

Practice of Software Development Based on Devops

Lanlan Gong^{1*}, Guifen Gong², Xinghong Ling³

¹Suzhou City University, Suzhou, China

²Kunshan Rural Commercial Bank, Kunshan, China

³Soochow University, Suzhou, China

*Corresponding Author.

Abstract:

Software project development is a comprehensive process, in current software development process, it is found to be lack of standard management, code management, test and quality assurance, and the unity of development, testing, deployment and operations. As to these problems, Devops (Development+operations) developing mode is introduced. This paper introduces the whole workflow of Devops which is based on continuous delivery, investigates the Devops process in detail, and brings it into software project implementation, carrying on the whole process management from project management, code version management, automatic test, continuous integration and continuous deployment.

Keywords: *Devops, Continuous Integration, Continuous Deployment, Automation, Agile Development, Software Project, Quality Assurance, Operations.*

I. INTRODUCTION

In recent years, Devops (a combination of Development and Operations) gains gradual popularity, and enterprises at home and abroad successively adopt Devops model for software development management. Domestic Devops still lags behind foreign countries, and there is a shortage of talents. Devops is rarely involved in the cultivation of computer-related talents. Software development is a process of comprehensive practice. At present, software developers mostly focus on coding, and rarely mention the overall project management, test, integration, deployment, operation and maintenance of the software project. In general, software project development mainly faces the following problems: (1) Lack of standardized project planning and management (2) Lack of standard code version management (3) Lack of effective unit test, integration test, automation test, code quality analysis, quality assurance (4) Lack of organic unity and connection in all development stages, and lack of awareness of overall system operation and maintenance. This paper introduces the Devops model into software project

practice, implements full-process management and practice from project management, code version control to automation test, integration, deployment, etc. It aims to strengthen developers' Devops process awareness, help them understand and participate in the complete software product production process, familiarize with and master related technologies and tools.

II. DEVOPS

Devops is a combination of development and operation and maintenance^[1], a complete set of IT operation and maintenance-oriented workflow^[2], which is used to promote communication and collaboration between software developers (Dev), technical operation and maintenance (Ops) and quality guarantee (QA) departments, achieve seamless connection in the work process^[1]. The Devops concept diagram is shown in Fig 1. Based on IT automation and continuous integration (CI), continuous deployment (CD), it aims to optimize all links such as development, test, and system operation and maintenance^[3]. Based on the automatic continuous software delivery process, it makes behaviors such as construction, test, and release faster, more efficient, and more stable^[4].



Fig 1: devops concept

Devops came into being because software organizations are increasingly aware that if product delivery time is to be shortened as much as possible under the premise of guaranteeing quality, development and operation and maintenance work must be closely integrated.



Fig 2: devops software development model

As shown in Fig 2, Devops covers multiple stages of demand planning, development, construction, test, release, and operation and maintenance^[1]. As the continuation and enhancement of agile development, it makes expansion and improvement on the basis of agile development. The processes of continuous integration, deployment, and operation and maintenance are added to guarantee stability of the entire system, rather than being limited to a certain development stage.

Devops practice is mainly reflected in three aspects: development methods, automation and quality assurance.

(1) The development method mainly refers to which development method is used, whether continuous project planning and management is performed during the development process. It mainly involves the two tasks of workflow planning and coding. The current agile method is consistent with Devops in terms of automatic construction, automation test, continuous integration and continuous delivery, which is thus widely recommended and used^[5].

(2) Automation. Through automated construction, test, release, deployment and operation and maintenance, it is possible to achieve continuous integration and test, continuous delivery and deployment, creating an automated delivery pipeline.

(3) Quality assurance. After the code submission, integration and automation test are performed to find problems and defects in time. Code rollback is triggered when errors are captured, which is an effective automated mechanism for quality assurance. Through visualization, project data is directly presented to strengthen communication and cooperation among software developers, quality assurance personnel, and operation and maintenance personnel.

Automation is the key to the practice of Devops, whose core content is to achieve continuous integration and continuous deployment (CI/CD) based on a series of automation

tools. Therefore, tools are the basis for automation^[6], and a complete and easy-to-use tool chain is essential for Devops. With the continuous development and maturity of Devops, more and more tools support Devops^[7]. Where, the widely used and recommended tools are shown in Table I.

Table. I. Devops automatic tools

Type	name
Build	Maven, Ant, ansible
CI	Jenkins, GitLab CI, Bamboo
Container management	Docker, Kubernetes
Test	Junit, SonarQube, Selenium
Project Management	Jira, TFS
Code management	Github, GitLab, git

III. DEVOPS IMPLEMENTATION

Practice is carried out in accordance with Devops continuous delivery process in the development of Python-based ECN system. As shown in Fig 3, the Devops system of the ECN system uses Scrum agile development method to plan and manage projects, and uses automated tools for continuous integration, construction, test, and deployment. At the same time, various tests and quality analysis methods are introduced in the development stage and integration stage to guarantee quality. In addition, data at various stages including operation and maintenance are visualized so that discovered problems or defects are fed back to relevant personnel in a timely manner, thus enhancing communication and cooperation among developers, operation and maintenance personnel, and quality assurance personnel in quality inspection, achieving timely problem solution.

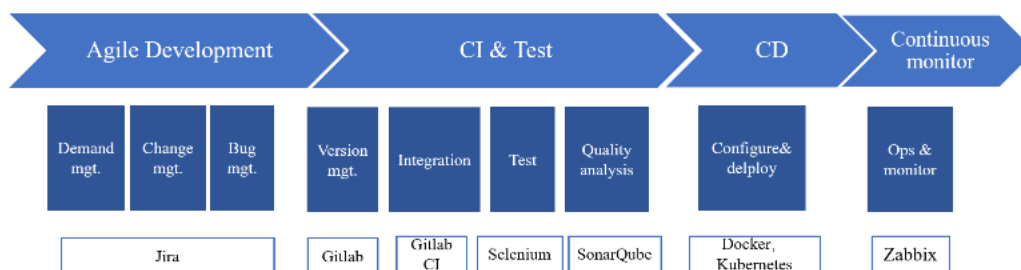


Fig 3: devops setup of ecn system

3.1 Project Management and Planning

According to the Scrum method, the requirements of the entire software product are divided into several user stories, and project development is divided into several iteration cycles (sprints) for completion. The requirements analysis during the development process is also carried out iteratively, and each iteration can be completed by selecting several user stories. Each iteration cycle is designed for the current target requirements.

The ECN project uses Jira as project management tool to formulate product iteration plans, split and assign tasks for each iteration. The task can be in one of several states: To Do, In Progress, Review, User test, Done and Pending. The initial task state is To Do. The developer estimates and fills in the time required to complete the task when receiving the task, and drags the task to the In Progress state when performing the task, and then drags the corresponding task to the corresponding state bar whenever the task progresses to a different stage. The project manager can view all user stories, iteration plans, track the state of all tasks through the jira Kanban board, and understand the project progress through the burndown chart, so as to make adjustments to the work progress.

Using the Scrum agile development method, complex projects are split and iterated to reduce the development difficulty, help developers (students) proceed with project development in a standardized and orderly manner, and master the system development process.

3.2 Code Version Control

The problem of code version control is an inevitable problem for all software development teams, and also the primary problem demanding urgent solution in current practice. This system

uses Git as a version control tool and GitLab as a code base server. Through version control, the problem of code submission and merging of different members is solved, and members are required to pull the latest code from the code base and combine it with the local code every day to implement development work. After completing a task and before submitting a code to the code base, one need make sure that the submitted code is runnable without any bugs. Otherwise, it will affect others' development work and operation of the entire system.

In the GitLab code base, one can view the number of code submissions of the entire team on a periodic basis, and count the number of code submissions by each developer. The contribution data of team members is clear at a glance, which facilitates convenient and intuitive team work assessment.

3.3 Continuous Integration and Deployment (CI/CD)

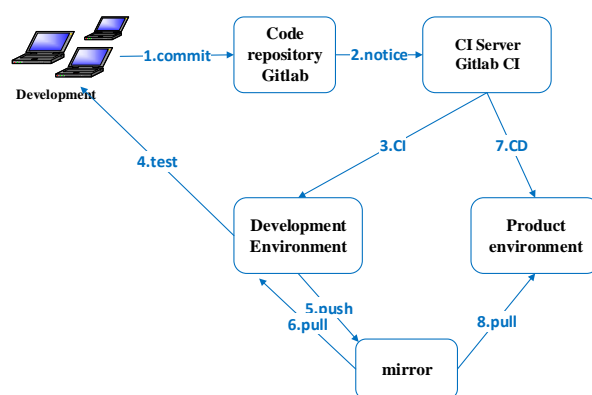


Fig 4: structure of continuous integration system

The core of Devops workflow is continuous integration and continuous deployment (CI/CD). Continuous integration belongs to developers' automated process. After developers submit code to the code warehouse, automated construction and automation test are carried out and combined with the original code. The main purpose is to find integration errors as early as possible. Continuous deployment means frequent automatic deployment of reviewed new version code to the production environment. It mainly solves the problem of slow manual deployment and long delivery time^[8].

GitLab CI integrates the CI/CD function and supports continuous code integration and

continuous deployment. The ECN system defines the continuous integration and continuous deployment workflow pipeline through GitLab CI to build a continuous integration system, as shown in Fig 4. The continuous integration server is connected with other links in the entire system, playing a core role as the key to automation^[8,9]. When the developer submits the code to the GitLab code base, GitLab notifies the GitLab CI integration server, and CI triggers automatic construction according to the conditions, performs integration and test, and makes deployment after the test is passed.

GitLab CI one-time integration pipeline can cover multi-stage processes, such as automatic construction, automation test, and automatic deployment. Any submission or combination request to the code base can trigger the pipeline. The ECN system builds a continuous integration workflow according to the following three steps.

(1) Define workflow

The core of continuous integration and deployment based on GitLab CI is to build the definition of workflow. GitLab CI is responsible for configuring and scheduling tasks (job), and distributing tasks to GitLab Runner for execution. GitLab CI uses `.gitLab-ci.yml` file to define and configure the workflow. The main contents are as follows:

1) In the `.yml` file, first define all stages of the pipeline, which are divided into four stages: test, report, build, and deploy in this project. The work of the four stages is executed in sequence.

2) Define and configure a job for each stage. A stage can have multiple jobs. Each job defines the command scripts to be executed (test script, build script or deployment script) and triggers conditions. The job operation can only be triggered when the developer's submission meets the corresponding conditions.

3) Specify a GitLab Runner for each job, and the Runner will perform the job.

4) Submit the `.gitlab-ci.yml` file to the GitLab code base.

(2) Install and configure Runner

GitLab Runner is a runner used to perform specific construction tasks, which can be a physical machine, a virtual machine, a container or a cluster of containers^[10]. GitLab CI and Runners communicate through API. Hence, as long as the Runner can access the network, it

can connect to the GitLab CI server.

Multiple Runners of the ECN system are installed on several different physical machines to perform different tasks. During installation, one need configure the corresponding CI server information (URL and token) for the Runner, so that the Runner can find and register the CI, and CI can recognize the Runner. In this way, communication is possible.

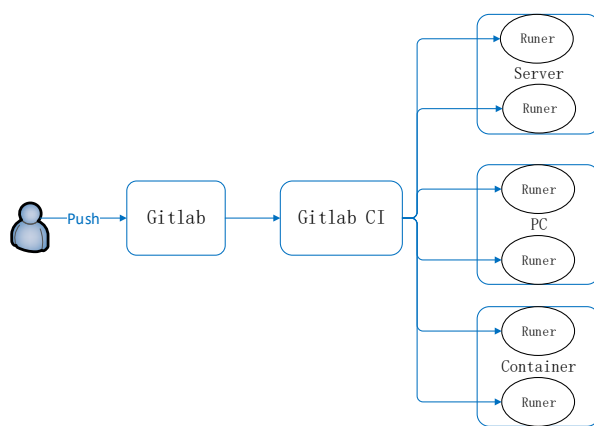


Fig 5: relationship of gitlab and runners

(3) Perform construction tasks

As shown in Fig 5, whenever a developer submits code to the code base, GitLab will notify GitLab CI of the change. Then, GitLab CI will find the Runner associated with this job according to the GitLab-ci.yml trigger condition, and notify this Runner to execute the job.

After receiving the Job, the Runner sends a confirmation request, and then starts to execute the task. During this period, the intermediate results are output to GitLab CI at regular intervals. After the task is processed, the Runner will send the final results, status, logs, etc. to GitLab CI.

In the pipeline, all jobs in the same stage are executed in parallel, and one can define whether the job at each stage must be executed after all the jobs in the previous stage are executed successfully.

The continuous integration pipeline in this paper executes the jobs in the test, report, build, and deploy stages in sequence. Finally, all the jobs in the deploy stage are executed successfully. Then, commit will be marked as successful. If any job in the previous stage fails, the commit will be marked as failure, and the job of the next stage will not be executed.

Through CD integration in the Pipeline, automatic release to multiple environments is possible, such as staging and production environments^[11]. The ECN system is deployed in four environments: test, dev, staging, and prod. In order to better support the release in multiple environments, this system uses Docker containerization technology^[12,13] to save the generation mirror configured in integrated build environment to the mirror library, so that the build environment can be reproduced and transplanted quickly. Kubernetes is used to perform container orchestration management.

CI/CD allows software teams to quickly release code, helps teams release applications automatically, thus shortening the release life cycle.

3.4 Quality Assurance

Based on continuous delivery process, Devops helps the software team automate the release and shorten the release cycle. At the same time, it also attaches importance to quality assurance QA. Not only delivery should be fast, but also the overall quality of the delivered products is guaranteed, so that operation and high quality are possible.

In the ECN system development process, in order to do a good job of QA, unit test, integration test, system test and code quality analysis are implemented respectively.

(1) Local code analysis: Before submitting the code, each developer need use the static code analysis tool Pylint to analyze the code quality locally, analyze the errors in the Python code, and find the code that does not meet the code style standards and has potential problems.

(2) Unit test: Each developer is required to compile the corresponding unit test code and submit it after the local test is passed. In the integration stage, the automated pipeline will re-perform unit test tasks, and use the code coverage tool Coverage for coverage checking.

(3) Automation test: Automation test is carried out in the continuous integration stage. According to the definition of CI/CD pipeline, the corresponding test script is executed to perform module function test and interface test. After the test is passed, it enters the next stage. Before the code is integrated into the trunk, if the system captures an error, an automatic rollback will be triggered.

(4) Automated code quality analysis: In the second stage of pipeline definition, the report stage performs automated code quality analysis to generate code quality reports. We use

SonarQube, the most popular code quality and security analysis tool in CI. When the test is passed, the quality of the startup code is comprehensively scanned and analyzed. The scan results will present data like the number of bugs, defects, coverage, repetition rate in the form of reports.

(5) Code review: In addition to automation test tools and code quality analysis tools, a code review link is also set up. After the code is submitted, a dedicated person will review the submitted code to check whether there are logic, business problems, or other coding deficiencies. After problem submissions, it will be directly fed back to the corresponding developers for improvement.

(6) System test: It is mainly to test whether the system meets the users' needs. After a series of previous tests, the user environment is simulated to conduct a complete test of the system. The ECN system uses a combination of manual test and WEB automation tool Selenium test for system test.

3.5 Visualization

Devops process involves all stages of the software development life cycle. From requirements, planning, code version control, automation test, quality analysis, integration and deployment, operation and maintenance, etc., a variety of and large amounts of data will be generated. Intuitive and clear display of these data is very important for the whole project.

To make project plans, split and assign tasks, project managers and development teams need view the task status at any time and follow up project progress. These data of the ECN system can be well mastered through the Web client of the project management tool Jira, as shown in Fig 6.

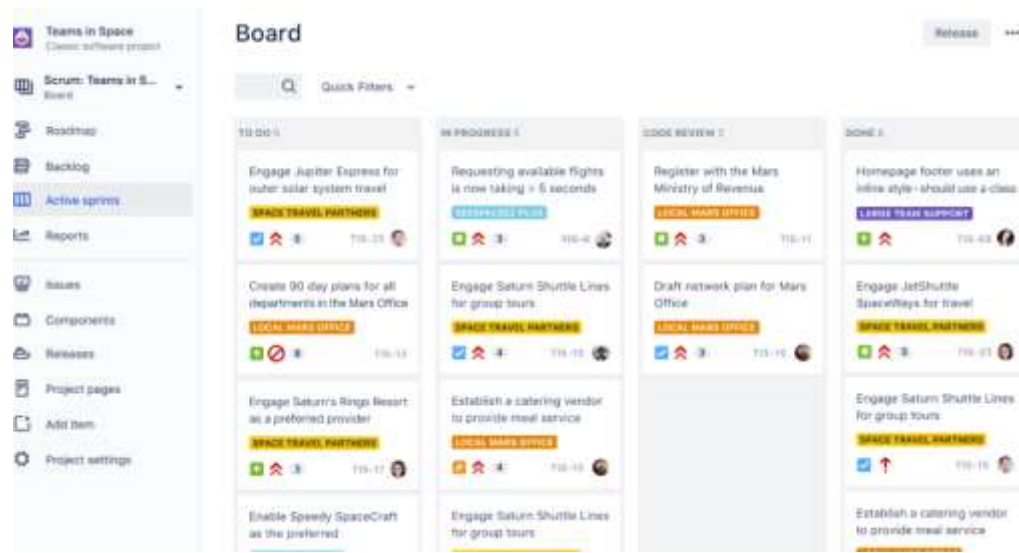
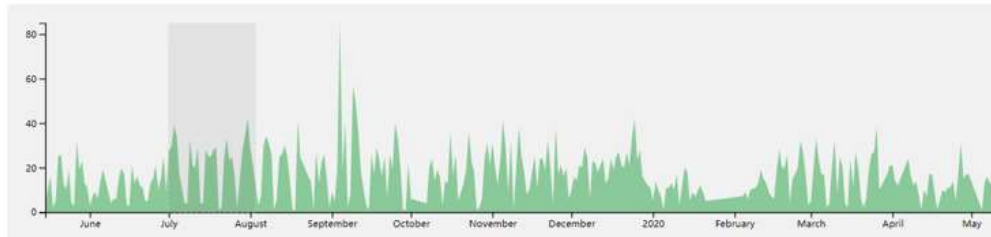


Fig 6: visualization of project management

June 30, 2019 – August 3, 2019

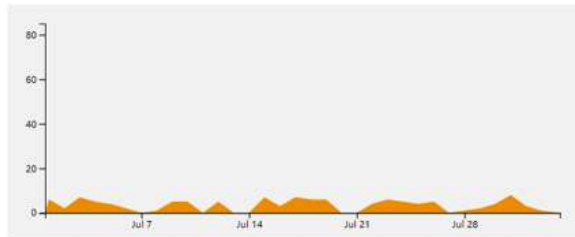
Commits to dev, excluding merge commits. Limited to 6,000 commits.



xerxes

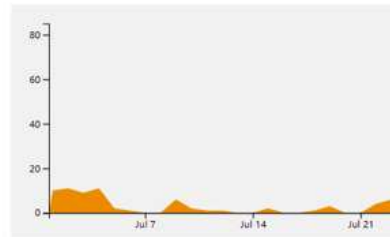
111 commits

346287244@qq.com



Dylan

105 commits



May

101 commits

zhuangcm@chinamaker.net



jeffreychan

78 commits



Fig 7: visualization of version control

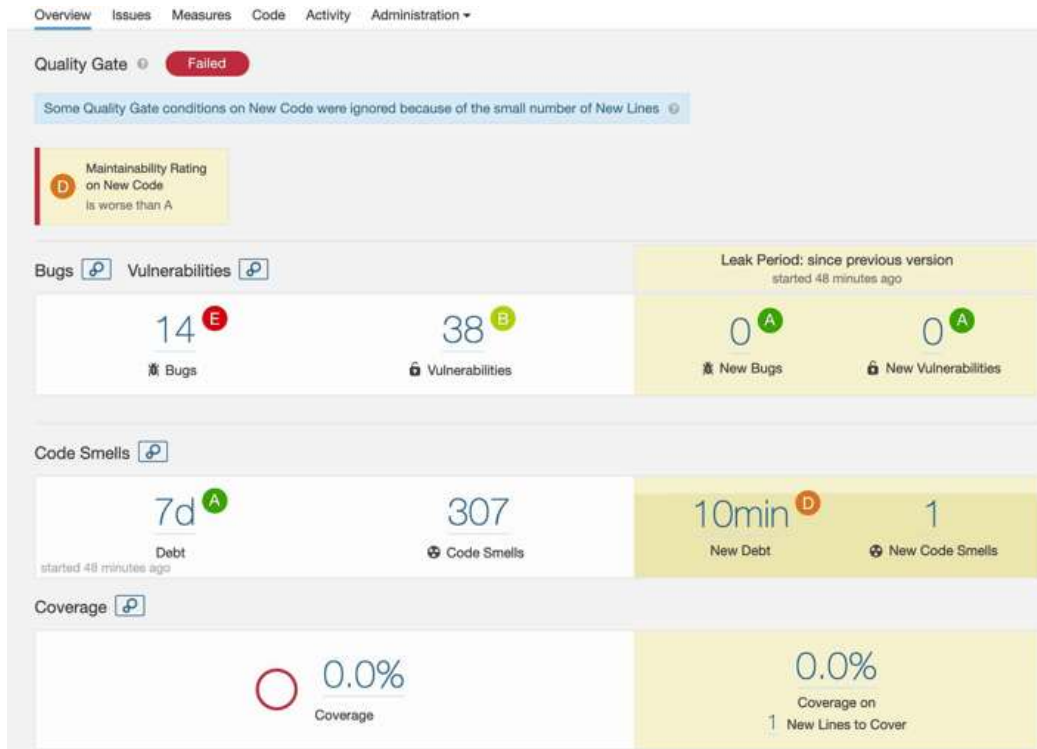


Fig 8: visualization of quality analyze

The code version data is particularly important to the development team. Through the version control tool GitLab, the division status of each developer's work, the daily pull of code, the time and number of updates submitted, the amount of code contribution are all visualized on the web in the form of chart, as shown in Fig 7.

The test and quality analysis data generated in the automation tools SonarQube and Selenium in the continuous integration stage are also presented in the form of reports, and the relevant data will be automatically sent to the responsible personnel for timely feedback and problem solution, as shown in Fig 8.

After the system is deployed online, the last stage is operation and maintenance. The system is maintained and monitored during the user's use. The data visualization generated in this process is a very important basis and guarantee for timely response of the operation and maintenance team. The system uses the currently popular web-based distributed operation and maintenance monitoring tool Zabbix to monitor the system status, logs, container status, and container application logs, so that operation and maintenance personnel can be flexibly notified, and existing problems can be quickly located and resolved.

IV. DEVOPS ABILITY DEVELOPMENT

ECN system development is the first software development using the Devops model. Through this Devops-based software practice, developers have a preliminary understanding and experience of Devops, and have a deeper understanding of each software development stage during the implementation process. At the same time, the use of many automation tools is also learned.

Devops is becoming an important skill for IT personnel. Introduction of the Devops model into software project development practice for higher comprehensive practical ability in continuous delivery-based software development is mainly carried out through Devops knowledge, Devops workflow, automation tools, roles and responsibilities, and Devops culture, as shown in table II.

Table. II. Devops knowledge and technique demands

Devops knowledge	Devops model, continuous integration, continuous deployment, automation test, containerized management.
Devops Workflow	Requirements analysis and planning, coding, test integration, and deployment of tasks in each stage.
Devops tool	Jira, Git, GitLab, GitLab-CI, Docker, Kubernetes, Selenium, SonarQube, Zabbix, etc.
Devops role	Project administrator (Scrum Master), development engineer, test engineer, operation and maintenance engineer, Devops engineer and the responsibilities of each role.
Devops culture	Team collaboration, continuous delivery, continuous improvement, automated instrument, monitoring and test.

Among the many roles of Devops, Devops engineer is different from other roles, which is a recently emerged role with the popularity of Devops. It is mainly responsible for the production and continuous maintenance of software products, which promotes close collaboration between the development team, operation and maintenance team and quality assurance team, playing an important role throughout the entire Devops process. It thus imposes higher requirements for comprehensive knowledge and skills of Devops engineers. Generally speaking, the skills and responsibilities of a qualified Devops engineer are shown in Table III.

Table. III. Skills and responsibilities of devops engineers

Skill requirements		Responsibilities
Programming language	More than one programming language, scripting language	Design and evaluate automation scripts and systems
Operating system	Linux, Windows, process management, file system, virtualization technology, etc.	Perform system troubleshooting and problem solving across platforms and application domains
Network and security	HTTP/HTTPS/ SSL/TLS and other network protocols	Design solutions for network communication, data transmission, system security and reliability
Infrastructure management	Container technology, container orchestration management, configuration management, etc.	Design and build R&D visualization systems, continuous integration system, and interface with automation test platforms and virtualization service systems
CI/CD	Continuous integration, build tools	
Software and hardware operation, maintenance and monitoring	Hardware monitoring, software monitoring, log management, etc.	Participate in the construction of the operation and maintenance system to improve project visibility and traceability
Team collaboration, communication	Communicate and collaborate with development, test and operation teams.	Effectively manage projects through open, standards-based platform; improve product quality and reduce development costs through collaboration,

V. CONCLUSION

Devops-based software development method advocates better communication, collaboration and integration between development, operation and maintenance and quality assurance teams to support faster delivery of high-quality software products. The process includes continuous development, continuous integration, continuous deployment and continuous monitoring links, in which automated continuous integration and continuous deployment are the core, and a complete automated tool chain is the necessary foundation.

Devops method is introduced into software project practice, and Devops practice is carried out in the ECN system development. The implementation process, methods, and the use of automated tools are introduced mainly from the aspects of project planning and management, code version control, continuous integration and deployment.

For improvement in Devops capabilities of the software team, the Devops knowledge and skills goals are given, and the skills and responsibilities of the Devops engineer are summarized.

Through Devops practice, the software project management is more standardized, the development cycle is shortened, the deployment frequency and quality are higher and more efficient.

ACKNOWLEDGMENTS

This work was supported by Livelihood Science and Technology Project of Suzhou, Jiangsu (SS201736) and Higher education research Project on teaching reform of Jiangsu (2017JSJG473).

REFERENCES

- [1] Liu B H, Zhang H, Dong L M (2019) Survey on State of DevOps in China. *Journal of Software*, 30 (10): 3206-3226.
- [2] Rong G P, Zhang H, Shao D (2017) *Devops: Theory, Method and Practice*. BeiJing: China Machine Press, : 15-36.
- [3] Cai J J, Ren Nver, Wei J J (2019) *The Practice of Devops in Software Development*. *Computer Knowledge and Technology*, Vol. 15, No. 29: 242-244.
- [4] Geng Q F, Li X, Ge W (2019) *Software Development Management Mode Based on DevOps*. *COMPUTER ENGINEERING & SOFTWARE*, Vol. 40, No. 1: 93-96.
- [5] Elberzhager F, Arif T, Naab M, Süß I, Koban S (2017) From agile development to Devops: Going towards faster releases at high quality—Experiences from an industrial context. In: *Complexity and Challenges of Software Engineering in Emerging Technologies*. Springer-Verlag, 33-44.
- [6] Huang H, Zhang H, Shao D (2019) Practical Impacts of Automation Tools in Support of DevOps in China. *Journal of Software*, 30 (10): 3056-3070.
- [7] Jin Z F, Zhang Y W, Ye W H (2019) Research on Application of DevOps in Documentation towards Full Value Delivery. *Journal of Software*, 30 (10): 3127-3147.
- [8] Dong X, Guo Y, Wang J (2018) Continuous Integration method based on Devops capability model. *COMPUTER ENGINEERING AND DESIGN*, Vol.39, No. 7: 1930-1937.
- [9] Zhang Z C, Luo T J (2018) CCI: a containerization-based continuous integration system. *Journal of University of Chinese Academy of Sciences*, Vol. 35, No. 4: 569-575.
- [10] Liu W L (2019) The Design of CI-CD automatic Integration in Multiple Environments. *Modern Computer*, 02: 83-87.
- [11] Getting started with GitLab CI/CD. (2020-07-23). https://docs.gitlab.com/ee/ci/quick_start/README.html.

- [12] Qiao W, Zhao W R (2018) Research of Devops development status and trend. *Digital Technology & Application*, Vol. 36, No. 4: 74-76.
- [13] Chen M, Yang Y (2019) Research on the Design and Implementation of DevOps System Based on Docker. *China Computer & Communication*, 11: 97-98, 101.