Architecture Scheme of DevOps for Cross Network and Multiple Environment Collaboration

Li Liu
Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan, China
liuli@caep.cn

Dongmei Xie
Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan, China
xiedongmei@caep.cn

Yunchang Cheng
Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan, China
yunchang@mail.ustc.edu.cn

Gongliang Li
Institute of Computer Application, China Academy of Engineering Physics, Mianyang, Sichuan, China
ligl@caep.cn

ABSTRACT

In the process of traditional enterprises exploring the middle platform architecture, DevOps is the cornerstone of its digital platform's long-term iterative construction and sustainable operation and maintenance. Aiming at the problems of physical isolation of multiple environments and cross regional collaboration of R&D teams faced by high security applications, based on the systematic analysis of DevOps concept, support tools and adoption situation, and based on the DevOps standard, this paper designs a DevOps architecture scheme for internal and external collaboration within cross-network multiple environments.

CCS CONCEPTS

• **Software and its engineering**; • **Software creation and management**; • **Software development process management**; • **Collaboration in software development**; • **Software notations and tools**; • **Software configuration management and version control systems**; • **Software libraries and repositories**; • **Security and privacy**; • **Software and application security**; • **Software security engineering**;

KEYWORDS

DevOps, Cross-network collaboration, Software outsourcing, High security, Physical isolation

ACM Reference Format:
Li Liu, Dongmei Xie, Yunchang Cheng, and Gongliang Li. 2021. Architecture Scheme of DevOps for Cross Network and Multiple Environment Collaboration. In *The 5th International Conference on Computer Science and Application Engineering (CSAE 2021), October 19–21, 2021, Sanya, China.* ACM, New York, NY, USA, 5 pages. https://doi.org/10.1145/3487075.3487116

CSAE 2021, October 19–21, 2021, Sanya, China

ACM ISBN 978-1-4503-8985-3/21/10…$15.00
https://doi.org/10.1145/3487075.3487116

1 INTRODUCTION

Gartner predicts that by 2025, the digital industry will promote the transformation of all industries, and digital transformation is the inevitable trend of all industries. The digital platform of traditional monomer application architecture has some problems, such as long development lifecycle, low demand response efficiency and difficult operation and maintenance [1]. At present, Internet enterprises and traditional enterprises are exploring a digital platform with a new front-middle-back application architecture to support the core competitiveness of rapid business response and modular innovation. The new digital platform requires that the software system can evolve rapidly online, so as to shorten the time to realize the commercial value, and always ensure high availability. The traditional model of separation of R&D and operation stages and mutual independence of teams is no longer competent. Therefore, DevOps (a combination of development and operations) has emerged [2].

DevOps aims to reshape the software process from the aspects of culture, automation, standardization, architecture and tool support, break the barriers between R&D and operation, so as to shorten the cycle from code submission to product launch on the premise of ensuring high quality. DevOps is the cornerstone of its digital platform's long-term iterative construction and sustainable operation and maintenance.

However, many traditional enterprises face the following problems: Alleviating the shortage of development resources through outsourcing, and even need cross regional cooperation; The production environment is strictly controlled, and the source code cannot be tested and saved; Physical isolation of multiple environments (development environment, test environment, production environment), etc. Such enterprises are difficult to directly adopt the standard DevOps architecture and need to explore a new DevOps architecture scheme for internal and external collaboration within cross-network multiple environments.

The structure of this paper is as follows: Firstly, this paper systematically analyzes the concept, support tools, adoption and maturity of DevOps, as well as the problems of physical isolation of multiple environments and cross regional collaboration of R&D teams faced by digital platforms. Then, an architecture scheme of DevOps for cross-network and multiple environment collaboration is designed, including overall process, environment construction and application release collaboration. Finally, it introduces the architecture application effect and the next research direction.

2 DEVOPS DEVELOPMENT ANALYSIS

DevOps has been proposed for more than ten years. As an extension of agile method in the complete software life cycle, DevOps aims to break the barriers between development and operation. The following systematically introduce DevOps, including the concept, support tools, adoption and maturity.

2.1 DevOps Concept

The concept of DevOps was first proposed in Europe in 2009. Its emergence is due to the software industry's increasingly clear recognition that in order to deliver software products and services on time, development and operation must cooperate closely, and pay attention to the communication and cooperation between software developers and operators. Although the DevOps movement has been discussed for more than ten years and new concepts, practices and tools continue to appear, there is no consistent and clear definition of DevOps. DevOps aims to better optimize software development (DEV), quality assurance (QA), operations (OPS) processes and integrate development and operation through highly automated tools and processes, so as to make software construction, testing, release, operation, maintenance and even the whole life cycle management faster, more frequent and reliable [3]. This paper adopts the definition proposed by Andrej et al. [4]: DevOps is a collaborative and multidisciplinary effort within an organization to automate continuous delivery of new software versions, while guaranteeing their correctness and reliability.

DevOps is the evolution of software development life cycle (requirements, design, development, testing, deployment and delivery) from waterfall to agile and then to lean [5]. The waterfall model is that each stage goal of the whole software must be 100% completed before entering the next stage (as shown in Figure 1a). Compared with the waterfall model, the agile development process focuses on minimizing the available software. Each sprint iteration cycle has the ability to deliver, and the overall delivery frequency is faster (as shown in Figure 1b). DevOps is a continuation of agile development. It expands and improves the continuous integration and release process, so it can ensure that the code is available and can bring value to customers (as shown in Figure 1c) [6].

2.2 DevOps Automation Tools

At present, popular technologies such as micro service software architecture, cloud computing infrastructure, continuous integration and continuous deployment, program behavior analysis, big data analysis and intelligent operation and maintenance have created technical conditions for the implementation of DevOps. DevOps is a change of traditional software development practice, in which automation is in the key position. Because high-quality delivery with short lead times requires a high degree of automation, and rapid feedback acquisition also requires automation.

The research on automation support tools in DevOps practice is also increasing, and there are many mature models for the classification of DevOps automation support tools. Xebialabs provides the periodic table of DevOps tools [7], and StackOverdrive provides the panorama of DevOps tools. A new method for classifying DevOps tools based on data mining technology is also proposed in the academic circles [8]. There are more and more DevOps practices

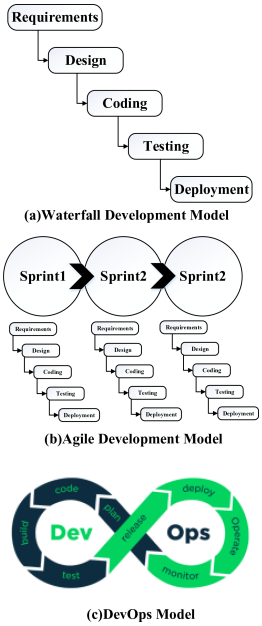

Figure 1: Evolution of Software Development Model.

in enterprises. For example, Taikang group introduces the whole process tool chain to improve the automation level in view of the problems of too much manual intervention and low automation of software delivery, as shown in Figure 2 [9].

2.3 DevOps Adoption

Many world-famous companies such as Google, Facebook, Amazon, LinkedIn and Netflix are adopting DevOps or providing relevant support products. From the six global DevOps state surveys conducted by puppet labs from 2014 to 2019 [10], the proportion of departments using DevOps has increased from 16% in 2014 to 28% in 2019, and further integrated safety related functions. The distribution of industries using DevOps also shows an expansion trend. A few years ago, DevOps was mainly used in science and technology industries, but now DevOps has penetrated into many industries, including finance, retail, telecommunications, education, medical treatment, government affairs, health, insurance, manufacturing and so on.

Liu et al. [6] investigates DevOps's eight aspects: IT performance, organizational culture and related practices, development and operation and maintenance practices, tool support, leadership, work proportion, employee engagement and satisfaction, and obstacles. At present, DevOps is not popular in China although some well-known enterprises, such as Tencent, Huawei and Alibaba, have adopted DevOps, most traditional industries have not adopted DevOps. Although domestic DevOps has shown a steady development trend, there is still an obvious gap compared with the international level [6].

2.4 DevOps Maturity Model

The overall framework of DevOps can be divided into three parts: agile development, continuous delivery and technical operation.

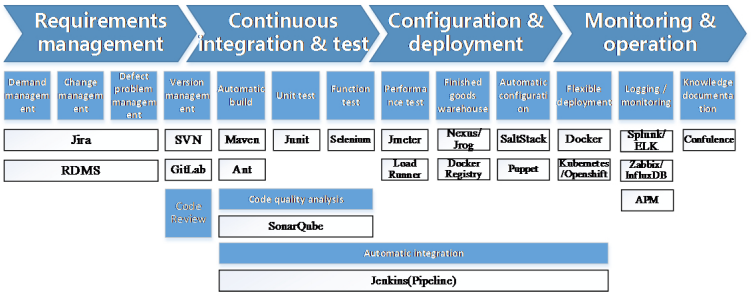

Figure 2: DevOps Whole Process Tool Practice.

DevOps capability maturity model is divided into five levels: Initial level, foundation level, comprehensive level, excellent level and fabulous level.

DevOps standard defines the capability maturity level of each capability item in detail. Participating enterprises can conduct capability analysis, improvement suggestions and expected effect analysis based on the evaluation results of the standard. From the current participating enterprises, the most common problems include [11]:

- Development mode based on release window;
- The automatic test system is not perfect, the accumulation of use cases is not enough, and the false alarm rate is high;
- Tool chains are scattered and lack of platform;
- Insufficient quality built-in of the assembly line;
- Weak capability of measurement visualization and driving improvement;
- There is a gap between project / team capability and organization capability;
- Organizational isolation: product (demand), development, testing, operation and maintenance, and engineering efficiency (platform).

3 DEVOPS PROBLEM IN HIGH SECURITY ENVIRONMENT

DevOps makes software construction, testing and release faster, more frequent and more reliable through automated processes. However, due to the high security and confidentiality requirements of traditional enterprises, it is difficult to fully realize the automation advantages of DevOps. Traditional enterprises mainly face the following problems:

- Physical isolation of multiple environments (development, test and production environment). The production environment of high security applications has strict control requirements and must be physically isolated from the development environment and test environment, and the source code cannot be tested and saved online. Data center servers must also be operated by specific operators. Therefore, the pipeline automatic high-frequency deployment advocated by DevOps faces the natural gap of network isolation.
- The composition of R&D team is complex and coordinated across regional. Digital platform is becoming the core engine of enterprise digital transformation. The IT team of Internet native enterprises generally has thousands of people, which can be qualified for continuous support. However, most traditional enterprises' IT team resources are very limited. In the initial construction for the large-scale application, a large number of manufacturers must be introduced. It is faced with the control problems of complex composition of R & D team, parallel development of project groups and cross regional coordination.

4 DEVOPS ARCHITECTURE FOR MULTIPLE ENVIRONMENT COLLABORATION

Aiming at the physical isolation status of three environments: manufacturer's development and testing environment, internal development and testing environment and production environment, based on the idea of DevOps and supporting tools, this paper designs a cross-network collaborative DevOps scheme, which includes three parts: overall process, application release collaboration and environment construction, As shown in Figure 3

4.1 Overall Process

The cross-network collaborative DevOps process is divided into five links: agile development, testing, verification, deployment online and operation monitoring. Outsourcing development and on-site development can be carried out in parallel.

- Agile development links defines the development management process, as well as requirements management, defect handling, code branching, product library management, application profile management, database script, package management and other specifications.
- The test and verification links are similar, define the specifications, test methods and requirements of unit test, function test, integration test, interface test, UI test, stress test, etc.
- Deployment online links defines the resource application, online deployment process and emergency problem handling process.
- Operation monitoring links defines the daily operation and maintenance, change management and application offline specifications to ensure the normal operation of applications in the production environment.

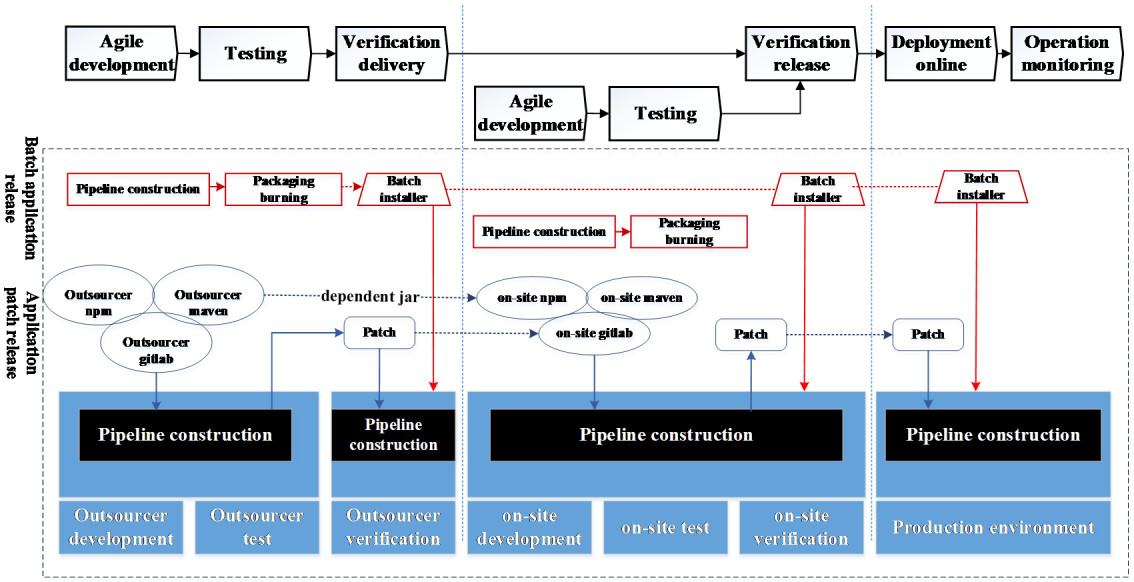

Figure 3: Cross-network Collaborative DevOps Architecture.

4.2 Environment Construction

Environment construction define seven sets of environments, including manufacturer development environment, manufacturer test environment, manufacturer verification environment, on-site development environment, on-site test environment, on-site verification environment and production environment, to manage the application operation environment required by the whole R&D process. If the test verification of the previous environment fails, the code cannot enter the next environment.

4.3 Application Release Collaboration

The installer mode is adopted for the first deployment, and the pipeline upload war mode is adopted for application patch deployment. When building a code running package, the code must pass static code security detection and running code security test. At the same time, when the code version changes, the manufacturer and the on-site Maven and NPM libraries need to be synchronized in time to keep the code version dependency consistent.

At the same time, different environments have different quality control requirements. The four types of environments support four types of continuous delivery pipeline tasks: development, testing, verification and production, as shown in Figure 4. The development environment pipeline includes merge to develop, code scanning, compilation build, deployment mirrors, automatic test, development test and requirement confirmation test. The test environment pipeline includes merge to develop, continuous integration, interface test, new function test, performance test, compatibility test, security test, regression test, requirements verification and test review. The verification environment pipeline includes merge to release, continuous integration, new function testing, regression testing, test review and project review. The production environment pipeline includes merge to master, deployment preparation, regression test (optional), gray test (optional) and code tagging.

4.4 Effect of the Scheme

This architecture has been used in a digital platform construction project of a traditional enterprise; the digital platform adopts a new front-middle-back application architecture. At present, there are more than 50 app parallel pipelines and more than 200 people for collaborative R&D. DevOps support tools include: npm, maven, GitLab, Junit, Zendao, Jira, pinpoint, SonarQube, Jenkins, etc. Through the standardized DevOps overall process, multiple environmental control and application release collaboration, the application development, construction, testing and release can be faster, more frequent and reliable on the premise of ensuring security and confidentiality.

5 CONCLUSIONS

In the process of traditional enterprises exploring the middle platform architecture, DevOps is the cornerstone of its digital platform's long-term Iterative construction and sustainable operation and maintenance. Aiming at the problems of physical isolation of multiple environments and cross regional collaboration of R&D teams faced by high security applications, based on the systematic analysis of DevOps concept, support tools and application situation, and based on the DevOps standard, this paper designs a DevOps architecture scheme for internal and external collaboration within cross-network multiple environments. The scheme The DevOps architecture includes three parts: overall process, environment construction and application release collaboration, which makes the application development, construction, testing and release more fast, frequent and reliable on the premise of ensuring security. Subsequently, based on the cross-network collaborative DevOps architecture, the detailed process, organizational structure, engineering practice specifications, environmental tools, etc. will be further designed to ensure the stable and efficient operation of all application systems and middleware.

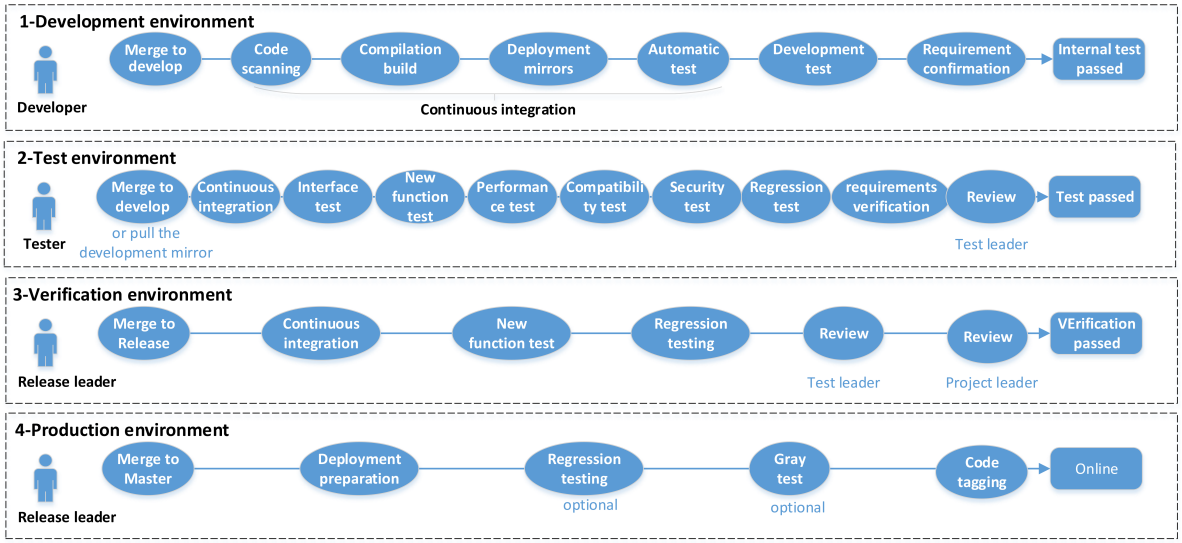

Figure 4: Pipeline in Different Environments.

REFERENCES

[1] L.Y Yan, X.Z Yang and Z.W He, *et al*(2018). Micro-service solution of operator's business platform. Telecommunications Science, 11, 166-174.

[2] G.P Rong, X.Y Bai and T Yue (2019). Preface of DevOps oriented software engineering new technology. Journal of Software, 30(10), 3054-3055

[3] X Dong, Y Guo and J Wang(2018). Continuous integration method based on DevOps capability model. Computer Engineering and Design,39(7), 1930-1937

[4] D Andrej, P Ralf and L Horst (2015). Towards definitions for release engineering and DevOps. In Proceedings of the IEEE/ACM 3rd International Workshop on Release Engineering, 3-3

[5] K Gene, H Jez, D Patrick, and W John. Translated by Z Liu (2018). The DevOps Handbook: How to Create World-Class Agility, Reliability, and Security in Technology Organizations, Posts and Telecommunications Press, Beijing, China.

[6] B.H Liu, H Zhang and L.M Dong (2019). Survey on State of DevOps in China. Journal of Software, 30(10), 3206-3226

[7] C Necco (2016). The Periodic Table of DevOps Tools v.2 Is Here. Retrieved from: https://blog.xebialabs.com/2016/06/14/periodic-table-DevOps-tools-v-2/

[8] H Huang, H Zhang, D Shao (2019). Practical Impacts of Automation Tools in Support of DevOps in China. Journal of Software,30(10), 3056-3070

[9] H Guo (2019). DevOps system and tool platform construction of TAIKANG group. Shenzhen GOPS2019

[10] Puppet Labs (2019). 2019 State of DevOps Report.https://puppet.com/resources/report/state-of-DevOps-report/

[11] X.L Niu, Y Jing (2019). Interpretation of DevOps certification standards and authoritative Case Guide. Shenzhen GOPS2019