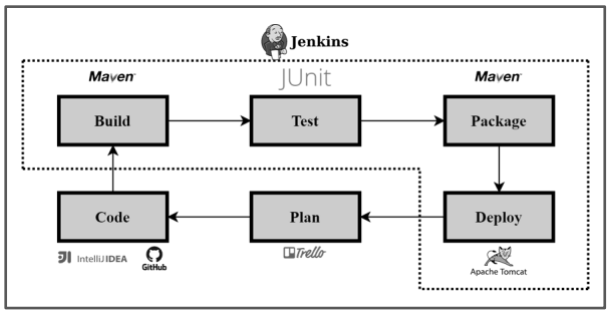


Figure 5. Proof-of-Concept implementation diagram

Figure 5 shows the development cycle of the proof-of-concept project, including the tools responsible for different tasks. The cycle starts with planning software used to track project progress and includes a Kanban-style list of tasks. The second step is writing the code according to the schedule from the *Trello* task list. After committing the code to the *GitHub* repository, *GitHub* notifies the *Jenkins server* that the change is made and starts steps defined on a *Jenkins server*. *Jenkins* then reads the *Jenkinsfile* (*Jenkinsfile)*, which includes the pipeline code with instructions on what to do in each stage of the pipeline. The first stage in the pipeline is *'Build'*. In the '*Build'* stage, code is compiled and checked for any compiling errors. The message is sent to the *Slack* channel containing the name of the project, branch, and a build number, and the result of the build (*Slack notification*). If the build is successful, the pipeline proceeds to the next stage, '*Test*'. '*Test'* stage runs a *Maven* test script that starts pre-coded *Junit* tests. If the build is successful, the pipeline proceeds to the next stage, '*Package*'. In the '*Package'* stage, the code is built and packaged into the .war file, and the next stage is started. The last stage is '*Deploy*' where, based on the branch, either the .war file is deployed to the *Tomcat server* - if the branch is main (the production branch), or a simple message is sent to *Slack* to notify the developer that the code is good for the pull request to be created. Suppose any errors are encountered during these stages. In that case, the pipeline will stop at the stage where the error has occurred, giving the detailed log about the error and notifying the developer on *Slack* that the stage has failed. This ensures that the deployed file is the one that passed all steps of the pipeline and is stable for production.

4.3 DevOps Proof-of-Concept Experiment Steps

Things needed for the environment setup: Oracle VirtualBox, Linux server installer.

The first step was to set up a Linux server on our machine. Because we use Windows, the same OS version of VirtualBox is downloaded. After the download, the VirtualBox is installed. Then we downloaded the Linux server installer from the Ubuntu download page.

The Linux server version used for this proof-of-concept was 20.04.2 LTS. After the installer was downloaded, we opened VirtualBox and set up a virtual machine for the Linux server. In VirtualBox, we created a virtual machine following the wizard dialog using all default values. The following settings were used for the virtual machine from the default values: Name - Linux Server, Type – Linux, and Version - Ubuntu (64-bit).

When the virtual machine was created, it was time to install the operating system (Linux server). The virtual machine was powered on, and upon the first run, the prompt window appeared asking for installation media. The Linux server was downloaded as an iso file, so we clicked on the folder icon in the prompt window, navigated to our iso file, and opened it. By clicking on the start at the dialog prompt window, we booted our virtual machine with the virtual disk (iso file) containing the installation files for the Linux server. During the Linux server install, we used all the default values except the hostname, user full name, username, password, write partition changes to disk, write to disk, and GRUB boot loader.

After these steps, the Linux server was successfully installed. We booted up our virtual Linux server and started the Linux terminal. We then installed Java open JDK and the Jenkins LTS automation server on our Linux Server machine. The first run of the Jenkins server started the post-installation setup wizard in which we used all the default values. The new user was created through the wizard with new credentials. Jenkins was accessible through the web browser on the localhost address and port 8081 (http://localhost:8081), which was forwarded to make it accessible remotely. In Jenkins, we installed the following plugins: Slack Notification, GitHub Pull Request Builder, Credentials, and Deploy WAR/EAR to a container Jenkins.

Tomcat server was installed on Raspberry PI, and a port to it was opened to access it online the same way as the Jenkins server. The pipeline was set up on Jenkins, who listened for changes on our GitHub repository through the webhook. The JSP project was built, tested, and packaged into a war file deployed to our Tomcat server. For the code writing, we used IntelliJ IDE. The project was split into five tasks: initial JSP template, Title change and adding a heading to the page, adding a paragraph with dummy text (lorem ipsum), adding basic CSS styling, and adding current time/date display with the live second counter. After each task, a commit was made to the master branch, which triggered the Jenkins pipeline.

5 Survey

5.1 Scope

So far, the essential distributed work environment issues have been approached strictly through the literature review, studying the relevant papers in this regard. Also, how DevOps participates in improving communication and collaboration was identified, streamlining the software development environments' processes. In parallel, companies who implement DevOps have different stages of transitioning or their owns views and approaches. Some have entirely transitioned to DevOps; others use DevOps principles in specific processes; others try to transition to DevOps by following steps appropriate to their resources and capabilities. Also, some companies do not use DevOps at all, opting for other project management methodologies.

To better understand the realities within organizations and how they approach the challenges identified so far and find out new possible opinions or visions, a survey tries to address these issues and interact with these organizational environments.

5.2 Implementation

The survey is built in a tree-like structure. The questionnaire aims to find the interviewed companies' vision about the distributed work, its challenges, and the project management methodologies used. Also, the survey aims at their vision regarding the present state and how they look at the near future, GDW/DevOps (or currently used project methodology) related. The platform used for this purpose was GoogleForms. The companies that have chosen to get involved in this research are NordAxon, Ericsson, Abstraktor, Elastic Move, and Xlent Analytics Syd (*11.3*).

Depending on the respondents' answers, in the essential cases that delimit significant aspects, new branches are opened to address the interviewee's specifics. The aim is to obtain a greater relevance of the questions addressed to each participant. Nodes opening new branches address issues considered essential during the survey. They aim at the following:

- Does the interviewee consider that GDW is on an upward trend if there is a need for the future?
- Does the respondent's company use globally distributed work?
- Does the respondent's company use DevOps?
- Does the respondent's company use DevOps concepts or environments to manage projects developed through globally distributed work?

Depending on the answer, the interviewee is taken to the specially built branch to approach the subject according to his specificity (*11.1 (Figure 7 – 8)*,*11.2*).

The responses are alternated between grid-type answers, predefined, and those in which the interviewee is asked to answer in writing. Also, where the answer's relevance was high (when grid-type), the interviewee is asked to motivate it. When the survey period was over, the responses were analyzed and interpreted.

The period allotted to conduct the survey, contacting the companies, and collecting the answers was about 12 weeks.

6 Results

6.1 DevOps Benefits That Can Support Distributed Work Environments

Software development environments that use a DevOps approach encourage collaboration, even in distributed work environments, the so-called "working alone together" [1]. Members /entire team may feel like working together with other members, teams, even when working separately. This fact is also facilitated by the organizational culture that promotes transparency and accountability, in parallel relying on a shared vision, purpose, responsibilities [36].

DevOps has a strong focus on team culture, which improves the collaboration and integration of different teams or individuals in distributed work [17,21,11]. DevOps can prevent, to some extent, the problems that may arise from the asymmetries of power, resources, knowledge. Its nature facilitates access to shared information and technologies, efficient data exchange, and even a balanced distribution of decisional power. This can be possible due to the collectively shared competencies that enable the DevOps members to be involved in all the stages of the product's lifecycle [11]. Common competencies facilitate communication, integration of members and encourage mutual respect, reducing the chances of creating situations such as "us and them" or having different teams or individuals feeling marginalized or wronged, misunderstood [31,2]. DevOps solves most of the issues of teams working solely for different purposes and provides a way to combine them and work together.

Due to process automation, DevOps supports a flexible work environment in which errors are identified in real-time (real-time feedback), facilitating a smooth workflow that can avoid conflicts due to delays or non-synchronizations between different teams. The responsibilities are clear, even if the overall responsibility is collective. This is an additional way to create cohesion in distributed teams [37,6,8,2]. The cloud-oriented CI/CD, servers, and VMs provide continuous and equal support for geographically distributed teams or team members [9,2,12].

DevOps uses mechanisms to maintain team culture and actively educate all team members. Critical members of DevOps teams can act as mediators [4] and conciliators or advisers on issues that require mediation in distributed work [1]. The most qualified members can assist in problems and can be easily accessed using the standard platforms used in communication and collaboration within DevOps teams. In this way, DevOps supports the creation of informal, ad-hoc communication flows that support distributed work [2,21].

Collective ownership and shared values, skills, and goals, using the same application process and tools, making DevOps a good option when collaboration and communication have to be enforced. *Table 3* outlines the aspects for which DevOps has tools and principles that support the challenges of virtual work environments.

Table 3. Main GDW Issues covered by DevOps

Main GDW Issue	DevOps Concepts Coverage	DevOps Engineering Practices Coverage
Difficulties in establishing informal streams of communication	✓	✓
Diminished affinity and empathy between colleagues	✓	-
Efficient transfer/control of software products	✓	✓
Psychological discomfort/ Lack of interpersonal connections	✓	-

Different time zones/ Flexible schedules	✓	✓
Identity threats/ 'Us vs Them'/ Lack of cohesion	✓	-
Duplication of work	✓	✓
Inefficiency in Information Sharing	✓	✓
Cultural and personal differences/Stereotyping people	✓	-
Knowledge, power, resources asymmetries	✓	✓

6.2 DevOps Environment

As mentioned in DevOps Environments and Engineering Practices, the role of DevOps is to help development teams with continuous planning, continuous integration, and continuous delivery of the software. The literature findings point out the standard DevOps environment does not exist but is tailored according to the needs and scope of the project. Continuous integration, continuous delivery, continuous deployment, testing, project management, version control, and build/configuration management are the building blocks of DevOps environments conforming to the literature findings discussed in 3.4.3. The keyword of the literature regarding the DevOps environments is automation. As the literature mentions, deployment automation is one of the most adopted practices among DevOps teams.

Table 4. DevOps Commits/ Time Performance

↓Stage / Iteration→	1	2	3	4	5
Build (s)	2	2	2	2	2
Test (s)	1	1	2	1	1
Package (s)	2	2	2	3	2
Deploy (s)	0.55	0.52	0.54	0.54	0.55
Overall time (s)	5.55	5.52	6.54	6.54	5.55

Table 5. Non-DevOps Commits/ Time Performance

↓Stage / Iteration→	1	2	3	4	5
Build (s)	3.2	3.1	3.4	3	3.1
Test (s)	2.9	2.5	2.4	2.4	2.6
Package (s)	2.5	2.9	2.8	2.9	2.9
Deploy (s)	3.3	3.6	3.2	3.1	3.5
Overall time (s)	11.9	12.1	11.8	11.4	12.1

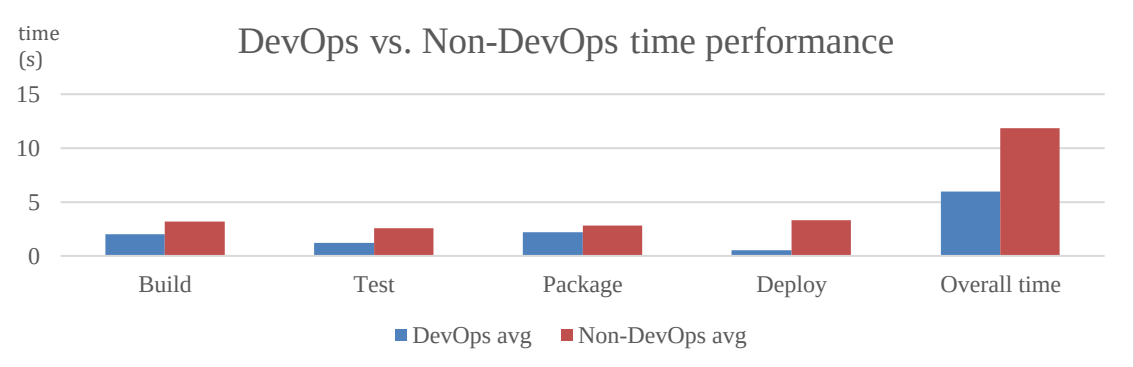


Figure 6. DevOps vs. Non-DevOps average time performance comparison

The proof-of-concept project was realized in two environments in a 100% distributed style of work. The first environment is a DevOps environment discussed in chapter 4. The second environment is a non-DevOps environment containing only the necessary tools for development, excluding the CI/CD tools. It is worth mentioning that the two authors of the thesis executed the proof-of-concept project. This information will be important when analyzing the results and drawing conclusions. During these tests, the team's efficiency was measured in the time needed to complete the same task on both DevOps and non-DevOps environments. The results of tests outlined the importance of automation as the DevOps environment provided faster development execution compared to the non-DevOps environment. The development time ratio compared for the same tasks was close to 2:1 (*non-DevOps: DevOps*), which is almost double the time difference between the aforementioned environments (*Table 4, Table 5,Figure 6*). Figure 6 shows a comparison of the average times presented in tables 5 and 6, which present the results of the experiments performed in DevOps and Non-DevOps environments. The DevOps environment offered automated steps such as building the project, tests, packaging and deploying. In contrast, in a non-DevOps environment, all these steps had to be done manually. During the automated stages, the developer was able to focus on other things. In contrast, in a non-DevOps environment, the developer had to manually trigger each step, wait for it to finish, read the log, check for errors, and then proceed to the next step. Regarding the communication, the DevOps environment offered the notification of the build status posted on Slack which minimized the need for contact and code review.

6.3 Survey

Given the limited time of this project and the subject's broad coverage, the survey was a success. It provided an overview of the local software development environment regarding the topics addressed in this bachelor's thesis. The input obtained is from companies that use DevOps partially or fully for various projects and others that do not use DevOps but consider that the transition is necessary for the future. Others, happy with other project management approaches.

The survey’s questions, together with the answers, are available in the Appendix, in subchapters *11.2* and *11.3*. The survey's result is presented in *Table 6.*

Table 6. Survey’s questions’ indexes

Index	Question
1	Do you consider that geographically distributed work (GDW) is a trend that will be accentuated in the future?
2	Do you believe that companies should consider changes in the organizational and management process to meet the challenges of these new “virtual” work environments?
3	Do you think that the academic environment should prioritize future engineers’ or software developers’ education in the various particularities of globally distributed work?
4	Does your company use services offered by teams, or employees, collaborators located in globally distributed locations?
5	Do you use DevOps concepts or environments to manage software development projects and resources in which these globally distributed entities are involved?
6	Do you use DevOps concepts or environments to manage software development projects and resources in which these globally distributed entities are involved?
7	Do you think that DevOps multifunctional teams can diminish possible miscommunications or difficulties of mutual respect, appreciation, and understanding within GDW environments?

8	Do you think that DevOps supports a flexible work schedule and improves collaboration within GDW teams?
9	Do you think that DevOps environments, by their interdisciplinary nature, diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature, which can occur in GDW environments?
10	Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as 'identity threats' or 'us vs. them' that can occur in GDW environments?
11	Do you think that DevOps environments contribute to a better symmetry of power and knowledge within teams?
12	What is your organization's strategy for managing GDW challenges? (Open question)
13	What kind of software development project management methodology does your company use? (Open question)
14	Do you consider important organizations' access to talent and human resources distributed globally to maintain competitiveness and efficiency?
15	Do you consider it important that people worldwide have access to an open labor market?
16	In the software development project management, what kind of repetitive problems are identified, if any? (Open question)
17	Do you use DevOps environments or concepts?
18	Do you think that DevOps optimizes communication and collaboration within the team?
19	Do you think that DevOps contributes to reducing the possible conflicts generated by the asymmetries of power, knowledge, or resources within these teams?
20	Do you think that DevOps multifunctional teams can diminish the possible lack of mutual respect, appreciation, and understanding within these teams?
21	Do you think that DevOps supports a flexible work schedule?
22	Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as 'identity threats' or 'us vs. them'?
23	What kind of project management methodology do you use? (Open question)
24	How do you manage possible miscommunications or difficulties of mutual respect, appreciation, and understanding within different teams or individuals? (Open question)
25	Do you think that a flexible work schedule improves collaboration within teams?
26	What is your future project management strategy?
27	Did your company consider the implementation or phased transition to DevOps in the future?
28	Is there anything you would like to add about the topics we discussed in this survey?

Table 7. First respondent responses

QI*	Responses
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	• "Shared knowledge, ownership, responsibilities, vision". • "The cohesion of a multifunctional team".
7	Yes. "A common platform gives opportunity and capability but leadership and management must strive to implement workflows and culture".
8	Yes. "A platform that unites and promotes cooperation and consensus".

9	No
10	Yes. “Not alone but combined with leadership”.
11	Yes. “Transparency and trail/audit”.
28	Yes. “The technical platform is the enabler, leadership and knowledge is what determines success or fail.”.

* The question’s index (from*Table 6*).

Table 8. Second respondent responses

QI*	Responses
1	Yes
2	Yes
3	Probably
4	Yes
5	Yes
6	• “Shared knowledge, ownership, responsibilities, vision”. • “Processes automation”. • “The cohesion of a multifunctional team”.
7	Yes. “A multifunctional team allows for more autonomous operations and requires certain flexibility that can only exist when all team members are well understood and work closely together”.
8	Yes. “DevOps promotes a flexible work schedule through automation and elimination of gatekeeping.”.
9	Yes. “As previously stated the flexibility demands a team that works closely together. As such, stereotypes will quickly be broken when the team starts communicating.”.
10	Yes. “DevOps specifically eliminates the old school tollgate approach thus by it's very nature with shared responsibility eliminates the "us vs them"”.
11	Yes. “Since everyone needs an understanding of each other's ways of working it does make every member more knowledgeable”.
28	No.

* The question’s index (from *Table 6*).

Table 9. Third respondent responses

QI*	Responses
1	I don’t know
13	Agile Kaban
14	No. “It depends on the goals that the organisation has. For example we are focusing on talents in Skåne and using that as our competitive edge.”.
15	Yes. “Equal opportunities.”.
16	• “Aligning teams on feature selection”.
17	Yes.
18	Yes. “By creating multi disciplinary teams that can predict future use cases”.
19	Yes. “By letting everyone involved raise their opinions and concerns early on in the development process”.
20	Yes. “Because working close to persons with a different skill set than one's own, uncovers areas of uncertainties and makes the teams members complement and appreciate each other”.
21	Yes. “Everyone can decide when they want to do work on their respective tasks.”.
22	Yes. “Because you get to work close to people that are different from yourself.”.
24	• “We hold 15 min standups every morning to get on the same page within the team. I can't say it improves difficulties with mutual respect though.”.
25	Yes. “I think the flexibility can be important for individual performance soas long as team convergence works properly I think it's a nice feature”.
26	• “We will continue with Agile Kanban”.
28	No

* The question’s index (from *Table 6*).

Table 10. Fourth respondent responses

QI*	Responses
1	Yes
2	Yes
3	Probably
4	Yes.
5	No.
12	• "Offering a tool to use locally, in the future we will look into offering it as part of cloud-based DevOps.".
23	"Agile, development in 2-week sprints".
24	• "Depending on the implementation and organizational culture". • "By focusing on our corporate culture, independent of who you are". • "Emphasize collaboration for mutual success". • "Making sure tasks are feasible for the assignees, give them the resources needed to complete the task".
26	Yes. "Certain work-hours are mandatory, others are distributed freely for each individual.".
27	Yes. "We are a start-up in an early stage, this will have to be set in a later stage.".
28	No

* The question's index (from *Table 6*).

Table 11. Fifth respondent responses

QI*	Responses
1	Yes
2	Yes
3	Yes
4	Yes
5	Yes
6	• "Processes automation". • "The cohesion of a multifunctional team"
7	Yes. "Because we have to do it".
8	Yes. "Use to it already".
9	Yes.
10	Yes.
11	Yes.
28	Yes. "Your topic is important and its already started to be used in a wider scale, but we know to little about it yet. The main problem for us is information security (we are ISO-27001 certified) and the process to govern that".

* The question's index (from *Table 6*).

7 Analysis

7.1 DevOps Concepts and Engineering Practices

DevOps, by its conceptual and flexible nature, can be implemented in different models that can be adjusted to the specific needs given by the particularities of the challenges of varying software development environments. It aims to help developers speed up new software development and maintain the existing products easier by automating specific processes and bridging the gap between the developers and operations teams. This knowledge was gained from the literature review and analyzed by implementing a DevOps environment tested with a proof-of-concept project. This subchapter analyzes the results presented in subchapters 6.1 and 6.2. These concern DevOps concepts and how they can support geographically distributed work and proof-of-concept testing.

The review of the literature on DevOps and geographically distributed work shows most of the problems encountered in GDW DevOps help fix these to a greater or lesser extent, and, when it does not, at least does not amplify them. The results show unequivocally that automation improves collaboration and communication. The extensive competencies support this within the DevOps teams that reduce the chances of problems occurring in geographically distributed work. Better communication and a high ability to understand all product life cycles help make work within DevOps transparent. In addition, the organizational culture focused on inclusiveness and individual and group responsibility seems to be very appropriate in environments that would naturally encounter difficulties in communication or collaboration. The proof-of-concept development and tests compared to the literature review results show that DevOps' principles and practices support a more direct collaboration, involved, assumed, and easy to maintain, through automation and organizational culture. Proof-of-concept demonstrates that the members of a DevOps team need extensive skills. The DevOps environment itself allows for more accessible and more transparent collaboration.

The proof-of-concept project realization tested the environments' efficiency regarding the GDW and software development. The two environments were categorized as DevOps and non-DevOps, meaning that the DevOps environment included the common tools that a typical DevOps environment should include and the organizational culture of the DevOps team. In contrast, the non-DevOps environment used tools needed for the development, and each team member had to take on role responsibilities such as developer, tester, and production manager. As mentioned before in chapter 6.2, the realization of the project is with a team of two persons (the authors of this thesis), so each team member had to take multiple roles. The results of the environment tests confirm the literature findings. They show that the DevOps environment saves time by automating tasks, thus giving the developers time to work on other tasks, which leads to faster development of the product. While the results show that 2:1 ratio comparing the time needed for the execution of tasks such as running tests, building, compiling, packaging, and deploying the artifacts implies that (in this case) DevOps environment takes half the time compared to the non-DevOps environment, the time saved could be more significant when a project includes more than two team members.

In non-DevOps environments, circumstances must be considered. Different time zones, availability, personal interpretations, or divergent interests can add complexity to the processes and extend the necessary times discussed above. This topic will be further discussed in chapter 9 and presented as a part of the conclusion.

7.2 Survey

The companies participating in the survey believe that geographically distributed work is on an upward trend that will increase in the future. Companies should take the requirements of the distributed and virtual work environment seriously, making organizational and management changes to facilitate the efficiency of work in these environments.

The Academic environment should prepare software engineers and developers to integrate into globally distributed work environments.

With one exception, all respondents use globally distributed work. This shows that DevOps is already popular and widely used in software development companies. Three companies currently use DevOps concepts or environments in projects involving globally distributed work.

Three out of five companies felt that DevOps could enhance globally distributed work through *the cohesion of multifunctional teams*. Two out of five companies considered that DevOps could enhance globally distributed work through *process automation*. Two out of five companies thought DevOps could enhance globally distributed work through *shared knowledge, ownership, responsibilities*.

Three of the respondents agreed that *DevOps could diminish miscommunications* and other professional or personal collaboration issues in GDW environments.

Four of the respondents considered that *DevOps supports a flexible work schedule* and improves collaboration within GWD teams. One respondent replied that they are using a flexible work schedule to improve the cooperation in GDW.

Two respondents believe that DevOps, by its *interdisciplinary nature*, can avoid people being stereotyped on geographical or personal grounds. Three respondents believe that DevOps can help prevent "us vs. them" situations in GDW environments.

Four respondents believe that DevOps can contribute to a better symmetry of power, resources, and knowledge within GWD.

One responded suggested that in the future will offer the development tools used now locally into a cloud-based DevOps environment to manage GWD challenges.

One respondent not using DevOps at present is intending to transition to DevOps when the company grows. Another, using DevOps in the present, considers that the *DevOps environment optimizes the collaboration within teams,* supports a flexible work schedule, and *diminishes the asymmetries* of power, knowledge, resources, and possible *conflicts* as "us vs. them" or "identity threats".

The interviewed companies confirmed that DevOps could play a role in creating virtual work environments in globally distributed working conditions and offer improved possibilities for formal and informal communication and collaboration within or between teams. It is appreciated that distributed work will continue to become increasingly important and that organizations must adapt organizationally and culturally to this new reality. One respondent very eloquently stressed that "*the technical platform is the facilitator, leadership and knowledge are what determines success or failure*." In this sense, DevOps brings a combination of organizational culture, management, and technological solutions combined in a framework that improves efficiency for software development, on the site, or in virtual environments.

8 Discussions

8.1 Information Security

One of the difficulties in implementing DevOps environments is the different security standards that organizations choose to implement. One of the survey respondents mentions this aspect, saying, "*Your topic is important, and it's already started to be used on a wider scale, but we know too little about it yet. The main problem for us is information security (we are ISO-27001 certified) and the process to govern that*".

Even if his paper does not seek to analyze this issue, it must be considered in the transition to a DevOps environment. Even though, as the name suggests, DevOps' main goal was to eliminate the friction between the two silos, Development and Operations, DevOps has amplified and tends to use its concepts to create different paradigms that redefine the way software development is done frictionless. There will always be trade-offs when decisions are needed to reconcile divergent interests, such as security and usability requirements. It remains up to the skill and commitment of organizations to find the most inspiring ways to overcome these challenges.

8.2 Dev Ops as an Organizational Culture

Much of DevOps is about automation. A survey respondent states: "*The technical platform is the enabler, leadership and knowledge is what determines success or fail*". This shows that the environment setting in itself, technologies, and automation is not enough to guarantee the efficiency or success of these work environments. The leadership, tenacity, and skills of those who create this environment are essential to its success. Geographically distributed work environments have shortcomings and challenges that require increased management and attention to organizational culture and a constant interest in reducing the risk of conflict, improving overall efficiency, not just the one strictly related to technical performance.

8.3 Social and Ethical Issues Covered by GDW and DevOps

It is worth noting the importance of the organizational cultural environment in supporting distributed work in software development environments. GDW and DevOps participate in streamlining organizations' activities and creating a more equitable ethical and social work environment for all involved. Both contribute to increasing global social equity by giving worldwide workers the chance to access geographically unrestricted jobs. By encouraging collaboration, DevOps supports a positive work environment in which shared responsibilities, vision, and knowledge help to understand better and improve communication, collaboration, and overall performance. These have a direct personal, organizational and social impact.

8.4 DevOps and Context-Specific Performance

The need for different roles in non-DevOps teams is reduced by using DevOps. In GDW, multiple factors affect the teams' performance, one of them being the timezone difference. In our proof-of-concept project, we tested the DevOps environment within the same time zone. We got a time performance of twice as fast when we compared DevOps versus non-DevOps environments. When considering the timezone difference, the time saved can be more than double in DevOps environments as automated tasks are executed by the machine. In contrast, in non-DevOps settings, we would, in some cases, have to wait for hours to get the test results from testers that working from a location that is indifferent timezone. This is just one of many topics left for discussion.

9 Conclusions

This thesis report presents how the managerial, cultural, and engineering concepts and practices DevOps' associated support the key issues facing GDW environments. The literature review is the starting point of this investigation and contributes to the formulation of the first results. It is also the support and benchmark for the realization of the proof-of-concept and the survey. Proof-of-concept results show that the information found in the literature review is supported in practice. Also, the survey participants' answers largely support the literature and bring extra knowledge of the subject approached. The conclusions, related to the two research questions that represent the landmark and purpose of this paper, are the following:

1. *How can DevOps contribute to embracing a distributed style of working?*

DevOps is a paradigm not very strictly defined precisely because of its application beyond the technical or purely managerial sphere. It aims to address a new software development paradigm that seeks to improve overall performance within an organization. The fundamental problems of distributed work are related to possible communication and collaboration deficiencies. These have multiple causes of a technical, managerial, organizational, psychological, cultural, and other nature. DevOps, through its core principles, tends to find solutions to these problems. Some examples in this regard, mentioned in this paper: reducing the impact of time zone differences in the software development process; access to standard tools available online in cloud environments; educating members of multidisciplinary teams; use of human resources in an inclusive manner; focusing on group cohesion through shared responsibilities, knowledge and vision; an organizational culture that allocates resources in streamlining activities, with an emphasis on communication, collaboration, and extended general understanding; and so on. According to these data, DevOps can add value to distributed work environments.

2. *Do companies think that DevOps methods can improve the new pandemic context's distributed work conditions? How? What other options do they have?*

The companies participating in the survey show that they use DevOps or Agile. They believe that DevOps provides principles and models to support distributed work. Also, only automation or management is not enough. A broad approach is needed, where the organizational culture can support the increased trend of distributed work. One company has suggested continuing with Agile Kaban, while the rest are already using DevOps and distributed work or see themselves transitioning to DevOps in the future. In conclusion, DevOps is already successfully used on a large scale, in tandem with distributed work.

9.1 Future Work

DevOps seems like the next option when it comes to software production in small and medium enterprises. However, there are still not enough scientific sources to address this concept in detail. This means that further scientific investigations into this concept are needed and timely. The number of university courses that DevOps addresses are still minimal.

From the literature analysis and the input received from the companies participating in the survey, GDW and DevOps are in an upward trend and a reality of almost all IT companies. It would be interesting to investigate which DevOps tools and models are best suited to be part of the university curriculum to prepare students for the professional environment, the labor market, and the entrepreneurial reality.

10 Bibliography

1. Hemon-Hildgen , Lyonnet B, Rowe F, Fitzgerald B. From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*. 2020; 22(3): p. 927-945.
2. N ST, R DAKA. Automation of Software Development using DevOps and its Benefits. *International Journal of Engineering Research & Technology (IJERT)*. 2020 June; 9(06).
3. McTaggart V, McLaughlin C. Remote Working the New Reality. In *14th EIASM COLLOQUIUM on Organisational Change and Development. Outside in: Organisational Change and Development in a Globalised Economy*; 2020; Brussels - Belgium.
4. Eisenberg J, Mattarelli E. Building Bridges in Global Virtual Teams: The Role of Multicultural Brokers in Overcoming the Negative Effects of Identity Threats on Knowledge Sharing Across Subgroups. *Journal of International Management*. 2017; 23(4): p. 399-411.
5. Bello MJG. Software Engineering in Globally Distributed Teams. In *4th International Conference on Software Engineering*; 2018; Copenhagen, Denmark. p. 2-8.
6. Dr. Ashok KAR, Sushma TN. Automation of Software Development using DevOps and its Benefits. *International Journal of Engineering Research & Technology (IJERT); ISSN: 2278-0181*. 2020 June; Vol. 9(Issue 06).
7. Zaitsev A, Gal U, Tan B. Reviewing the Role of the Agile Manifesto and Agile Methods in Literature. In *The Americas Conference on Information Systems*; 2018; New Orleans.
8. Bobrov E, Bucchiarone A, Capozucca A, Guelfi N, Mazzara M, Naumchev A, et al. *DevOps and Its Philosophy: Education Matters!* In Microservices Science and Engineering. Cham: Springer; 2019. p. 349-350.
9. Wiedemann A, Wiesche M, Gewald H, Krcmar H. Understand how DevOps aligns development and operations: a tripartite model of intra-IT alignment. *European Journal of Information System*. 2020; 29(5): p. 458-473.
10. Banica L, Radulescu M, Rosca D, Hagiu A. Is DevOps another Project Management Methodology? *Informatica Economica*. 2017; 21(3): p. 39-51.
11. Wiedemann , Wiesche , Gewald H, Krcmar H. *Transforming Disciplined IT Functions: Guidelines for DevOps Integration*. In Balancing Agile and Disciplined Engineering and Management Approaches for IT Services and Software Products. Hershey: IGI Global; 2021. p. 293-306.
12. Khan MO, Jumani AK, Mahar F, Siddique WA, Shaikh AA. Fast Delivery, Continuously Build, Testing and Deployment with DevOps Pipeline Techniques on Cloud. *Indian Journal of Science and Technology*. 2020; 13(5): p. 552-575.
13. Amaradri AS, Nutalapati SB. Continuous Integration, Deployment and Testing in DevOps Environment. 2016. Degree thesis.
14. Capizzi A, Mazzara M, Distefano S. *From DevOps to DevDataOps: Data Management in DevOps Processes*. In Bruel JM, editor. Software Engineering Aspects of Continuous Development and New Paradigms of Software Production and Deployment. Cham: Springer International Publishing; 2020. p. 52-62.
15. Angara J, Prasad S, Sridevi G. DevOps Project Management Tools for Sprint Planning, Estimation and Execution Maturity. *Cybernetics and Information Technologies*. 2020 Jun: p. 79-92.

16. Zampetti F, Vassallo C, Panichella S, Canfora G, Gall H, Di Penta M. An empirical characterization of bad practices in continuous integration. *Empirical Software Engineering*. 2020 March; 25(2).
17. Ghantous GB, Gill A. DevOps: Concepts, Practices, Tools, Benefits and Challenges. In *PACIS 2017 Proceedings*; 2017.
18. Waseem M, Liang P, Shahin M. *A Systematic Mapping Study on Microservices Architecture in DevOps*; 2020.
19. Karamistos I, Albrahami S, Apostolopoulos C. Applying DevOps Practices of Continuous Automation for Machine Learning. *Information*. 2020; 11.
20. Skenderi M, Luma-Osmani S. Ethics in DevOps, The Attitude of Programmers Towards it. *Journal of Natural Sciences and Mathematics*. 2020 November; 5(9-10).
21. Fakieh B, AL-Azhrani S. How DevOps Practices Support Digital Transformation. *International Journal of Advanced Trends in Computer Science and Engineering*. 2020 June; 9(3).
22. Watson Fritz B, Narasimhan S, Rhee HS. Adoption of Remote Work Arrangements: An Initial Analysis. In *Proceedings of the 29th Annual Hawaii International Conference on System Sciences*; 1996; Wailea - HI, USA. p. 118-127.
23. Olson MH, Primps SB. Working at Home with Computers: Work and Nonwork Issues. *Journal of the Society for the Psyhological Study of Social Issues*. 1984; 40(3): p. 97-112.
24. Olson MH. Remote office work: changing work patterns in space and time. *Communications of the ACM*. 1983 Mar: p. 183-187.
25. Watson Fritz ME, Higa K, Narasimhan S. Toward a Telework Taxonomy and Test for Suitability: A Synthesis of the Literature. *Group Decision and Negotiation*. 1995 Jul; 4(4): p. 311-334.
26. Begel A, Nagappan N. Global Software Development: Who Does It? In *3rd IEEE International Conference on Global Software Engineering*; 2008; Bangalore, India.
27. Asatiani , Hämäläinen , Pentti. Constructing continuity across the organisational culture boundary in a highly virtual work environment. *Information systems journal*. 2018; 31(1): p. 62-93.
28. Jabangwe , Börstler , Peterson K. Handover of managerial responsibilities in global software development: a case study of source code evolution and quality. *Software Quality Journal*. 2014; 23(4): p. 1-28.
29. Toscano , Zappalà S. Social Isolation and Stress as Predictors of Productivity Perception and Remote Work Satisfaction during the COVID-19 Pandemic: The Role of Concern about the Virus in a Moderated Double Mediation. *Article | WHO COVID | ID: covidwho-945922*. 2020; 12(23).
30. Bano M, Zowghi D, Sarkissian N. Empirical study of communication structures and barriers in geographically distributed teams. *The Institution of Engineering and Technologies Journals*. 2016; 10(5): p. 147-153.
31. Mortensen M, Hinds PJ. Conflict & Shared Identity In Geographically Distributed Teams. *International Journal of Conflict Management*. 2001 Jan; 12(3): p. 212-238.
32. Hinds P, McGarth C. Structures that Work: Social Structure, Work Structure and Coordination Ease in Geographically Distributed Teams. In *Computer Supported Cooperative Work*; 2006; Banff Alberta, Canada.
33. Architectures, Coordination, and Distance: Conway's Law and Beyond. *IEEE Software*. 1999; 16(5): p. 63-70.

34. Brooks JW, Ravishankar MN, Oshri I. Paradox and the negotiation of tensions in globally distributed work. *Journal of Information Technology*. 2020; 35(3): p. 232-250.
35. Fogoros TE, Maftei M, Olaru SM, Bitan GE. From Traditional to Digital: A Study on Business Models in The Context of Digitalization. In *Innovative models to revive the global economy*; 2020; Bucharest.
36. Leite L, Rocha C, Kon F, Milojicic D, Meirelles. A Survey of DevOps Concepts and Challenges. *ACM Journals*. 2019 Nov; 52(6).
37. Hemon A, Lyonnet B, Rowe F, Fitzgerald B. From Agile to DevOps: Smart Skills and Collaborations. *Information Systems Frontiers*. 2020; 22(3): p. 927-945.

11 Appendix

11.1 Survey Design

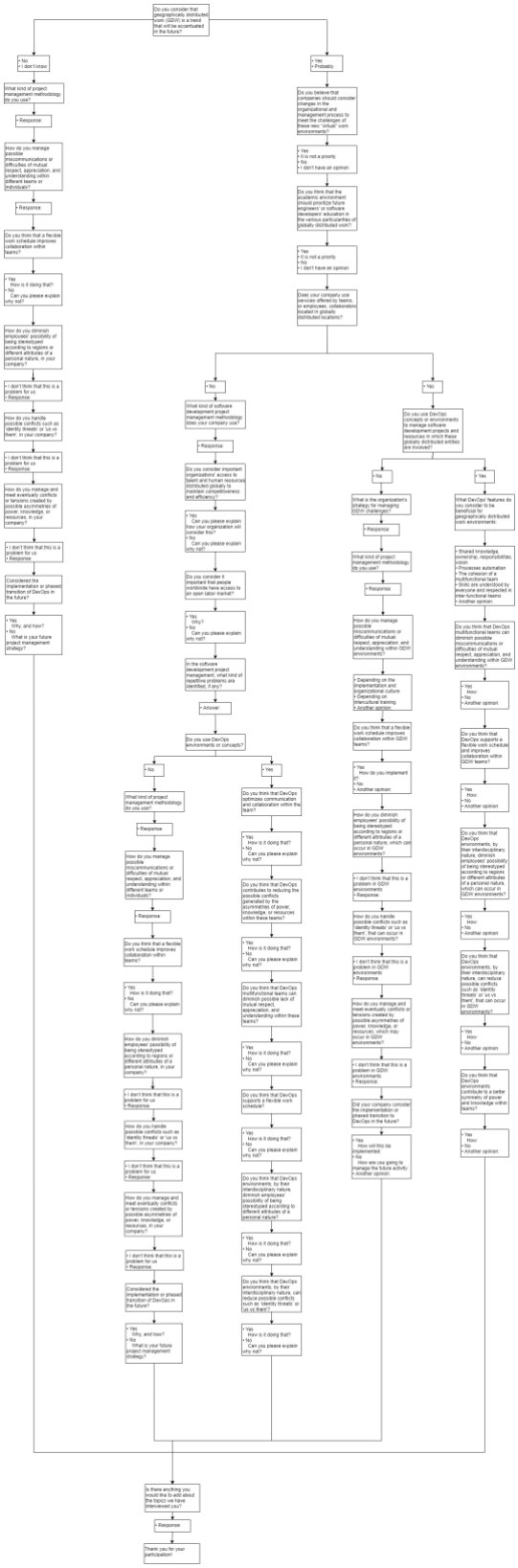


Figure 7. Survey's logical structure

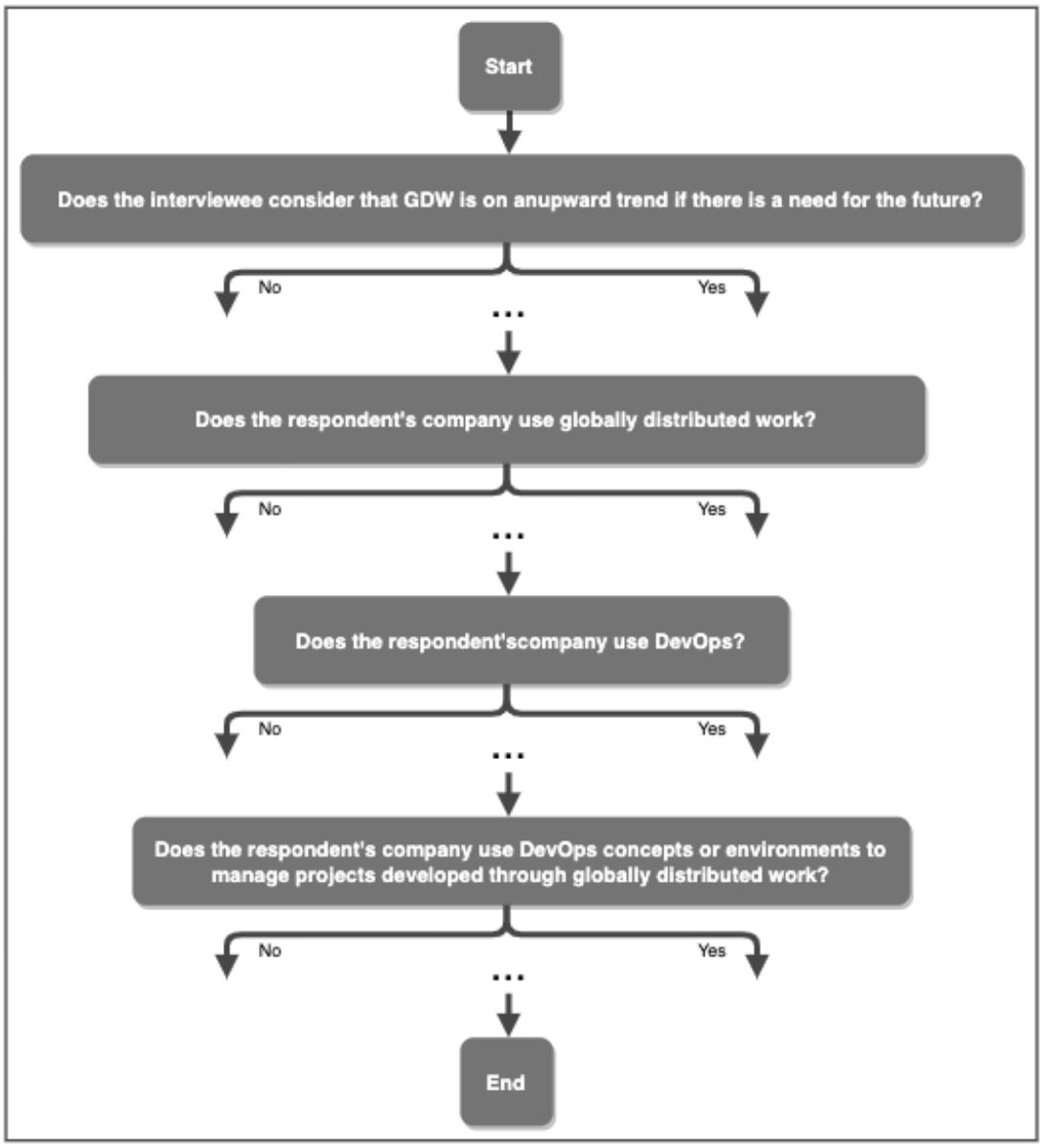


Figure 8. Survey's nodes

11.2 Survey Questions

Title: “DevOps environments' opportunities for better adoption of a globally distributed working culture”.

1. Do you consider that geographically distributed work (GDW) is a trend that will be accentuated in the future?
   • Yes
   • Probably
   • No
   • I don't know

   YES, PROBABLY

   1.1. Do you believe that companies should consider changes in the organizational and management process to meet the challenges of these new "virtual" work environments?
   • Yes
   • It is not a priority
   • No

• I don't have an opinion

1.2. Do you think that the academic environment should prioritize future engineers' or software developers' education in the various particularities of globally distributed work?
- Yes
- Probably
- No
- I don't have an opinion

1.3. Does your company use services offered by teams, or employees, collaborators located in globally distributed locations?
- Yes
- No

YES

1.3.1. Do you use DevOps concepts or environments to manage software development projects and resources in which these globally distributed entities are involved?
- Yes
- No

YES

1.3.1.1. What DevOps' features do you consider to be beneficial for geographically distributed work environments:
- Shared knowledge, ownership, responsibilities, vision
- Processes automation
- The cohesion of a multifunctional team
- Skills are understood by everyone and respected in inter-functional teams
- Another opinion: _____

1.3.1.2. Do you think that DevOps multifunctional teams can diminish possible miscommunications or difficulties of mutual respect, appreciation, and understanding within GDW environments?
- Yes

  How: _____
- No
- Another opinion: _____

1.3.1.3. Do you think that DevOps supports a flexible work schedule and improves collaboration within GDW teams?
- Yes

  How: _____
- No
- Another opinion: _____

1.3.1.4. Do you think that DevOps environments, by their interdisciplinary nature, diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature, which can occur in GDW environments?
- Yes

  How: _____
- No
- Another opinion: _____

1.3.1.5. Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as ‘identity threats’ or 'us vs them', that can occur in GDW environments?
- Yes
  How: _____
- No
- Another opinion: _____

1.3.1.6. Do you think that DevOps environments contribute to a better symmetry of power and knowledge within teams?
- Yes
  How: _____
- No
- Another opinion: _____

NO

1.3.1.7. What is the organization's strategy for managing GDW challenges?
- Response: _____

1.3.1.8. What kind of project management methodology do you use?
- Response: _____

1.3.1.9. How do you manage possible miscommunications or difficulties of mutual respect, appreciation, and understanding within GDW environments?
- Depending on the implementation and organizational culture
- Depending on intercultural training
- Another opinion: _____

1.3.1.10. Do you think that a flexible work schedule improves collaboration within GDW teams?
- Yes

1.3.1.11. How do you implement it?
- No
- Another opinion:

1.3.1.12. How do you diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature, which can occur in GDW environments?
- I don’t think that this is a problem in GDW environments
- Response: _____

1.3.113. How do you handle possible conflicts such as 'identity threats' or 'us vs. them' that can occur in GDW environments?
- I don’t think that this is a problem in GDW environments
- Response: _____

1.3.1.14. How do you manage and meet eventually conflicts or tensions created by possible asymmetries of power, knowledge, or resources, which may occur in GDW environments??
- I don’t think that this is a problem in GDW environments
- Response: _____

1.3.1.15. Considered the implementation or phased transition to DevOps in the future?
- Yes
  How will this be implemented?
- No
  How are you going to manage the future activity?
- Another opinion: _____

NO

1.3.2.1. What kind of software development project management methodology does your company use?

• Response: _____

1.3.2.2. Do you consider important organizations' access to talent and human resources distributed globally to maintain competitiveness and efficiency?

• Yes

Can you please explain how your organization will consider this?

• No

Can you please explain why not?

1.3.2.3. Do you consider it important that people worldwide have access to an open labor market?

• Yes

Why?

• No

Can you please explain why not?

1.3.2.4. In the software development project management, what kind of repetitive problems are identified, if any?

• Response: _____

1.3.2.5. Do you use DevOps environments or concepts?

• Yes

• No

YES

1.3.2.5.1.1. Do you think that DevOps optimizes communication and collaboration within the team?

• Yes

How is it doing that?

• No

Can you please explain why not?

1.3.2.5.1.2. Do you think that DevOps contributes to reducing the possible conflicts generated by the asymmetries of power, knowledge, or resources within these teams?

• Yes

How is it doing that?

• No

Can you please explain why not?

1.3.2.5.1.3. Do you think that DevOps multifunctional teams can diminish the possible lack of mutual respect, appreciation, and understanding within these teams?

• Yes

How is it doing that?

• No

Can you please explain why not?

1.3.2.5.1.4. Do you think that DevOps supports a flexible work schedule?

• Yes

How is it doing that?

• No

Can you please explain why not?

1.3.2.5.1.5. Do you think that DevOps environments, by their interdisciplinary nature, diminish employees' possibility of being stereotyped according to different attributes of a personal nature?

• Yes

How is it doing that?

• No

Can you please explain why not?

1.3.2.5.1.6. Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as 'identity threats' or 'us vs. them'?

• Yes

How is it doing that?

• No

Can you please explain why not?

No, I don't know (1. – I don’t know, No; 1.3.2.5. – No)

2. What kind of project management methodology do you use?
   • Response: _____
3. How do you manage possible miscommunications or difficulties of mutual respect, appreciation, and understanding within different teams or individuals?
   • Response: _____
4. Do you think that a flexible work schedule improves collaboration within teams?
   • Yes

   How is it doing that?

   • No

   Can you please explain why not?
5. How do you diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature in your company?
   • I don’t think that this is a problem for us
   • Response: _____
6. How do you handle possible conflicts such as 'identity threats' or 'us vs. them' in your company?
   • I don’t think that this is a problem for us
   • Response: _____
7. How do you manage and meet eventually conflicts or tensions created by possible asymmetries of power, knowledge, or resources in your company?
   • I don’t think that this is a problem for us
   • Response: _____
8. Considered the implementation or phased transition of DevOps in the future?
   • Yes

   Why, and how?

   • No

   What is your future project management strategy?
9. Is there anything you would like to add about the topics we have interviewed you?
   • Response: _____

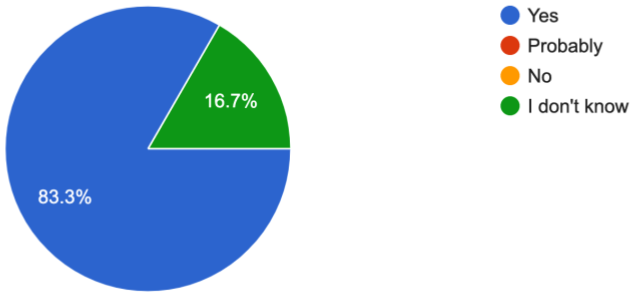
Thank you for your participation!

11.3 Survey Responses

NordAxon
Ericsson
Abstraktor AB
Elastic Move
Xlent analystics Syd AB

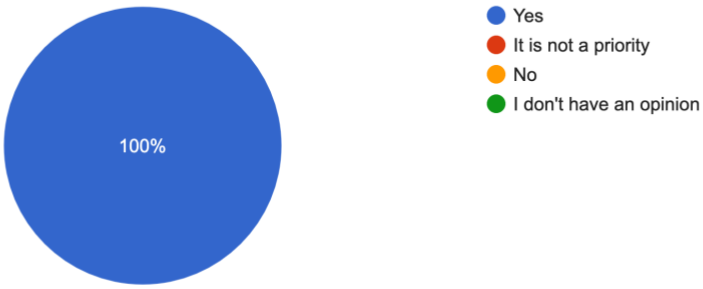
Do you consider that geographically distributed work (GDW) is a trend that will be accentuated in the future?

6 responses



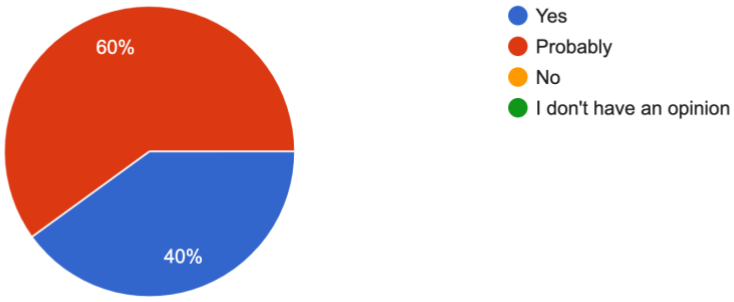
Do you believe that companies should consider changes in the organizational and management process to meet the challenges of these new "virtual" work environments?

5 responses



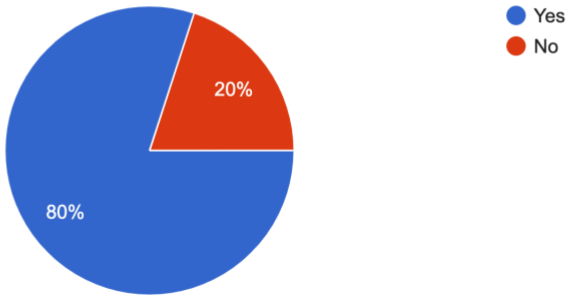
Do you think that the academic environment should prioritize future engineers' or software developers' education in the various particularities of globally distributed work?

5 responses



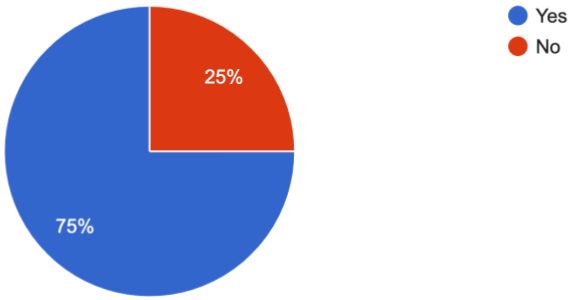
Does your company use services offered by teams, or employees, collaborators located in globally distributed locations?

5 responses



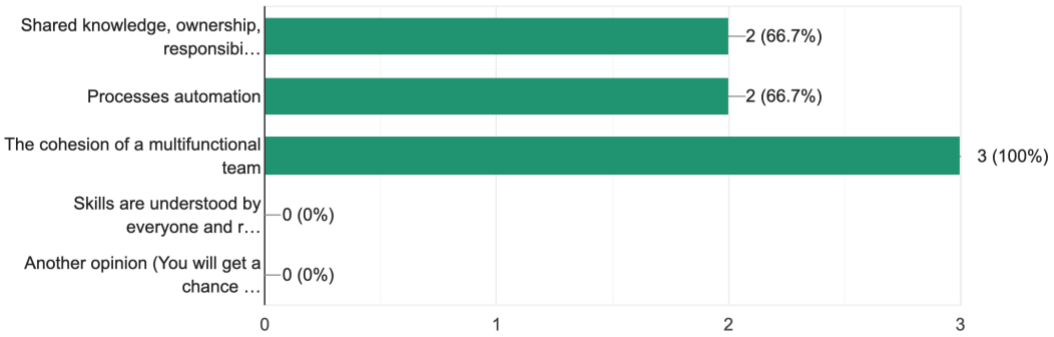
Do you use DevOps concepts or environments to manage software development projects and resources in which these globally distributed entities are involved?

4 responses



What DevOps' features do you consider to be beneficial for geographically distributed work environments? (You can select multiple answers)

3 responses



Do you think that DevOps multifunctional teams can diminish possible miscommunications or difficulties of mutual respect, appreciation, and understanding within GDW environments?

3 responses

How?

3 responses

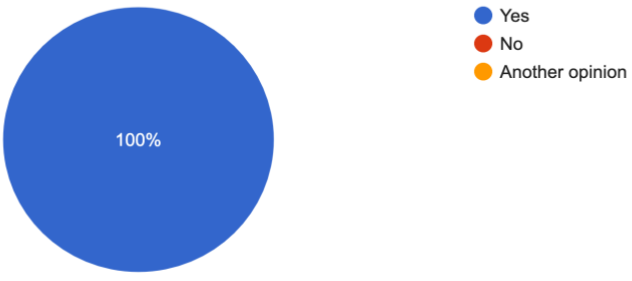
A common platform gives opportunity and capability but leadership and management must strive to implement workflows and culture.

A multifunctional team allows for more autonomous operations and requires certain flexibility that can only exist when all team members are well understood and work closely together

Because we have to do it

Do you think that DevOps supports a flexible work schedule and improves collaboration within GDW teams?

3 responses



How?

3 responses

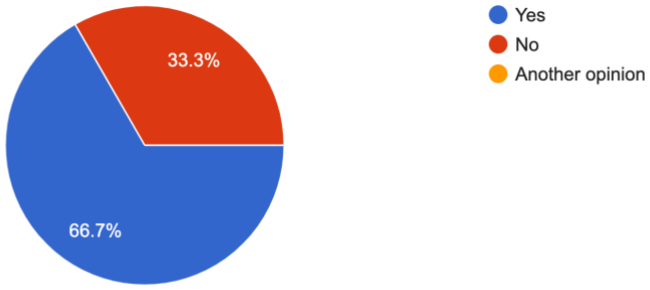
A platform that unites and promotes cooperation and consensus

DevOps promotes a flexible work schedule through automation and elimination of gatekeeping.

Use to it already

Do you think that DevOps environments, by their interdisciplinary nature, diminish employees' possibility of being stereotyped according to reg...nal nature, which can occur in GDW environments?

3 responses



How?

2 responses

As previously stated the flexibility demands a team that works closely together. As such, stereotypes will quickly be broken when the team starts communicating.

-

Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as 'identity threats' or 'us vs them', that can occur in GDW environments?

3 responses

How?

3 responses

Not alone but combined with leadership

DevOps specifically eliminates the old school tollgate approach thus by it's very nature with shared responsibility eliminates the "us vs them"

-

Do you think that DevOps environments contribute to a better symmetry of power and knowledge within teams?

3 responses

How?

3 responses

Transparency and trail/audit

Since everyone needs an understanding of each other's ways of working it does make every member more knowledgeable

-

What is your organization's strategy for managing GDW challenges? (Open question)

1 response

Offering a tool to use locally, in the future we will look into offering it as part of cloud-based DevOps.

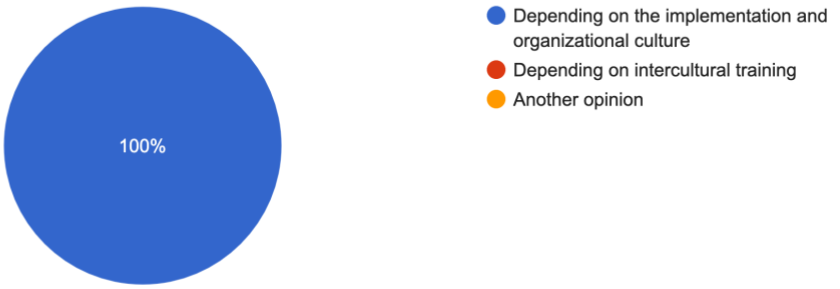
What kind of project management methodology do you use? (Open question)

1 response

Agile, development in 2-week sprints

How do you manage possible miscommunications or difficulties of mutual respect, appreciation, and understanding within GDW environments?

1 response



Do you think that a flexible work schedule improves collaboration within GDW teams?

1 response

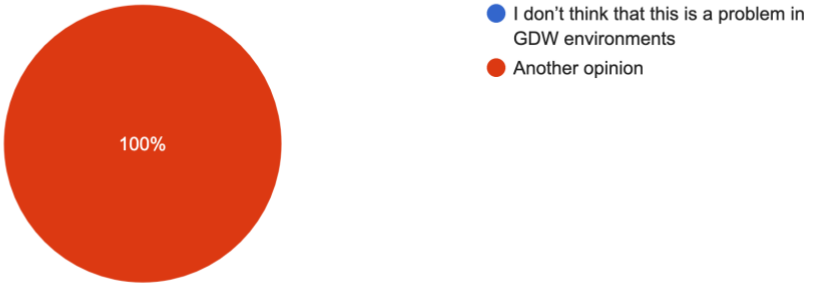
How do you implement it?

1 response

Certain work-hours are mandatory, others are distributed freely for each individual.

How do you diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature, which can occur in GDW environments?

1 response



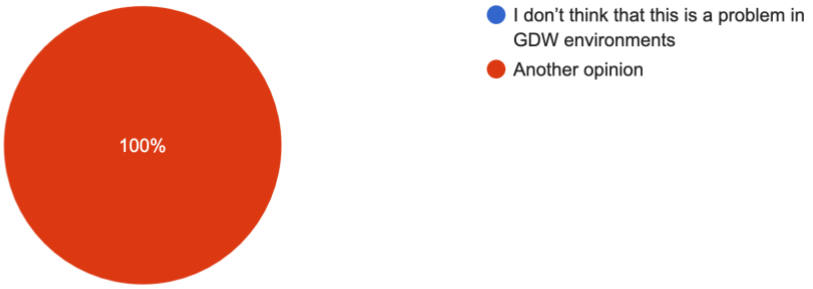
How do you diminish employees' possibility of being stereotyped according to regions or different attributes of a personal nature, which can occur in GDW environments?

1 response

By focusing on our corporate culture, independent of who you are.

How do you handle possible conflicts such as 'identity threats' or 'us vs them', that can occur in GDW environments?

1 response



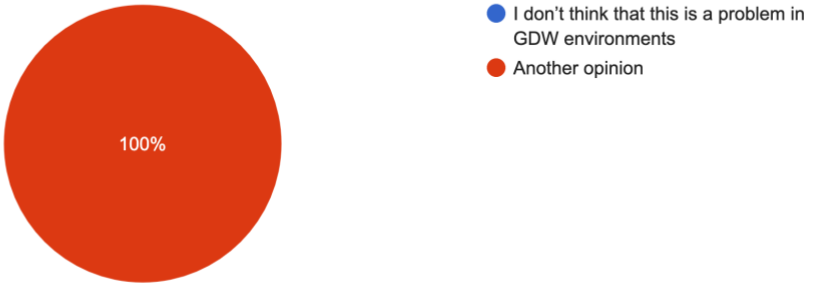
How do you handle possible conflicts such as 'identity threats' or 'us vs them', that can occur in GDW environments?

1 response

Emphasize collaboration for mutual success.

How do you manage and meet eventual conflicts or tensions created by possible asymmetries of power, knowledge, or resources, which may occur in GDW environments?

1 response



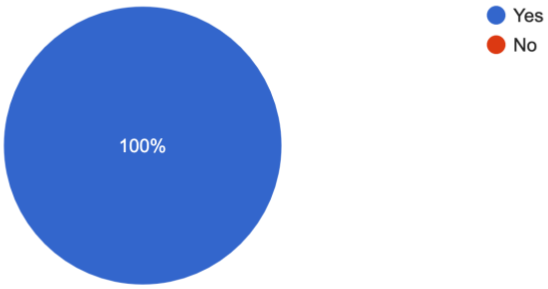
How do you manage and meet eventual conflicts or tensions created by possible asymmetries of power, knowledge, or resources, which may occur in GDW environments?

1 response

Making sure tasks are feasible for the assignees, give them the resources needed to complete the task.

Did your company consider the implementation or phased transition to DevOps in the future?

1 response



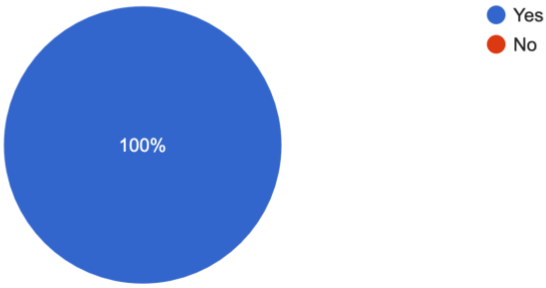
How will this be implemented?

1 response

We are a start-up in an early stage, this will have to be set in a later stage.

Did your company consider the implementation or phased transition to DevOps in the future?

1 response



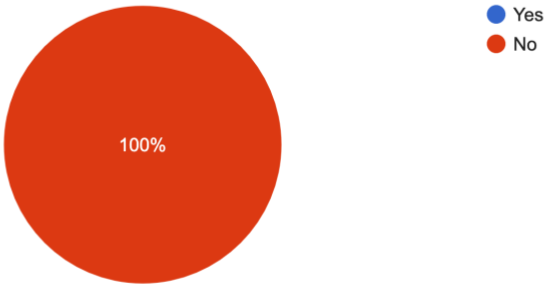
What kind of software development project management methodology does your company use? (Open question)

1 response

Kanban (agile work)

Do you consider important organizations' access to talent and human resources distributed globally to maintain competitiveness and efficiency?

1 response



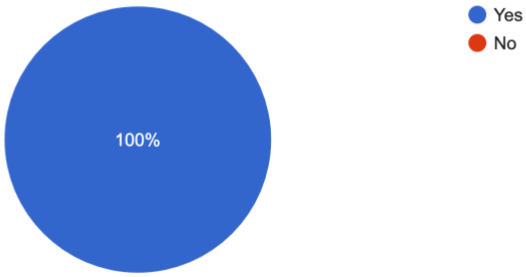
Can you please explain why not?

1 response

It depends on the goals that the organisation has. For example we are focusing on talents in Skåne and using that as our competitive edge.

Do you consider it important that people worldwide have access to an open labor market?

1 response



Why?

1 response

Equal opportunities.

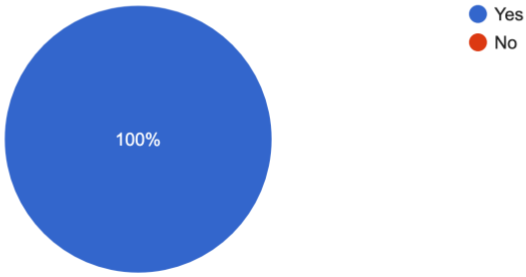
In the software development project management, what kind of repetitive problems are identified, if any? (Open question)

1 response

Aligning teams on feature selection.

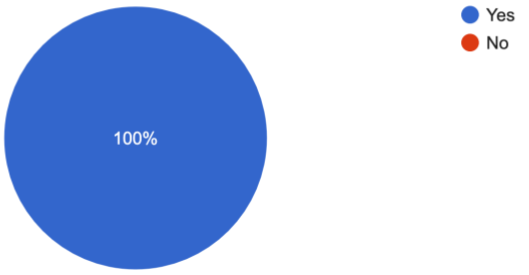
Do you use DevOps environments or concepts?

1 response



Do you think that DevOps optimizes communication and collaboration within the team?

1 response



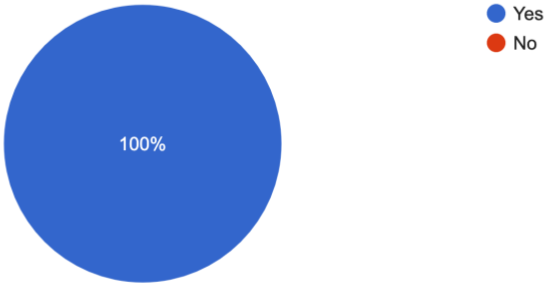
How is it doing that?

1 response

By creating multi disciplinary teams that can predict future use cases.

Do you think that DevOps contributes to reducing the possible conflicts generated by the asymmetries of power, knowledge, or resources within these teams?

1 response



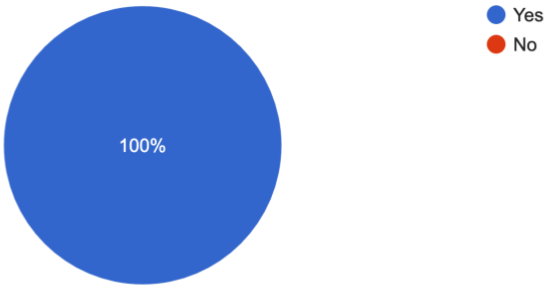
How is it doing that?

1 response

By letting everyone involved raise their opinions and concerns early on in the development process.

Do you think that DevOps multifunctional teams can diminish possible lack of mutual respect, appreciation, and understanding within these teams?

1 response



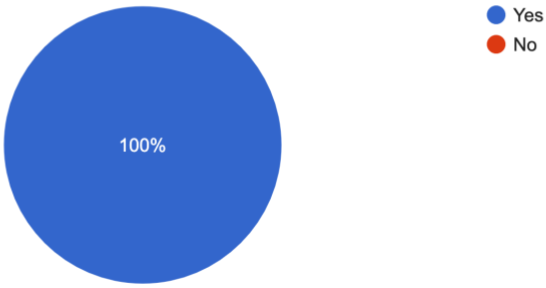
How is it doing that?

1 response

Because working close to persons with a different skill set than one's own, uncovers areas of uncertainties and makes the teams members complement and appreciate each other.

Do you think that DevOps supports a flexible work schedule?

1 response



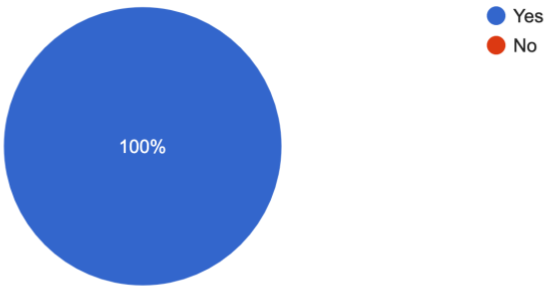
How is it doing that?

1 response

Everyone can decide when they want to do work on their respective tasks.

Do you think that DevOps environments, by their interdisciplinary nature, can reduce possible conflicts such as 'identity threats' or 'us vs them'?

1 response



How is it doing that?

1 response

Because you get to work close to people that are different from yourself.

How do you manage possible miscommunications or difficulties of mutual respect, appreciation, and understanding within different teams or individuals? (Open question)

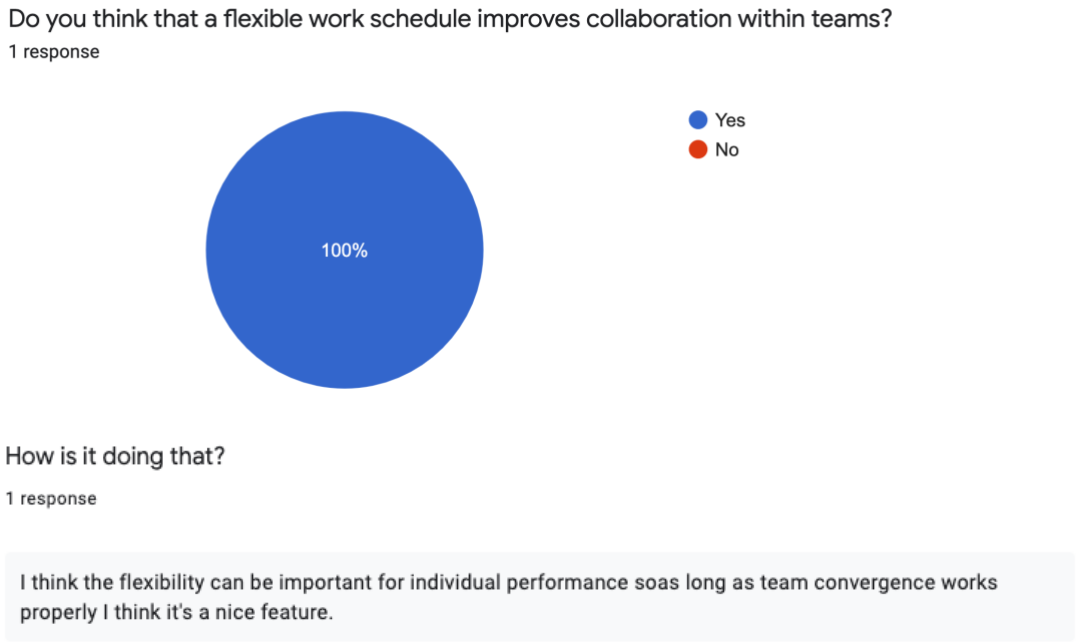
1 response

We hold 15 min standups every morning to get on the same page within the team. I can't say it improves difficulties with mutual respect though.

What kind of project management methodology do you use? (Open question)

1 response

Agile Kanban



11.4 Jenkinsfile

```
pipeline {
    agent any
    tools {
        maven 'Maven'
    }

    stages {
        stage("post-link"){
            steps {
                echo 'posting the link to slack...'
            }
            post {
                always {
                    slackSend (color: '#FFFFFF', message:
"(${env.BUILD_URL})")
                }
            }
        }

        stage("build"){
            steps {
                echo 'building the app...'
                sh 'mvn clean install'
            }
            post {

                failure{
                    slackSend (color: '#FF0000', message: "|Build___|
${env.JOB_NAME}[${env.BUILD_NUMBER}] FAILED")
                }
                success{
                    slackSend (color: '#008000', message: "|Build___|
${env.JOB_NAME}[${env.BUILD_NUMBER}] SUCCEEDED")
                }
                unstable{
```

```
                    slackSend (color: '#FFFF00', message: "|Build___|
${env.JOB_NAME}[${env.BUILD_NUMBER}] UNSTABLE")
                }
            }
        }

        stage("test"){
            steps {
                echo 'testing the app...'
                sh 'mvn test'
            }
            post {

                failure{

                    slackSend (color: '#FF0000', message: "|Test____|
${env.JOB_NAME}[${env.BUILD_NUMBER}] FAILED")
                }
                success{
                    slackSend (color: '#008000', message: "|Test____|
${env.JOB_NAME}[${env.BUILD_NUMBER}] SUCCEEDED")
                }
                unstable{
                    slackSend (color: '#FFFF00', message: "|Test___|
${env.JOB_NAME}[${env.BUILD_NUMBER}] UNSTABLE")
                }
            }
        }

        stage("package"){
            steps {
                echo 'packaging the app...'
                sh 'mvn package'
            }
            post {

                failure{
                    slackSend (color: '#FF0000', message: "|Package|
${env.JOB_NAME}[${env.BUILD_NUMBER}] FAILED")
                }
                success{
                    slackSend (color: '#008000', message: "|Package|
${env.JOB_NAME}[${env.BUILD_NUMBER}] SUCCEEDED")
                }
                unstable{
                    slackSend (color: '#FFFF00', message: "|Package|
${env.JOB_NAME}[${env.BUILD_NUMBER}] UNSTABLE")
                }
            }
        }

        stage("deploy"){
            steps {
                echo 'deploying the app...'
                deploy adapters: [tomcat9(url: 'http://dev-
jenkins.duckdns.org:8082',
                                                credentialsId: 'deploy user')],
                                        war: '**/*.war',
                                        contextPath: 'war-build'
                script {
                    if (env.BRANCH_NAME == 'main') {
```

```
                        deploy adapters: [tomcat9(url: 'http://dev-
jenkins.duckdns.org:8082',

credentialsId: 'deploy user')],
                                                                war: '**/*.war',
                                                                contextPath:
'war-build'
                    } else if (env.BRANCH_NAME == 'develop') {
                        slackSend (color: '#008000', message: "All is good!
You can create pull request now!")
                    }
                }
            }
            post {

                failure{
                    script {
                        if (env.BRANCH_NAME == 'main') {
                            slackSend (color: '#FF0000', message: "|Deploy_|
${env.JOB_NAME}[${env.BUILD_NUMBER}] FAILED")
                        }
                    }
                }
                success{
                    script {
                        if (env.BRANCH_NAME == 'main') {
                            slackSend (color: '#008000', message: "|Deploy_|
${env.JOB_NAME}[${env.BUILD_NUMBER}] SUCCEEDED \nhttp://dev-
jenkins.duckdns.org:8082/war-build/")
                        }
                    }

                }
                unstable{
                    script {
                        if (env.BRANCH_NAME == 'main') {
                            slackSend (color: '#FFFF00', message: "|Deploy_|
${env.JOB_NAME}[${env.BUILD_NUMBER}] UNSTABLE")
                        }
                    }
                }
            }
        }
    }
}
```

11.5 Slack notification

(http://dev-jenkins.duckdns.org:8081/job/thesis-proof/job/develop/10/)

|Build___| thesis-proof/develop[10] SUCCEEDED

|Test____| thesis-proof/develop[10] SUCCEEDED

|Package| thesis-proof/develop[10] SUCCEEDED

All is good! You can create pull request now!

(http://dev-jenkins.duckdns.org:8081/job/thesis-proof/job/main/1/)

|Build___| thesis-proof/main[1] SUCCEEDED

|Test____| thesis-proof/main[1] SUCCEEDED

|Package| thesis-proof/main[1] SUCCEEDED

|Deploy_| thesis-proof/main[1] SUCCEEDED
http://dev-jenkins.duckdns.org:8082/war-build/