UDC 004.9:005.8

Software security risk management in DEVOPS methodology

Olga Kolisnichenko[1], Mykhailo Kolomytsev[1], Svitlana Nosok[1]

[1]*National Technical University of Ukraine «Igor Sikorsky Kyiv Polytechnic Institute», Educational and Research Institute of Physics and Technology*

Abstract

It's impossible to talk about cloud technologies, modern applications and, in general, digital transformation, and not to mention security. The same applies to software development, in particular the DevOps methodology.

DevOps is a software development methodology that focuses on communication, integration, and collaboration between IT professionals ensuring rapid product deployment. DevOps practice reflects the idea of continuous improvement and automation. Many practices are designed for one or more stages of the development cycle.

Three hundred hours spent on software development can be wasted in just 30 seconds, if only one defect during operation is detected. This, subsequently, can ruin reputation of the whole product, and as a result there will be no choice but to simply remove it from the market. And this establishes the importance and necessity of quality control [1].

To ensure quality of software products during development risk management should be used at every stage of the DevOps lifecycle. Implementing DevOps without paying attention to security will definitely increase risks of attacks. Risk is the occurrence of an uncertain event that positively or negatively affects measured criteria of project success. These can be events that have happened in the past or current events, or something that may happen in the future. These uncertain events can affect target, business, technical and qualitative objectives of the project [2].

Main stages of risk management include:

- risk identification;
- risk analysis and assessment;
- risk response;
- risk monitoring and control.

The purpose of the article is to choose the method of risk analysis and assessment in the DevOps methodology. When using risk management model in DevOps, it is important to choose risk assessment method that is optimal for the criteria relevant to DevOps. To such criteria authors attribute cost of resources, time spent, accuracy of evaluation. Therefore, the choice of risk assessment method is an important component of the process of creating secure software. Risk assessment methods such as PRisMA, PRAM, FMEA, DREAD and FTA were considered in this work.

Keywords: Risk Management, Product Security, Vulnerability Search, Development Methodology, DevOps

Introduction

Relevance . Nowadays, with the rapid growth of technology, software being hosted in the cloud with support of multiple operating systems, multiple platforms, complex IT infrastructures, etc. Whilst quality becomes a determining factor in software supply, it is constantly improved to meet requirements of users.

Development team works within limited time frame, so testing might not be given enough attention. Usually, assembly is submitted for testing at the last minute. In such case, there is not enough time and resources to perform all developed tests, and coverage of automation is not always 100%. There is no other way but to simply perform limited and important tests within available time and resources.

The solution here is using risk management to identify the most important risks, analyze them and identify the most necessary tests.

In distinction from the traditional method of simple detection of software defects, approaches and goals of quality assurance have changed over time due to changing technology, increasing competition in the market of qualitative software, implementing "Automate Everything" principle and general practice of flexible development methods, such as both Agile and DevOps for software delivery in a matter of hours [3].

So, the current trend of testing is not only to "detect defects", but also to:

- focus on the area of the product that can have a major impact on the business due to the high probability of failure during operation;
- focus on early detection of defects and allow team to correct them as soon as possible;
- pay maximum attention to customer service, emphasizing process of software support [4].

Task setting. For the most effective risk-oriented testing, it is necessary to conduct research on various methods of risk analysis and assessment. To analyze obtained results comparative characteristics of the methods effectiveness, need to be put together.

The purpose of this article. Conduct a comparative analysis of risk assessment methods in the DevOps methodology. Demonstrate that the most effective approach is the FMEA method, which shows itself best in risk assessment, and with the involvement of automated tools meets the principles of DevOps.

1. Research of methods analysis and risk assessment

To ensure that proper and accurate risk management is selected, a study was conducted to identify the best method of risk analysis and assessment, the results of which are put below. The following methods participated in the study - DREAD Microsoft, PrisMA, PRAM, FMEA and FTA.

Presented methods belong to different categories - informal and formal. Informal methods include PRisMA and PRAM, and formal methods include FMEA, DREAD and FTA.

A list of risks was obtained after testing a simple web application and these risks were assessed using each method. These risks are:

- Risk # 1 - XSS
- Risk # 2 - SQL Injection
- Risk # 3 - OS commands execution
- Risk # 4 - Path traversal
- Risk # 5 - DOS
- Risk # 6 - version information discousure.

The FMEA (Failure Mode and Effects Analysis) is a formal methodology used to identify and eliminate known or potential failures to improve reliability and security of complex systems and is designed to provide information for risk management decisions.

Each cause of failure is evaluated using three attributes - severity (Severity), priority (Priority), probability (Likelihood). The lower the risk priority (RPN), the higher the level of risk. The results of the evaluation for FMEA are presented in table 1 .

Table 1. Table of results for FMEA

	Severity	Likelihood	Priority	RPN
Risk #1	2	1	2	4
Risk #2	3	2	3	18
Risk #3	2	3	2	12
Risk #4	3	4	2	24
Risk #5	3	5	4	60
Risk #6	2	3	3	18

The FTA (Fault Tree Analysis) - a tool for risk management that detects adverse events or failures and presents them in a tree shape structure by means of simple logic and graphic projecting.

The PRISMA (Product RISk Management) is an approach to identify the areas that are most important for testing, ie the areas that have the highest level of business and technical risk. PRISMA has been successful in testing organizations because they use risk-based testing. The basis in the PRISMA process is creation of the so-called risk matrix shown in Figure 1.

Results of PRISMA assessment are presented in table 2.

PRAM (Probabilistic Risk Assessment Methodology) covers processes, techniques and methods that allow to analyze and manage the risks related to project.

For ease of use, PRAM is divided into two parts: risk analysis and risk management. Risk analysis is divided into two stages: qualitative analysis, that focuses on the identification and subjective assessment of risks, and quantitative analysis, which focuses on objective risk assessment. PRAM results of evaluation of are presented in table 3.

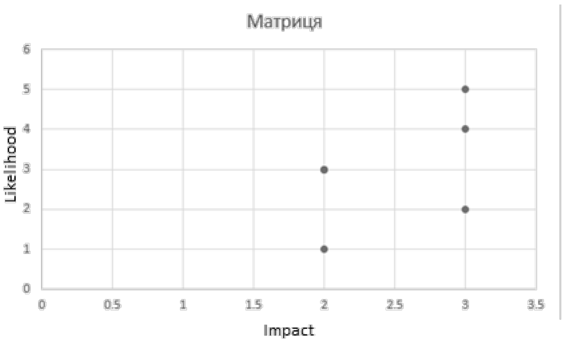

Fig. 1. PRISMA Risk matrix

Table 2. - Table of results for PRISMA

	Impact	Likelihood	Result
Risk #1	2	1	High
Risk #2	3	2	Average
Risk #3	2	3	Average
Risk #4	3	4	Low
Risk #5	3	5	Low
Risk #6	2	3	Average

Table 3. Table of results for PRAM

	Impact	Likelihood	Result
Risk #1	High	High	High
Risk #2	Average	Average	Average
Risk #3	High	Average	Average
Risk #4	Average	Low	Low
Risk #5	Average	Low	Low
Risk #6	High	3	Average

DREAD is part of computer security risk assessment system previously used by Microsoft, and although it is now used by OpenStack and other corporations [5], its' creators have stopped using it. DREAD provides assessment of security threats in five categories:

- damage potential (D);
- reproducibility (R);
- exploitability (E);
- affected users (A);
- discoverability (D)

Each vector is assigned a numeric value from 1 to 10, depending on the severity. Results of DREAD evaluation are presented in table 4.

Table 4. Table of DREAD results

	D1	R	E	A	D2	Result
Risk #1	7	10	8	7	5	37
Risk #2	3	4	4	4	5	20
Risk #3	3	1	7	6	3	20
Risk #4	4	4	3	3	2	16
Risk #5	4	2	5	4	3	18
Risk #6	7	6	7	5	4	29

2. Analysis of research results

DevOps methodology sets some restrictions on the choice of optimal risk assessment method. For example, typical restrictions for Azure DevOps are [6]:

- query execution time limit - 30 seconds
- query results are limited to $20,000$ lines
- query length - no more than $32,000$ characters

From the above data, we can conclude that to compare risk assessment methods it's appropriate to set such criteria as quality (accuracy) of assessment and speed of execution (time spent assessment), as there are time constraints. Therefore, we compare experimental results according to DevOps methodology relevant criteria - resource costs, time spent, accuracy of evaluation.

Results of the comparison are shown in table 5.

Table 5. Table of results

Method	Resource costs	Number of attributes	Time spent	Precision
PRISMA	Medium	2	Executed quickly	Has average accuracy
PRAM	Medium	2	Executed quickly	Has average accuracy
FMEA	Above average	3	It takes time	High accuracy
FTA	Above average	2	It takes time	High accuracy
DREAD	Above average	5	It takes a lot of time	High accuracy

The research has shown that DREAD risk assessment is the leader in all indicators, whilst FMEA and FTA are in second place. But given that the DevOps development methodology needs to be addressed as quickly as possible, the DREAD methodology is not the best choice. Therefore, usage of FMEA is suggested, as it demonstrates better results, takes less time and has fewer attributes to evaluate.

Conclusion

Therefore, implementation of risk management processes in software development using DevOps methodology helps to create and maintain the most secure software product. An important part of risk management is risk analysis and assessment. Chosen risk assessment method satisfies requirements of DevOps completely, and is the best option among considered methods. For improvement of risk management process, it's appropriate to use reputable scanning tools for maximum automation.

References

[1] P. Tolokonina "Positive and negative risks on the project" — Quality laboratory — 2017. [Online]. Available: https://quality-lab.ru/blog/positive-and-negative-risks-on-the-project/.

[2] A. Hawkins "What DevOps Means for Risk Management" — Cloud Academy — 2018. [Online]. Available: https://cloudacademy.com/blog/what-devops-means-for-risk-management/.

[3] "OWASP Risk Rating Methodology" — OWASP — [Online]. Available: https://owasp.org/www-community/OWASP_Risk_Rating_Methodology.

[4] B. Piper "Applying risk management to DevOps practices with Snyk and Datadog" — Snyk — 2021. [Online]. Available: https://snyk.io/blog/applying-risk-management-to-devops-practices/.

[5] A. Hawkins "What Is the DREAD Cybersecurity Model?" — Logix — 2019. [Online]. Available: https://www.logixconsulting.com/2019/12/18/what-is-the-dread-cybersecurity-model/.

[6] "Service limits and rate limits" — Microsoft — 2020. [Online]. Available: https://docs.microsoft.com/en-us/azure/devops/user-guide/service-limits?view=azure-devops.