

A Multivocal Literature Review On The Correlations Between Devops And Agile, Lean, And Continuous Deployment

Author^{#1}

[#]My Lab, My University, USA

^{\$}Gollum World, New Zealand

^{*}My Research, U.S

[^]SolarSolar, U.S

¹Email

Abstract— *The main purpose of this study was to analyze the correlations between DevOps and agile, lean, and continuous deployment. This expansive literature review examines the intricate interplay between DevOps and three popular methodologies: Agile, Lean, and Continuous Deployment. In today's fast-paced software world, businesses are seeking swift development, flawless integration, and efficient deployment practices. These demands encourage organizations to explore synergies among these methodologies [1]. Through a thorough analysis of scholarly articles and papers from various sources, this review amalgamates different viewpoints to lay bare the subtle connections, convergences, and distinctions between DevOps and the methodologies. The exploration traverses the historical progression, fundamental principles, and practical implementations of each method while delving into their potential to enhance teamwork efforts, streamline operations, and hasten software deliveries [1]. By merging numerous voices found within this body of work, this investigation aims to offer a comprehensive comprehension concerning how DevOps harmonizes with Agile Lean as well as Continuous Deployment offering valuable perspectives for industry professionals, academics plus firms aiming to optimize software development cycles.*

Keywords— *DevOps, continuous deployment, agile, lean, automation, Continuous Delivery, Continuous Integration*

I. INTRODUCTION

In the fast-paced world of software development, effectively managing processes and methodologies is vital. Some methodologies that have gained prominence include DevOps, Agile, Lean, and Continuous Deployment. Each methodology champions unique principles and practices [2]. DevOps fosters collaboration between development and operations teams for smooth and continuous software delivery. Agile advocates iterative development with a focus on customers. Lean aims to minimize waste while maximizing value. Continuous Deployment supports frequent automated release cycles. Independently, these approaches bring benefits, but discussions continue about their relationship and potential integration [2].

This comprehensive review explores the intricate connections between DevOps and Agile, Lean, and Continuous Deployment. Drawing from diverse scholarly pieces, this review seeks to extract collective perspectives surrounding these methodologies. By juxtaposing varied viewpoints,

research discoveries, and practical experiences it engenders a deeper understanding of how these approaches align or collide with each other.

Throughout this review, we embark on an insightful journey into the evolutionary trajectory of these methodologies, retracing their roots, growth, and integration across various organizational settings. The fundamental principles anchoring each methodology are scrutinized closely, unveiling the intrinsic benchmarks that drive their successful application. Furthermore, this review takes a deep dive into the functional consequences arising out of amalgamating DevOps with Agile, Lean, and Continuous Deployment practices. This entails studying real-life case scenarios, instances of triumphs faced against odds, and vital takeaways gathered from real-world implementation endeavours[2].

Amidst the shifting landscape where the software industry now favors an all-encompassing and harmonized approach to development and deployment ideology; This review contributes towards ongoing conversations by pooling multifaceted viewpoints together. By curating a blend of opinions distilled from existing literature; this review throws light on pathways organizations can potentially explore if they aspire to align their developmental strategies for optimal efficiency. Subsequent segments of this review will delve deeper into distinct qualities harboured by each methodology; points where these interfaces with one another; and implications birthed from stitching together a coherent structure drawing upon strengths exhibited by all three paradigms[2,3].

II. RESEARCH PROBLEM

The main problem that will be addressed by this study is to assess the correlation between DevOps, lean, agile, and continuous deployment. The absence of a consistent, systematic process for managing IT services exposes IT organizations to the risk of compromising the entire business through ineffective, fragmented data systems, sluggish processes, and inadequate capacity to support business objectives. These days, businesses of all sectors are transforming into software companies [4]. The pressure is intense, and firms are striving to be as competitive as possible to keep pace with it. Continuous delivery[4,5] serves as a key element in achieving success as a software company. Not only does traditional DevOps pose a major challenge for small firms competing for skilled individuals but these very startups also have the most promising potential to incorporate DevOps scaling into their organizational structures and cultures. In doing so, DevOps can be optimized by employing a lattice-style team philosophy

where both generalists and specialists are tapped into and nurtured. Technology and software development represent key factors in delivering value to customers and stakeholders since they contribute towards enhancing process velocity [6]. Nevertheless, they are unable to deliver value on their own - more is needed [5]. DevOps presents a highly effective approach for improving value delivery through software development.

III. LITERATURE REVIEW

A. DevOps definition

DevOps, a portmanteau of Development and Operations, represents an all-encompassing and radical method of software development along with IT operations that aspires to bridge the gap between these historically distinctive areas of expertise. It is centered around nurturing cooperation, communication, and integration amongst development groups (entrusted with constructing software programs) as well as operations groups (entrusted with deploying and sustaining these programs). By breaking down barriers and stimulating cross-functional collaboration, DevOps strives to streamline the complete software development life cycle extending from original coding to production rollout along with continuous maintenance [7].

In practice, DevOps prioritizes automation, constant integration, and constant delivery (CI/CD), along with persistent monitoring. Automation tools are utilized to reduce manual labor along with error-prone processes, thus empowering venture teams towards reliably delivering software updates. As an example, let us consider a scenario wherein a firm is in the process of creating a web-based application. Through deploying DevOps methodologies, both the development and operations teams work together on defining infrastructure via code while automating server provisioning alongside implementing automatic deployment pipelines[8,9]. This ensures code alterations undergo thorough testing, integration procedures occur culminating in prompt deployment across production environments in an efficient consistent fashion.

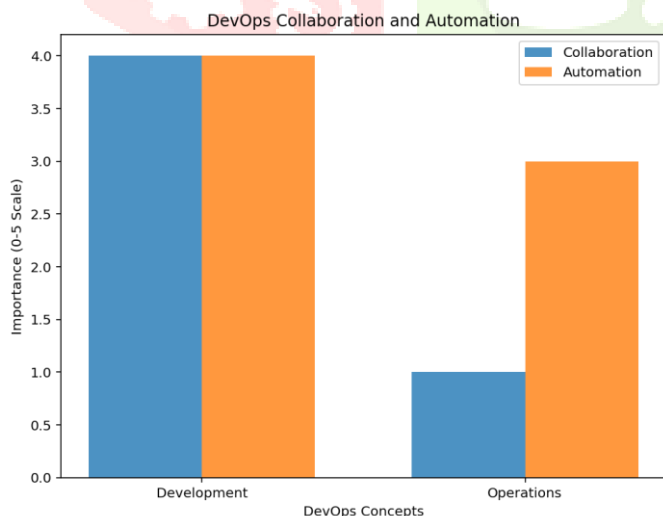


Fig i: DevOps collaboration and automation

A notable example of successful DevOps adoption can be seen in Amazon. The enterprise transitioned from traditional cycles of release to practices associated with DevOps, which led to a reduction in the time taken to bring new features and updates to market. This level of agility facilitated rapid responses to market requirements, such as the introduction of novel services via Amazon Web Services (AWS) [10]. The experience at Amazon highlights how DevOps can amplify

cooperation, elevate software standards, and stimulate innovation, ultimately culminating in heightened customer contentment and business results. Essentially, DevOps acts as a cultural transformation that empowers organizations to deliver top-notch software at an expedited rate by promoting collaboration, systemizing processes, and prioritizing continual enhancement.

B. Agile definition

Agile, a supple and iterative approach to project management and software development that emphasizes teamwork, adjustability, and customer focus, is distinct from traditional, linear project management methodologies by encouraging incremental progress and ongoing feedback throughout the software development lifecycle [11]. In practical terms, agile methodologies such as Scrum, Kanban, and Extreme Programming (XP) involve breaking projects into smaller segments called iterations or sprints. Typically lasting a few weeks each iteration culminates in an incrementally deliverable product. Agile teams conduct regular meetings to assess progress, plan upcoming tasks, and address any hindrances [11]. Customer feedback is gathered at the end of each iteration facilitating adjustments and fine-tuning based on shifting requirements or priorities. This iterative procedure nourishes responsiveness facilitating adaptability enabling teams to deliver added worth more frequently while incorporating changes seamlessly.

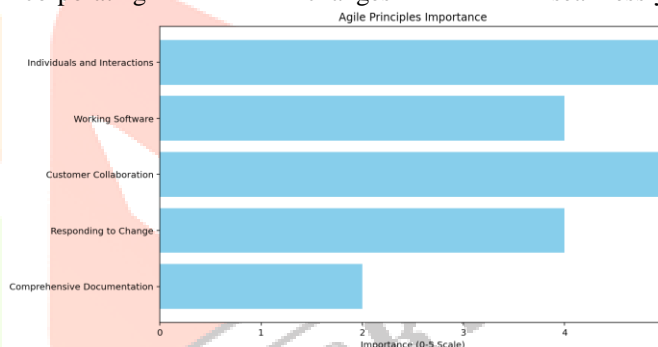


Fig ii: Agile principles

For instance, a software development endeavor focused on constructing a mobile application. An agile approach could involve dissecting the project into smaller functions or user anecdotes, each of which would be built, tested, and potentially given to end users promptly, frequently within weeks. This way, the development team can obtain feedback early and frequently, modifying the product as required based on user reactions and shifting market circumstances [11]. The repetitive characteristic of agile approaches guarantees that the final product closely aligns with user demands and furnishes value throughout its evolution—instead of postponing until a lengthy development circle concludes to reveal possibly outdated or incongruous resolutions.

C. DevOps vs. agile

DevOps and Agile are two complementary approaches that have greatly impacted the field of software development and IT operations. While they share some principles and objectives, they each focus on aspects of the development lifecycle and tackle unique challenges. Agile methodologies such as Scrum and Kanban prioritize development, collaboration, and adaptability to changing requirements. Their main goal is to deliver increments of a product regularly. For example, if an Agile team is building an e-commerce website they might start by implementing core shopping cart functionality before adding features like user authentication and payment processing [12].

The iterative nature of Agile allows teams to gather user feedback and make adjustments accordingly.

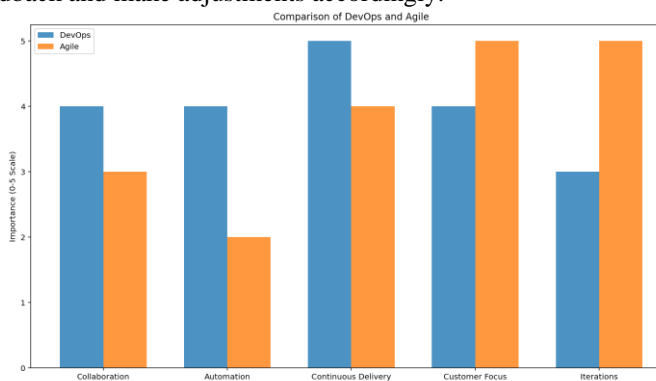


Fig iii: Comparison of DevOps and Agile in a graph

On the other hand, DevOps revolves around improving collaboration between development and operations teams to streamline software deployment and maintenance processes. It emphasizes automation, continuous integration, and continuous delivery to ensure that software changes can be efficiently deployed from development to production environments. In a DevOps approach to software deployment tasks such as server provisioning can be automated while implementing automated testing pipelines helps reduce errors during the release process [12].

1. How do Agile and DevOps work together?

Experts of both Agile and DevOps have similar mindsets. Not only that, their aims align too [12]. When it comes to DevOps, its main goal is to identify ways to streamline existing or new workflows. It focuses on maximizing efficiency and increasing automation. Improving stability, consistency, and planning efficiency is also at the forefront of DevOps [13]. It brings together cross-functional teams who collaborate on software development and delivery projects [9]. Agile however has more to do with pinpointing and crafting features that meet user expectations. This process is specific to dev teams aiming to boost productivity.

The grand objective of a DevOps transition lies in automating the deployment of fully functional code into production [13]. This signifies utter DevOps awareness [13,14]. That said, the Agile manifesto clearly states its utmost priority as the continuous delivery of software for clients. Both Agile and DevOps hold a shared belief that optimal software construction, transparency assurance, and sustainable development can be achieved when working software gets delivered directly into clients' hands. Furthermore, Agile and DevOps encourage the use of modern tools along with cutting-edge processes for rapid software distribution [14].

D. Lean IT

The core concept of Lean IT revolves around maximizing value for the customer while minimizing wastage. In essence, lean implies generating more worth for clients using fewer resources [14]. An organization that adopts a lean approach possesses a deep understanding of unique customer value. The primary focus lies in optimizing core processes to perpetually enhance said value. Ultimately, the end goal is to deliver perfect value to customers through an immaculate creation process that boasts zero waste. Instead concentrating on product and service flow across entire value streams traversing technologies, assets, and departments horizontally to customers. By eradicating waste throughout entire value streams – as opposed to isolating

specific points – processes become streamlined requiring less human input, less space, less capital outflow, and less time investment when manufacturing products or delivering services at significantly reduced costs and encountering fewer defects when compared with traditional business systems[14]. Such companies are well equipped to respond swiftly and effectively to changing customer preferences offering a wide variety of high-quality products/services at low prices and with exceptionally fast turnaround times. Moreover, information management becomes increasingly simplified leading to enhanced accuracy levels.

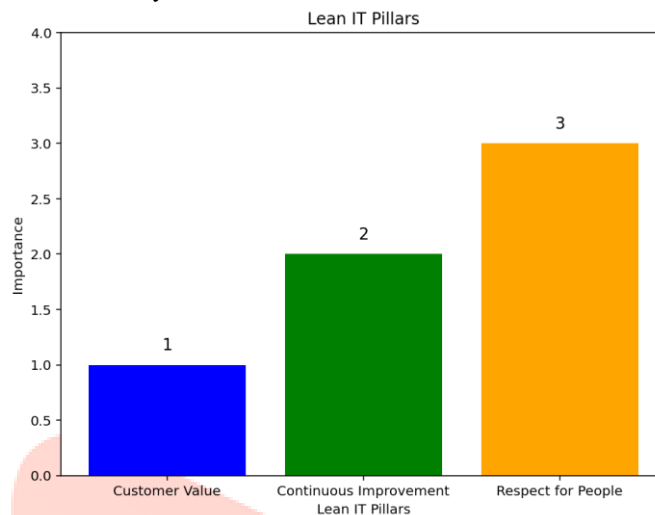


Fig iv: Lean IT pillars

E. Lean IT and DevOps

DevOps represents the merging of people, operations, and tools to foster flexible IT adjustments and business responsiveness. It functions as a means for businesses to regain faith in their IT sections. Both development teams and operations teams need to figure out methods to decrease inefficiency while streamlining procedures[14]. At this juncture, Lean IT enters the picture. It is an extension of Lean Principles, utilized in an Information Technology setting [14]. The approach forms a mindset and course of action with significant emphasis on the organizational culture. Linked with developing and managing Information Technology services and products, Lean IT bears paramount significance [14]. Understanding that DevOps isn't isolated is vital[14,15]. This methodology complements Agile development, which focuses on continuous enhancement along with Lean management that urges companies to build environments fostering constant improvement. DevOps adopts analogous iterative patterns like persistent creation, and continuous delivery as well as unceasing integration concepts employed in Lean and Agile methodologies towards realizing constant betterment during production phases. However, it goes further by utilizing those same principles for shortening the time taken to introduce fresh solutions into the market while reducing the time taken for customers to achieve value[15].

1. How do DevOps and Lean IT work together?

DevOps and Lean IT principles are like two peas in a pod. Where DevOps underscores the importance of working together with development and operations teams, Lean methodology helps streamline operational processes to deliver value faster to customers while aligning with business objectives [15,16]. As a result, this synergy results in more efficient developer teams, smoother daily operations, and ultimately more satisfied customers.

F. Continuous deployment

As soon as there is any need for change, software must be updated accordingly to incorporate any desired functionalities. Once coding is completed it gets released whereupon uploading onto the server commences taking care of individual device updates [16]. The process of unbroken deployment is achieved via Configuration Management practices which utilize containerization tools for effecting updates. Continuous development serves as an all-encompassing concept encapsulating several DevOps processes [16]. Understanding continuous development will ensure each of your DevOps processes remains optimized and effective. Continuous Deployment systems can fabricate and initiate software plans with the utmost efficiency, potency, and the highest level of optimization [17]. The most exemplary approach to implementing Continuous Deployment is to examine each aspect individually rather than the entire bundle. This way, after finalizing the software it can be set into motion right away without any glitches. To comprehend its composition, the primary branch is referred to as mainline, default, or trunk depending on which control system is being utilized [17]. For some, the idea of branches may seem exceedingly perplexing; in such instances, branches are smaller and have relatively fewer lines of code making them infinitely simpler to assess. Code branches serve as an exceptional mechanism for managing diminutive development features. However, their usefulness primarily lies in merging with the trunk resembling how tree branches amalgamate. In a continuous Deployment, these branches must be brief and assimilated swiftly. Well-known global players such as Netflix, Google, Facebook and even Tesla make use of the Continuous Deployment procedure thereby accomplishing several releases during a week or a month.

IV. SIGNIFICANCE AND BENEFITS

The interplay between DevOps to Agile, Lean, and Continuous Deployment is profoundly significant within contemporary software development and IT operations. It introduces a comprehensive structure that strengthens effectiveness, cooperation, and constant enhancement throughout the entire lifecycle of development. To begin with, the interplay between DevOps and Agile strategies is crucial for achieving desired outcomes. Agile methodologies emphasize iterative development, customer collaboration, and adaptability. DevOps further extends these principles into the operations phase through the automation of deployment pipelines, continuous integration, and continuous delivery. By aligning these two approaches, incremental releases created by Agile teams can be moved smoothly from the developmental stage to production environments while minimizing manual intervention [17]. This in turn expedites the delivery cycle whilst maintaining regular feedback loops with users which allows for fast-paced iteration based on real-world input.

Lean principles, sourced from manufacturing practices, concentrate on removing inefficiency, enhancing processes, and delivering value effectively. DevOps perfectly complements Lean by automating and streamlining different stages of software development and deployment procedures. Automation minimizes chances for human mistakes, reduces waiting periods, and boosts resource utilization - all key aspects of Lean ideology. Incorporating DevOps methodologies within a Lean framework encourages an environment where teams have the freedom to continuously refine and optimize their workflows for utmost value dissemination.

The relationship between DevOps and Continuous Deployment reinforces dedication towards prompt and reliable software delivery. Continuous Deployment represents an extension of Continuous Delivery wherein code changes are automatically pushed into production environments after clearing automated tests. Notably, DevOps techniques play a pivotal role in enabling this level of automation as well as confidence in deployments. Through leveraging automated testing protocols, and infrastructure modelling via code, alongside diligent monitoring practices [18]. DevOps secures each code change's thorough examination before it reaches end-users. Such synchronization nurtures high levels of trust in deployment mechanics granting efficient execution.

V. FUTURE

The U.S. will experience an ever-changing landscape characterized by technological advancements, environmental sustainability, progress in healthcare, and a metamorphosing work culture. Within this path, the interplay between DevOps and Agile, Lean, and Continuous Deployment strategies is poised to play a fundamental role in reshaping America's approach towards software development, innovation, and effectiveness. As technology continues to redefine sectors, the bond between DevOps and Agile approaches will continue to remain essential [18]. Agile's step-by-step approach to software development will enable U.S. businesses to react swiftly to varying market demands and customer preferences. DevOps practices will integrate seamlessly into Agile workflows guaranteeing that rapid step-by-step improvements are implemented effectively. An American software firm looking to develop a new mobile app could utilize Agile principles to prioritize user-centric features and employ DevOps automation so that these features can reach end-users promptly and consistently. Moreover, the USA's dedication to sustainability aligns with Lean principles which endorse waste reduction as well as drive efficiency. In such a situation, DevOps methods will help achieve this mission by simplifying the deployment process [18]. For instance, companies in the United States that prioritize green initiatives might develop software solutions to keep track of energy consumption. Employing efficient Lean practices to maximize resource usage and integrating DevOps for automated deployment, these enterprises can contribute towards an eco-friendlier future.

VI. CONCLUSION

This paper looked at assessing the correlations between DevOps and agile, lean, and continuous development. The interplay between DevOps on the one hand and Agile, Lean, and Continuous Deployment methodologies on the other will act as a vital spark for endeavoring these aspirations. As the nation strives to retain its dominance in futuristic advancements, Agile's incremental development philosophy coupled with customer-oriented tactics will find themselves well matched with DevOps' smooth automation capabilities that enable swift delivery of state-of-the-art solutions in a steadfast manner. Combining Lean principles with eco-centered initiatives will be rendered easier through DevOps practices fostering optimization of procedures and resources towards a greener future. Additionally, integration of DevOps into Continuous Deployment will fortify healthcare sectors' ability to swiftly respond guaranteeing the prompt arrival of crucial medical software to professionals and patients alike. As Lean and Agile methods continue to serve as the foundation of development approaches, their incorporation into DevOps and Continuous Deployment will become increasingly important.

This seamless joining together won't just reduce inefficiencies, optimize systems, and improve collaboration but will also allow teams to respond rapidly to market demands and offer actual value to customers. Continuous Deployment is a central feature of the DevOps paradigm playing a key role in this transformation. By identifying and resolving code differences and discrepancies early on in the development cycle, Continuous Deployment will enable faster, more dependable software delivery. Its proactive principles align well with Agile's iterative character driving gradual progress and swift iterations. This will equip organizations with the ability to navigate complexities inherent in modern technological landscapes with assurance. With optimized workflows and reduced life cycle times all aspects of creating valuable software developments become notably easier.

REFERENCES

- [1] S. Carrizo, Sorin Cucu, Moisés Domínguez García, S. Modir, and IBM Redbooks, *Using Liberty for DevOps, Continuous Delivery, and Deployment*. IBM Redbooks, 2015.
- [2] S. Carrizo, Sorin Cucu, Moisés Domínguez García, S. Modir, and IBM Redbooks, *Using Liberty for DevOps, Continuous Delivery, and Deployment*. IBM Redbooks, 2015.
- [3] B. Mccune, *Learning AV Foundation : a hands-on guide to mastering the AV Foundation framework*. Upper Saddle River, Nj: Addison-Wesley, 2014.
- [4] Mitesh Soni, *DevOps for web development : achieve the continuous integration and continuous delivery of your web applications with ease*. Birmingham, Uk: Packt Publishing Limited, 2016.
- [5] P. Duvall, *DevOps in the Cloud LiveLessons*. 2012.
- [6] H. Sharp, H. Robinson, and M. Woodman, "Software engineering: community and culture," *IEEE Software*, vol. 17, no. 1, pp. 40–47, 2000, doi: 10.1109/52.819967.
- [7] L. Chen, "Continuous Delivery: Huge Benefits, but Challenges Too," *IEEE Software*, vol. 32, no. 2, pp. 50–54, Mar. 2015, doi: 10.1109/ms.2015.27.
- [8] M. Airaj, "Enable cloud DevOps approach for industry and higher education," *Concurrency and Computation: Practice and Experience*, vol. 29, no. 5, p. e3937, Aug. 2016, doi: 10.1002/cpe.3937.
- [9] G. Kim, P. Debois, J. Willis, J. Humble, and J. Allspaw, *The Devops Handbook How to Create World-class Agility, Reliability, and Security in Technology Organizations*. It Revolution Pr, 2015.
- [10] J. Verona, *Practical DevOps : harness the power of DevOps to boost your skill set and make your IT organization perform better*. Birmingham, UK: Packt Publishing Ltd., Jul. 2016.
- [11] R. J. Rensaa, "Investigating engineers' needs as a part of designing a professional development program for engineers who are to become mathematics teachers," *The Mathematics Enthusiast*, vol. 9, no. 3, pp. 327–346, Jul. 2012, doi: 10.54870/1551-3440.1248.
- [12] R. F. Elmore, "Comment on 'Towards Rigor in Reviews of Multivocal Literatures: Applying the Exploratory Case Study Method,'" *Review of Educational Research*, vol. 61, no. 3, pp. 293–297, Sep. 1991, doi: 10.3102/00346543061003293.
- [13] V. Garousi and M. V. Mäntylä, "When and what to automate in software testing? A multi-vocal literature review," *Information and Software Technology*, vol. 76, pp. 92–117, Aug. 2016, doi: 10.1016/j.infsof.2016.04.015. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0950584916300702>
- [14] V. Braun and V. Clarke, "Using Thematic Analysis in Psychology," *Qualitative Research in Psychology*, vol. 3, no. 2, pp. 77–101, 2006, doi: 10.1191/1478088706qp0630a. [Online]. Available: <https://www.tandfonline.com/doi/abs/10.1191/1478088706qp0630a>
- [15] Y. Yan and J. Lu, "Guest editorial: Bridging the semantic gap in multimedia understanding," *Neurocomputing*, vol. 208, pp. 1–2, Oct. 2016, doi: 10.1016/j.neucom.2016.05.051.
- [16] Y.-P. Fang and E. Zio, "Unsupervised spectral clustering for hierarchical modelling and criticality analysis of complex networks," *Reliability Engineering & System Safety*, vol. 116, pp. 64–74, Aug. 2013, doi: 10.1016/j.res.2013.02.021.
- [17] J. Hope, "Consider program requirements, student needs when developing note-taker program," *Disability Compliance for Higher Education*, vol. 22, no. 6, pp. 2–2, Jan. 2016, doi: 10.1002/dhe.30257.
- [18] A. Balalaie, A. Heydarnoori, and P. Jamshidi, "Microservices Architecture Enables DevOps: Migration to a Cloud-Native Architecture," *IEEE Software*, vol. 33, no. 3, pp. 42–52, May 2016, doi: 10.1109/ms.2016.64.