Int. J. Communications, Network and System Sciences, 2023, 16, 97-104
https://www.scirp.org/journal/ijcns
ISSN Online: 1913-3723
ISSN Print: 1913-3715

Development Methodologies for Network Softwarization: A Comparison of DevOps, NetOps, and Verification

Mehmet Beyaz

R&D, TTG International LTD, Istanbul, Turkey
Email: mehmet.beyaz@ttgint.com

How to cite this paper: Beyaz, M. (2023) Development Methodologies for Network Softwarization: A Comparison of DevOps, NetOps, and Verification. *Int. J. Communications, Network and System Sciences*, **16**, 97-104.
https://doi.org/10.4236/ijcns.2023.165007

Received: April 3, 2023
Accepted: May 28, 2023
Published: May 31, 2023

Abstract

This white paper explores three popular development methodologies for network softwarization: DevOps, NetOps, and Verification. The paper compares and contrasts the strengths and weaknesses of each approach and provides recommendations for organizations looking to adopt network softwarization.

Keywords

Development Methodologies, Network Softwarization, DevOps, NetOps, Verification, Software-Defined Networking, Network Function Virtualization, Automation, Collaboration, Testing, Validation, Network Operations, Network Management

1. Introduction

The widespread adoption of network softwarization has fundamentally transformed the way organizations manage and operate their networks by using software-defined networking (SDN) and network function virtualization (NFV) technologies. Network softwarization has revolutionized the management and operations of networks by automation. However, the adoption of network softwarization poses several challenges, especially in the development and management of software-based networks. To overcome these challenges, various development methodologies have emerged that are better suited to the software-based nature of modern networks.

This paper focuses on three popular development methodologies for network softwarization: DevOps, NetOps, and Verification by revealing the theoretical underpinnings of each approach with a focus on comparing and contrasting these methodologies.

The goal of this paper is to provide organizations with a comprehensive understanding of the available development methodologies for network softwarization, and to help them make informed decisions about which approach is best suited to their needs. By examining the strengths and weaknesses of each approach, organizations can develop a strategic plan for the successful adoption of network softwarization [1] [2] [3] [4].

2. Fundamental Principles of Network Softwarization

The fundamental principles of network softwarization revolve around three development methodologies; DevOps, NetOps, and Verification.

DevOps methodology enables efficient communication between software developers and IT operations teams which can ease the software development process for both parties, leading to significant changes in software quality.

NetOps methodology is crucial for network softwarization as it reduces downtime and outages by integrated automation of network's hardware, software and services which favors network reliability.

Verification methodology emphasizes testing, analyzing and validation of software applications to acquire expected software standards and to eliminate potential performance issues and/or undesirable security breaches. Verification methodology can also be automated to minimize the software code's vulnerabilities and to reduce time and effort of developers to detect such vulnerabilities by initializing manual testing [1] [5] [6].

2.1. Development Operations (DevOps)

DevOps has emerged as a popular approach to software development, which emphasizes the integration of development, testing, and operations. It originated as a response to the limitations of traditional software development approaches, which tended to be siloed and fragmented. With DevOps, development and operations teams work together throughout the software development lifecycle, with a focus on collaboration, communication, and continuous improvement.

In the context of network softwarization, DevOps can be particularly beneficial. The deployment and configuration of network functions can be complex and time-consuming, and the automation of network management tasks can be challenging. DevOps can help organizations address these challenges by providing a framework for collaboration and communication between development and operations teams.

One of the key benefits of DevOps for network softwarization is its ability to streamline the software development process. By breaking down silos and fostering collaboration between teams, DevOps can help to reduce the time-to-market for new network functions and services. This is particularly important in a rapidly changing market, where organizations need to be agile and responsive to stay competitive. With DevOps, organizations can accelerate the development process, ensuring that they can quickly deploy new network functions and services as needed.

In addition to accelerating the development process, DevOps can also help organizations to improve the quality and reliability of their network functions. By integrating testing and quality assurance throughout the software development lifecycle, DevOps can help to identify and address problems more quickly. This can lead to a reduction in downtime and service disruptions, and ultimately, improved customer satisfaction.

Another benefit of DevOps for network softwarization is its ability to promote automation. DevOps emphasizes the use of automation tools and processes to streamline the software development process and reduce manual errors. This can be particularly important in network softwarization, where the automation of network management tasks is a key challenge. With DevOps, organizations can leverage automation tools and processes to improve the reliability and scalability of their network functions [1] [5] [7] [8].

Despite these benefits, there are also some potential challenges associated with DevOps. For example, the adoption of DevOps may require significant cultural and organizational changes, as well as investments in new tools and technologies. Additionally, organizations with strict regulatory or compliance requirements may find it challenging to adopt DevOps, as it can be difficult to ensure compliance with regulatory standards when using agile development methodologies.

In conclusion, DevOps has emerged as a popular approach to software development that emphasizes collaboration, communication, and automation [9]. In the context of network softwarization, DevOps can help organizations streamline the software development process, improve the quality and reliability of their network functions, and promote automation. While there may be some challenges associated with adopting DevOps, the benefits of this approach can be significant for organizations looking to stay competitive in a rapidly changing market.

2.2. Network Operations (NetOps)

NetOps is a development methodology that specifically focuses on network infrastructure operations. Its primary goal is to enable organizations to manage network hardware, software, and services more effectively. NetOps is a crucial element of network softwarization, as it helps organizations automate network configuration and management tasks while improving network monitoring and troubleshooting.

One of the primary benefits of NetOps is that it enables organizations to reduce the time and resources required to manage network infrastructure. By automating network configuration and management tasks, organizations can streamline network operations and free up valuable resources that can be devoted to other tasks, such as developing new network functions and services. This can help organizations remain competitive in a rapidly changing market where agility and speed are essential.

NetOps also helps organizations improve the reliability and scalability of their networks. By providing network operations teams with the tools and processes to automate network configuration and management tasks, NetOps can help organizations identify and address problems more quickly. This can reduce downtime and increase network uptime, ensuring that network services are always available and performing optimally.

One of the key features of NetOps is its emphasis on automation. By automating network configuration and management tasks, organizations can reduce the risk of human error, which is a common cause of network downtime and outages. Automation can also help organizations scale their networks more efficiently, as it eliminates the need for manual configuration and management tasks.

Another important aspect of NetOps is its focus on network monitoring and troubleshooting [1] [2] [3] [5]. NetOps provides network operations teams with the tools and processes to monitor network performance, identify potential issues, and troubleshoot problems when they occur. This can help organizations improve the reliability and availability of their networks while minimizing the impact of network outages on end-users.

In addition, NetOps can help organizations improve their compliance with regulatory and industry standards [10] [11]. By automating network configuration and management tasks, organizations can ensure that their networks are configured according to best practices and standards [12]. This can help organizations avoid costly compliance violations and maintain the trust of their customers and partners.

However, NetOps may not be well-suited for organizations that require more cross-functional collaboration between development and operations teams [13]. While NetOps can help organizations manage network infrastructure at scale, it may not be effective in promoting collaboration between teams. Therefore, organizations should carefully consider their specific needs and requirements before choosing NetOps as their primary development methodology.

In summary, NetOps is an important development methodology for organizations looking to adopt network softwarization [14]. By providing network operations teams with the tools and processes to automate network configuration and management tasks, NetOps can help organizations streamline network operations, improve network monitoring and troubleshooting, and increase the reliability and scalability of their networks.

2.3. Verification

Verification is an essential methodology for network softwarization as it helps ensure that software-based networks are reliable, secure, and perform as expected.

In today's fast-paced business environment, software development has become increasingly complex, and software defects and security vulnerabilities can have significant consequences. In the context of network softwarization, software defects or security vulnerabilities can potentially cause network downtime, data loss, or breaches, which can have serious financial and reputational impacts on organizations.

Verification addresses these challenges by providing a set of tools and processes that enable developers to test and validate software applications more effectively. Verification methodologies involve testing, analyzing, and validating software code to ensure that it meets the expected functionality, security, performance, and quality standards. Verification tools can include automated testing frameworks, code analysis tools, and performance monitoring tools, which are essential for developing and managing software-based networks.

Automated testing frameworks can help organizations test software code more effectively and efficiently, thereby reducing the risk of defects and vulnerabilities. Automated testing frameworks can also reduce the time-to-market for new network functions and services, as they enable developers to test code quickly and efficiently, without the need for manual testing. Code analysis tools can help organizations analyze software code for defects and vulnerabilities and identify potential areas for improvement. Code analysis tools can also help organizations ensure that software code adheres to coding standards, best practices, and compliance requirements.

Performance monitoring tools can help organizations monitor the performance of software-based networks and identify potential performance issues before they impact the network's availability or reliability. Performance monitoring tools can also help organizations identify potential bottlenecks in the network and optimize network performance.

The benefits of verification for network softwarization are significant. By ensuring that software is thoroughly tested and validated before deployment, organizations can reduce the risk of security vulnerabilities and performance issues. Verification can help organizations improve network reliability and security, while ensuring that software performs as expected. This can help organizations remain competitive in a rapidly changing market where agility, reliability, and security are essential.

However, verification can also present some challenges. Verification methodologies can be time-consuming and require significant investments in tools, technologies, skills, and expertise. Verification methodologies can also require significant knowledge of software code and network architecture, which may not be readily available in some organizations. Additionally, verification may not be suitable for organizations with strict time-to-market requirements, as it can potentially delay the release of new network functions and services.

In conclusion, Verification is an essential methodology for network softwarization, as it helps ensure that software-based networks are reliable, secure, and perform as expected [3] [5] [15]. By adopting Verification methodologies, organizations can reduce the risk of defects and vulnerabilities, improve network reliability and security, and remain competitive in a rapidly changing market.

3. Comparison and Contrast

Fundamental principles of network principles should be chosen carefully, depending on organizations' specific needs and requirements by considering the approach methodologies and ensure that they have the necessary tools, technologies, skills, and expertise to implement them effectively. All of these methodologies, DevOps, NetOps, and Verification have its own theoretical strengths and weaknesses.

DevOps emphasizes collaboration, communication, and automation between software developers and IT operations teams. The theoretical foundation of DevOps lies in Agile software development, where iterative and incremental development is prioritized. DevOps also draws on Lean management principles, emphasizing the elimination of waste and continuous improvement. By integrating development and operations teams, DevOps aims to reduce the time-to-market, improve software quality, and foster a culture of collaboration and accountability.

NetOps, on the other hand, is focused on network operations, automating network configuration and management tasks, and improving network monitoring and troubleshooting. The theoretical foundation of NetOps lies in the principles of network automation, which aim to simplify network management and improve network visibility. NetOps emphasizes the use of network automation tools, programming languages, and frameworks to improve the scalability, reliability, and security of network infrastructure.

Verification is based on the principles of software testing and quality assurance. The theoretical foundation of verification lies in the software engineering discipline, where testing is a critical part of the software development life cycle. Verification methodologies aim to ensure that software applications are thoroughly tested and validated before deployment, reducing the risk of security vulnerabilities and performance issues. Verification includes techniques such as unit testing, integration testing, regression testing, and performance testing, as well as code review and analysis [1]-[10].

4. Conclusions

The adoption of appropriate development methodologies is critical to the successful adoption of network softwarization. The choice of development methodology depends on various factors such as organizational culture, regulatory requirements, network size and complexity, and specific goals and priorities. Organizations should carefully consider these factors and weigh the benefits and limitations of each approach before choosing a particular method.

It is important to note that each development methodology has its own strengths and weaknesses. DevOps emphasizes collaboration and communication between software developers and IT operations teams, streamlining the software development process, reducing time-to-market, and improving software quality. NetOps focuses specifically on network operations, providing a set of tools and processes to automate network configuration and management tasks, and improve network monitoring and troubleshooting. Verification emphasizes testing and validation in software development, ensuring that software is reliable, secure, and performs as expected.

Organizations may benefit from combining elements of multiple methodologies or adapting them to suit their specific needs and circumstances. For example, a DevOps approach may work well for rapidly developing new network functions and services, while a NetOps approach may be better suited for managing and maintaining existing network infrastructure at scale. Similarly, a Verification approach may be necessary to ensure the reliability and security of network functions, even in organizations that prioritize speed and agility.

The adoption of appropriate development methodologies is critical to the successful adoption of network softwarization, enabling organizations to effectively manage network infrastructure, improve the automation of network management tasks, and ensure that software is reliable, secure, and performs as expected. For businesses to stay ahead of the competition, they must take advantage of the full capabilities offered by softwarization and benefit from its advantages. This will help them stand out and thrive in a rapidly changing market.

In summary, organizations must carefully consider their unique needs and circumstances to select the development methodology that best suits their goals and priorities. By choosing an appropriate methodology and integrating it effectively, organizations can achieve their network softwarization goals, streamline operations, and gain a competitive advantage in the market.

Further Reading

- "Network Softwarization: From Legacy Networks to NFV and SDN" by Marcos Dias de Assunção and Guillaume Pierre.
- "Network Function Virtualization: Challenges and Opportunities for Innovations" by Raj Jain, JieHui, Subbarao Kalyanaraman, Narseo Vallina-Rodriguez, and Liren Wang.
- "Software-Defined Networking: A Comprehensive Survey" by Diego Kreutz, Fernando M. V. Ramos, Paulo E. Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, and Steve Uhlig.

Conflicts of Interest

The authors declare no conflicts of interest regarding the publication of this paper.

References

[1] Kreutz, D., Ramos, F.M.V., Verissimo, P.E., Rothenberg, C.E., Azodolmolky, S. and

Uhlig, S. (2015) Software-Defined Networking: A Comprehensive Survey. *Proceedings of the IEEE*, **103**, 14-76. https://doi.org/10.1109/JPROC.2014.2371999

[2] Han, B., Gopalakrishnan, V., Ji, L.S. and Lee, S. (2018) Network Function Virtualization: Challenges and Opportunities for Innovations. *IEEE Communications Magazine*, **56**, 218-225. https://ieeexplore.ieee.org/document/7045396

[3] Bifulco, R., Canonico, R., Carpentieri, B. and Persico, V. (2019) A Comprehensive Survey on Verification and Validation Techniques for Network Functions Virtualization. *IEEE Communications Surveys and Tutorials*, **21**, 675-712.

[4] Silva, T.L., Jamshidi, P., Pasquini, R., Rothenberg, C.E. and Guimarães, R. (2019) A Comprehensive Review of Verification and Validation Techniques for Soft-Ware-Defined Networking and Network Function Virtualization. *IEEE Communications Surveys and Tutorials*, **21**, 1164-1193.

[5] Sharma, P. and Dhaka, V. (2020) NetOps: Enabling Network Operations in Software-Defined Networks. *Journal of Network and Computer Applications*, **146**, Article No. 102470.

[6] Han, Y., Li, Q. and Li, H. (2018) A Survey of Network Verification Techniques. *IEEE Communications Surveys and Tutorials*, **20**, 591-605.

[7] Huang, X., Yang, Y., Zhou, J., Li, J., Zhang, Y. and Li, Y. (2019) A Survey on DevOps: Aligning Development and Operations. *IEEE Access*, **7**, 172603-172622.

[8] Bhuyan, S. and Sanyal, S. (2018) DevOps in SDN/NFV: A Systematic Literature Review. *IEEE Communications Magazine*, **56**, 78-84.

[9] Sodhi, J.S. and Bansal, M. (2020) Comparative Analysis of DevOps Tools: A Review. *IEEE Access*, **8**, 58568-58591.

[10] Ocampo-Martinez, C., Zhang, Q. and Boucadair, M. (2018) A Survey on the Interplay between Network Softwarization and Security. *IEEE Communications Surveys and Tutorials*, **20**, 2097-2120.

[11] Jiang, X., Huang, J., Zhang, W., Chen, X. and Jin, H. (2020) An Overview of Network Softwarization: Opportunities and Challenges. *IEEE Access*, **8**, 112705-112725.

[12] Hasan, M.A. and Ben Salem, M. (2020) NetOps and Automation in 5G: Opportunities and Challenges. IEEE Communications Magazine, **58**, 148-155.

[13] Hassan, Q.M., Othman, M. and Abubaker, A. (2017) DevOps: State-of-the-Art and Research Challenges. *Journal of Software Engineering Research and Development*, **5**, 1-21.

[14] Anderson, M., Corso, G., Li, J. andVellaithurai, C. (2020) Network Operations (NetOps): An Overview. In 2020 *IEEE Conference on Network Function Virtualization and Software Defined Networks* (*NFV-SDN*), 113-118.

[15] Nguyen, D.H., Nguyen, H.T., Nguyen, T.A. and Nguyen, H.H. (2020) A Survey on Network Softwarization: From SDN to SDA, NFV, and Beyond. *IEEE Communications Surveys and Tutorials*, **22**, 858-892.