DevOps Tools: Silver Bullet for Software Industry

Divya Srivastava, Madhushi Verma, Shashank Sheshar, and Madhuri Gupta

Abstract DevOps is a development and operations practice where engineers collaborate throughout the full-service lifecycle, from design to development to production support. DevOps is a newly emerging field in the field of Software application. Most of the giant companies have now shifted towards DevOps practices as it channelizes the development and operation process for any software development. The present chapter aims to provide the DevOps information for the basic to reach out to maximum people working in or planning to shift towards DevOps. Starting from the definition, the phases, tools, and security are discussed in the present paper. Each phase of the DevOps life cycle is discussed with the tools used in that phase. The DevOps practices are powered with several tools to provide end-to-end automation in software development. The present paper presents the basic knowledge of prevalent tools available for DevOps practices. Along with the DevOps automation, the chapter also gives an overview of DevOps security, its need, and its tool. The chapter covers the software phases and the tools used to automate it. It also provides information regarding the tool platform, availability, and usage. To emphasize more on DevOps, the chapter has also summarized the industrial and academic opportunities in DevOps.

Keywords DevOps · DevOps tools · Docker · Ansible · Git · Jenkins · DevOps security · CICD

These authors contributed equally to this work.

D. Srivastava (✉) · M. Verma · M. Gupta
CSET, Bennett University, Greater Noida, Uttar Pradesh, India
e-mail: divya.srivastava@bennett.edu.in

M. Verma
e-mail: madhushi.verma@bennett.edu.in

M. Gupta
e-mail: madhuri.gupta@bennett.edu.in

S. Sheshar
CSED, Thapar University, Patiala, Punjab, India

H. K. Thakkar et al. (eds.), *Predictive Data Security using AI*, Studies in Computational Intelligence 1065, https://doi.org/10.1007/978-981-19-6290-5_6

1 Introduction

DevOps has emerged as a "silver bullet" for improving software development efficiency and its operations. In 2009, Patrick Debois gave the word DevOps, which is a combination of two words: Development and Operation. It is an umbrella that describes all the phases of software development, coding, building, testing, delivery, and deployment, as shown in Fig. 1.

DevOps practices [1] help the organizations produce a quality product in less time. The state-of-DevOps report 2020 shows that even during the tough times in the year 2020, DevOps practices have yielded excellent results in "software delivery" and "change management." DevOps practices have helped IT organizations faster and easy delivery of quality products on a secured platform.

DevOps is an advancement of Agile Software Development [2]. DevOps practices advocated CICD (Continuous Integration and Continuous Delivery) that makes it fast and more efficient as compared to pioneer agile techniques [3, 4]. DevOps principles work to collaborate people towards a common business goal by focusing on People and Culture, Process and Practices, and Tools and Technologies. Thus, it covers all the areas required by any organization for speedy and efficient product delivery. Traditional software development practices focused only on the development part, and there used to be a chain for every phase of software development. The traditional practices were slow, and there has been an imbalance between delivery and unstable production in most cases. DevOps was thus created to balance by bringing everyone involved in software development and deployment under one umbrella. It encourages collaboration between the development and operational team; therefore, a more collaborative and efficient product is produced. DevOps is not only for developers and operators but also a combined effort of managers, engineers, and administrators. Every single person has an important role to play and contribute to DevOps.

DevOps is no doubt acting as a silver bullet in IT Industry but it comes along with a cost of security. DevOps Security also termed DevSecOps is a collection of practices that integrate software development (Dev), IT operations (Ops), and security (Sec) to improve an organization's ability to run and deliver applications and services quickly and securely, while DevOps practices bring tremendous advantages to IT sector

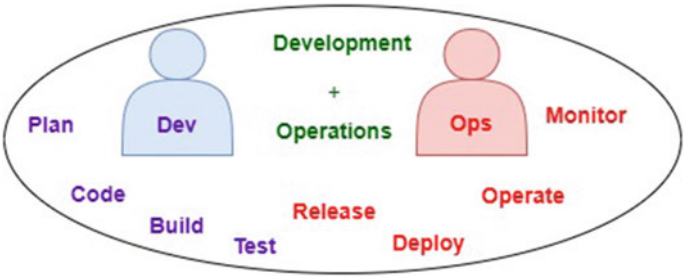

Fig. 1 DevOps: development and operation

by keeping the applications safe and secure. DevOps environments need to protect services, applications, and tools along with traditional development environments.

1.1 Background

Many renowned companies like Google, Microsoft, Amazon, etc., are now providing different tools and platforms for DevOps practices. The main reason behind DevOps' popularity is that people interested in system administration and software development find space while working in DevOps. They work together towards a common goal and achieve an efficient product quickly. Amazon refers to DevOps as a "combination of cultural philosophies" because it combines approaches used by software developers, testers, managers, operations, and other expertise. There has been great growth in DevOps practices in the recent few years, both in the field of industry and research [5]. The graph shown in Fig. 2 shows that the research in the field of DevOps geared up from 2015.

DevOps practices work in a cross-functional environment and thus require a set of tools that allow this. It is equipped with automated tools, making it a "SILVER Bullet" for software industries. Sometimes these tools are even termed "Toolchains" as DevOps practices focus on a collaborative work environment. The paper aims to build knowledge of various DevOps tools based on their respective job. It also provides the platform for the tool and is also available in terms of Open Source (Os), Enterprise (En), Paid (Pd), Free(Fr), and Free+paid (Fm (Free trial version and later on payment basis)).

The rest of the paper is organized as follows: DevOps Lifecycle is discussed in Sect. 2. DevOps Tools are discussed in Sect. 3. Section 4 discusses the use of DevOps in the Industry and Education Filed. Section 5 concludes the chapter along with the future prospects of the DevOps market all over the world.

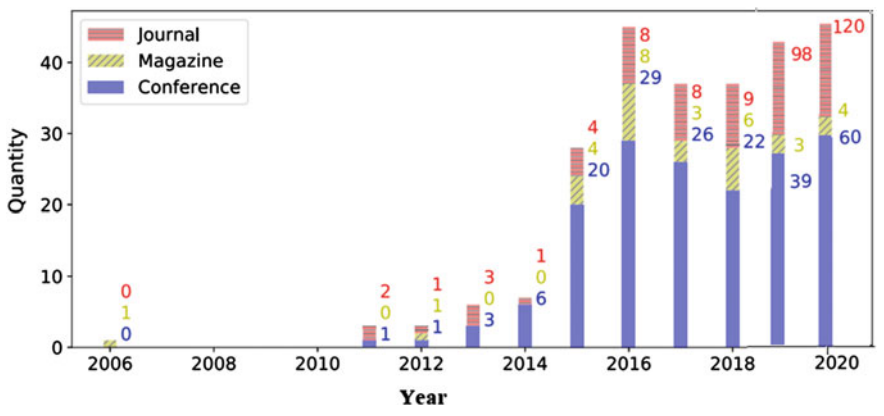

Fig. 2 DevOps Publications based on type and year [5]

2 DevOps Life Cycle

The DevOps lifecycle enhances advancement processes from beginning to end and connects with the association in a nonstop turn of events, bringing about quicker conveyance times. This cycle fundamentally comprises of the accompanying seven phases.

2.1 Continuous Development

Continuous Development includes arranging and coding the product. The whole improvement process becomes separated into more modest advancement cycles. This interaction makes it simpler for the DevOps group to speed up the general programming advancement process. This stage is instrumental in planning the vision for the whole improvement cycle, empowering engineers to comprehend project assumptions completely. Through this, the group begins imagining its ultimate objective too. There are no DevOps instruments needed for arranging; however, numerous adaptation DevOps tools are utilized to keep up with code. This course of code upkeep is called source code management or source version control system. Famous DevOps tools for source code upkeep incorporate JIRA, Git, Mercurial, and SVN. Also, there are various DevOps tools for bundling the codes into executable documents, like Ant, Gradle, and Maven. These executable records are then sent to the next phase of the DevOps lifecycle.

2.2 Continuous Integration

Continuous Integration (CI) incorporates various advances connected with the execution of the test interaction. Alongside this, customers also give feedback/data to add new elements to the application. Most changes occur in the source code during this stage. CI turns into the center for settling these regular changes every day or month-to-month premise. Developing code is a blend of the unit and reconciliation testing, code survey, and bundling. Since developers roll out continuous improvements, they can rapidly detect issues (if any) and resolve them at the beginning phase. This stage encounters persistent new code functionalities with the current source code. Jenkins is one of the most well-known instruments for continuous combination. It helps in getting the refreshed code and setting up an executable form.

2.3 Continuous Testing

Next in the DevOps lifecycle is the Continuous Testing stage, wherein the created code is tried for bugs and mistakes that might have advanced into the code. This is

where quality examination (QA) assumes a significant part in really taking a look at the ease of use of the created programming. Fruitful consummation of the QA interaction is essential in deciding if the product meets the customer's details. DevOps automated tools, like JUnit, Selenium, and TestNG, are utilized for continuous testing, empowering the QA group to examine various code bases simultaneously. Doing this guarantees no blemishes in the usefulness of the created programming. Also, Docker compartments are utilized in continuous testing to copy the real test environment. A Docker holder is an independent, lightweight executable container with everything to run an application: framework apparatuses, framework libraries, runtime code, and settings. Automated testing is done on DevOps tools like Selenium, after which the reports are produced on another computerization apparatus, for instance, TestNG. Automation of the whole testing stage additionally becomes conceivable with the assistance of the ceaseless combination apparatus Jenkins. Automated testing assumes an indispensable part in saving time, work, and exertion.

2.4 Continuous Deployment

Continuous Deployment (CD) ensures easy application deployment without influencing the application's presentation. It is important to guarantee that the code is sent unequivocally to all suitable waiters during this stage. This cycle disposes of the requirement for planned deliveries and speeds up the input component, permitting developers to resolve all the issues more rapidly and with more noteworthy exactness. The containerization concept in DevOps assists with accomplishing consistent deployment through various configuration management tools. A containerization device like Vagrant executes consistency across test, improvement, arranging, and creation conditions. Containerization manages to carry virtualization to the level of a working framework over various operating systems.

Continuous Deployment ensures to help the association to have a reliable testing environment setup. Configuration management tools hold significant worth in the persistent deployment stage. It includes arranging and keeping up with consistency in the useful prerequisite of the application. Well-known DevOps devices utilized for design the executives incorporate Ansible, Puppet, and Chef that assist with executing a speedy arrangement of new code.

2.5 Continuous Monitoring

Monitoring the execution of a product is essential as it helps to decide the general adequacy of the product/application. This stage processes significant data about the developed application. Through nonstop observing, developers can recognize general trends and ill-defined situations in the application where more exertion is required. Consistent observing is a functional stage where the goal is to improve the general productivity of the product application. Additionally, it screens the exhibition of the

application also. In this way, it is one of the essential periods of the DevOps lifecycle. Different framework mistakes, for example, “server not reachable”, “low memory”, and so forth, are settled in the consistent monitoring stage. It likewise keeps up with the accessibility and security of the administrations. Network issues and different issues are consequently fixed during this stage at the hour of their discovery. Tools like Nagios, Splunk, Sensu, ELK Stack, and NewRelic are used by the monitoring team to screen client activities for inappropriate conduct. Accordingly, during continuous monitoring, engineers can proactively look at the general strength of the framework. Proactive checking works on the framework’s dependability and usefulness and decreases upkeep costs. In addition, significant issues are straightforwardly answered to the development team to be adjusted in the underlying stages.

2.6 Continuous Feedback

Continuous Feedback is an essential input to learn and investigate the ultimate result of the application. It establishes the vibe for working on the current form and delivering another adaptation in view of feedback given by the customers. This stage plays a vital role in improving the existing version of the application based on the feedback input. This input is the data assembled from the customer’s end. The data is huge, as it conveys all the information about the existing version of the product and its related issues. It additionally contains ideas given by end clients of the product.

2.7 Continuous Operations

The last stage in the DevOps lifecycle is the briefest and least demanding to handle. Continuity is at the core of all DevOps tasks that automate release processes, permit developers to recognize issues rapidly, and assemble better forms of programming items. Continuation is critical to redirections and other additional means that block improvement. Development cycles in ceaseless tasks are more limited, permitting associations to publicize continually and speed up the general chance to showcase the application.

Thus, all the stages in DevOps Lifecycle improve the worth of applications by improving them and being more proficient. Therefore, it has emerged as a silver bullet in the software industry.

3 DevOps Tools

DevOps Tools have been categorized based on key aspects they perform. The categorization of tools is fuzzy i.e. one tool may fall into one or more categories. The tools are broadly categorized as:

- Code
- Build
- Test
- Delivery
- Deployment
- Monitor

3.1 Code

In this phase, the developers decide the toolkit to be used and install the plugins required to develop the application. Coding Nomenclature, clean code practices, etc., are part of this phase. Code Development, Merging of Codes, Code Review, Source Code Management, etc. Git is one of the most popular Source Code Management (SCM) used to maintain software versioning systems.

3.1.1 Code Management Tools

Git is the most popular tool for code repositories. It is an open source (Os) and is used for SCM. It operates on a client–server basis, where a central code repository is maintained. The central repository acts as a hub for both clients and developers, and both of them can simultaneously download the code from there. Linus Torvalds [6] is the creator of Git. Git makes it easier for remote teams to collaborate.

Git and its variants are used for version control as shown in Fig. 3, and all of them are OS. The other popular version control tools are Atlassian Bitbucket[7] whose trial version is available free, whereas, after that, it allows the user after payment (Free + Premium (Fm)). Similarly, another popular tool for version control is subversion (SVN) [8] and is made available as OS by Apache License.

3.2 Build

Once the developer shares the code on the repository and requests to merge their new code with existing ones on the repository, this is where the build operation comes in role. Building the codebase, Running end-to-end series, Integrating the code, Unit Testing, etc. If the build operation fails at any point of time, the developer is notified to resolve the issue. It minimizes the integration challenges that arise while working on a common codebase. The errors are exposed in the early stages of the development lifecycle by continually verifying code changes into a common repository and executing builds and tests.

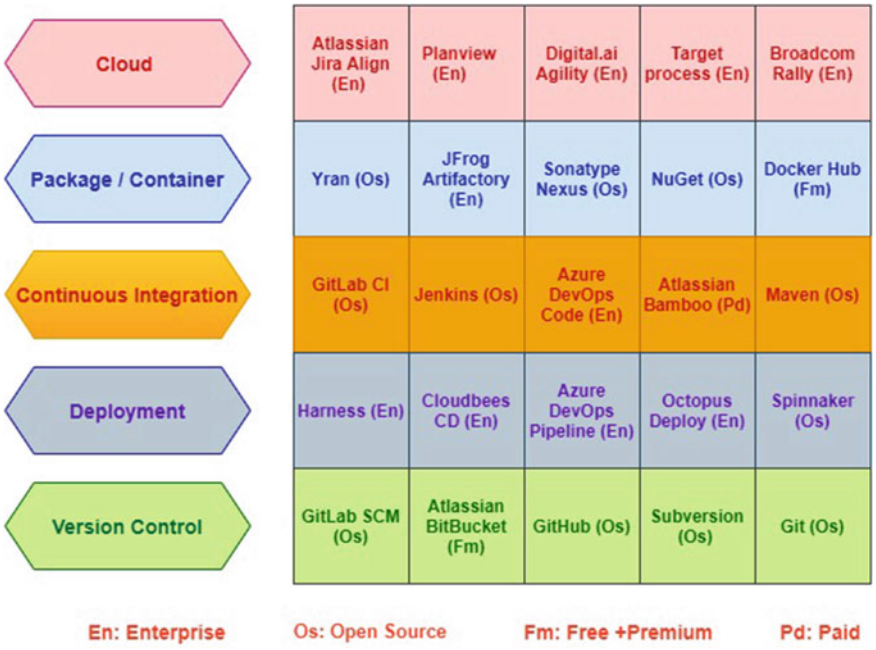

Fig. 3 DevOps tools

3.2.1 Build Management Tools

There are different tools available for the building phase. Some of the popular tools, such as Gradle [9], Jenkins [10], Docker [11], Kubernetes [12], are listed in Table 1. All of them are OS and are easily available.

3.3 Test

After the code is built, it is moved to the test environment. Test operations have been performed both ways, Manual as well as Automated. One of the most common forms of Manual testing is User/Client Acceptance Testing. People use the app as a client to indicate any bugs or enhancements that need to be fixed before going live. In the case of Automated Testing, security scanning against the application, checking for changes to the infrastructure and compliance with hardening best practices, evaluating the application's performance, and load testing are all examples of automated tests. The type of testing done during this phase depends on the organization and what is relevant to the application, but this stage may be thought of as a testbed where you can plug in new features.

Table 1 Popular DevOps tools and their usage

DevOps tools			
Tool name	DevOps phase	Availability	Icon
Git	Code, Build	Os	
Gradle	Build	Os	
Selenium	Test	Os	
Jenkins	Build, Test, Deploy	Os	
Puppet	Deploy, Operate	Os	
Chef	Deploy, Operate	Os	
Docker	Build, Deploy, Operate	Os	
Kubernetes	Build, Deploy, Operate	Os	
Ansible	Deploy, Operate	Os	
eG Enterprise	Monitor	Fm	

3.3.1 Test Management Tools

Several tools are available for automated tests. Selenium [13] is one of the most popular open-source tools for automated tests. It is used for automation testing of web-based applications. Selenium IDE is a simple tool for anyone to use, and it's built to operate with Firefox. The Firefox plugin allows users to record and replay test actions using a GUI (Graphical User interface). It also supports the creation and execution of test cases without knowing scripting languages. It supports Java,

C#, Perl, Python, Ruby, and Groovy. Another tool available for test automation is JUnit [14], which is the unit testing tool for Java programming language and is freely available over the internet. NEOTYS NEOLOAD [15] is a paid testing tool used by fortune companies to do the performance and load testing. It is one of the best tools covering all the enterprise testing requirements. Native hybrid and Mobile based applications can perform the test by using the enterprise version of Appium [16] automated tool. The other tools available are Tricentis Tosca [17], Sauce Labs [18], Parasoft [19], JMeter [20], Cucumber [21], etc.

3.4 Delivery

It is also called a Release phase in DevOps. It is when the application is ready to be delivered or released for deployment over the production environment. Thus, the first milestone is achieved in DevOps Pipeline. Depending on an organization's DevOps maturity, any build that reaches this pipeline level may be deployed automatically. Feature flags allow developers to switch off new features so that users don't notice them until they're ready to use. This is the nirvana of DevOps, and it's how companies manage to release numerous versions of their product at the same time.

3.4.1 Delivery Management Tools

The process of managing, planning, scheduling, and controlling the entire process of producing a piece of software through various phases of development and environments—such as testing and distributing software releases—is known as release management. There are several tools for release management. Some of the common tools are XL Release [22], IBM UrbanCode Release [23], Plutora Release [24], CA Release Automation [25], etc.

3.5 Deployment

The process of releasing the application to the production is deployed. There are several tools and methods that can be used to automate the release process and ensure that releases are reliable and have no downtime.

3.5.1 Deploy Management Tools

To automate the deployment tools, the common tools available are Jenkins, Puppet [26], Docker[26], Kubernetes [26], Chef [27], Ansible [28], etc.

3.6 Monitor

This phase observes the application being launched in the market. It is based on the market review, application performance, customer feedback/reviews, etc.

3.6.1 Monitor Management Tools

Several tools are available for monitoring the delivered product. Top in the list are Sensu [29], Pager Duty [30], Datical Deployment Monitoring Console [31], Splunk [32], etc.

Sensu is both infrastructure as well as application monitoring solution. It can be used to analyze, measure, and monitor the infrastructure, service health, application health, and business KPIs, among other things. Sensu seeks to answer modern-day difficulties in modern infrastructure platforms by combining static, dynamic, and ephemeral infrastructure at scale. Sense isn't a SaaS solution, but it does provide you with complete control over the availability of your monitoring system. It is widely used as it facilitates dynamic addition and deletion of the client, sends notifications and alerts, works with multi-tiered networks, and most importantly, fulfills the automation needs.

DevOps practices are implemented using the tools mentioned above, having functionalities like Configuration Management, Version Control, Test, Deploy, etc. The knowledge of these tools opens the areas for the engineer for various designations in Industry. Some of them are shared in the section below.

4 DevOps in Industry and Education

DevOps has helped industries maximize production in terms of quality and quantity. The DevOps practices following CICD help in continuous integration and delivery based on feedback. The popularity of DevOps in the industry is very well described in [33] and the authors termed it Industry 4.0 Industries are providing internships and recruiting people for several designations in the field of DevOps like:

1. DevOps Architecture
2. DevOps Tester
3. DevOps Developer
4. Release Manager
5. DevOps Security Expert

DevOps has recently gained so much popularity among the industries that now it is being run as a specialization course in some of the reputed universities both at undergraduate as well as postgraduate levels. Some of the postgraduate programs available are:

1. DevOps and Continuous Software Engineering by the University of Limerick [34]
2. DevOps by Letterkenny Institute of Technology, Ireland [35]
3. Software Engineering (Cloud Computing) by Torrens University, Australia [36]
4. Cloud Computing and DevOps by IIT Roorkee & Wiley, India [37]
5. Cloud and DevOps by EICT Academy IIT Guwahati & Microsoft, India [38]
6. Software Development—Specialization in DevOps by IIIT Bangalore, India [39]
7. DevOps Postgraduate Program by Caltech CTME & Simplilearn [40]

5 Conclusion and Future Perspective

The present paper discussed various DevOps tools, their usage, availability, etc. DevOps practices have improved the continuous delivery of valuable software. It supports modern interaction techniques that help in quick and reliable feedback for fast and efficient software delivery. DevOps has changed the way it needs to achieve IT security. When moving from long-term planned delivery of monolithic applications to an agile development environment, security needs to be deeply integrated into the development and operational processes. DevSecOps begins with a secure development lifecycle for the services and applications created and defined security patterns and ends with automated security for automated operations.

The paper has presented its importance in the industry as well as academics. The Global market insight predicts the DevOps Market growth by 2026 shown in Fig. 4. Japan and the U.S have also given their prediction for the development of DevOps as shown in Figs. 5 and 6, respectively. The market prediction shows that DevOps has an excellent future perspective and will build a better software world.

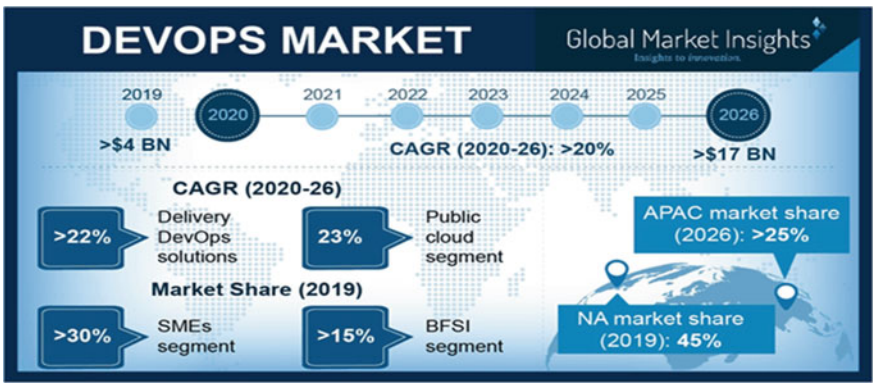

Fig. 4 DevOps market [41]

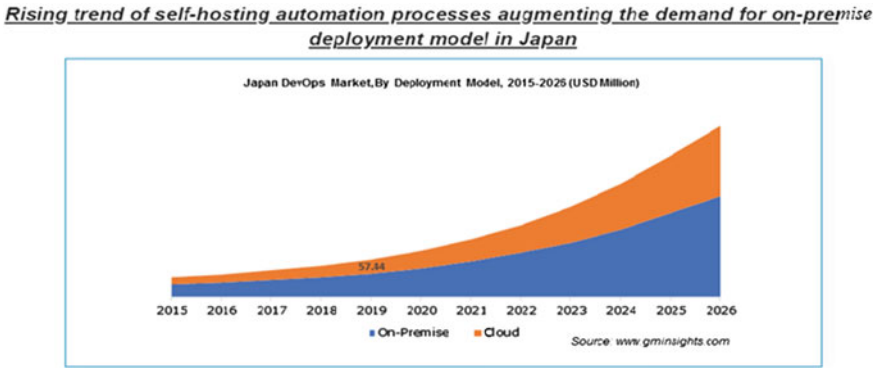

Fig. 5 DevOps market Japan [41]

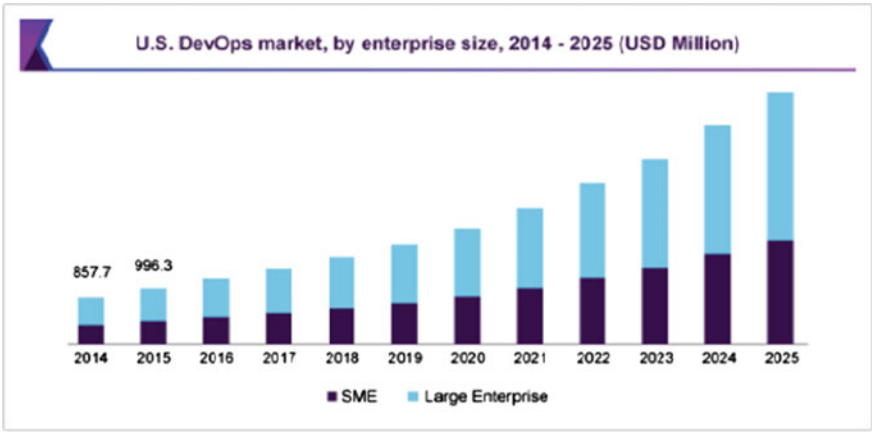

Fig. 6 DevOps market [41]

References

1. https://www.netsparker.com/devops-security-tools/
2. Christensen, H.B.: Teaching DevOps and cloud computing using a cognitive apprenticeship and story-telling approach. In: Proceedings of the 2016 ACM Conference on Innovation and Technology in Computer Science Education (2016)
3. Kroll, P., Kruchten, P.: The Rational Unified Process Made Easy: A Practitioner's Guide to the RUP: A Practitioner's Guide to the RUP. Addison-Wesley Professional (2003)
4. Pressman, R.S.: Software Engineering: A Practitioner's Approach. Palgrave Macmillan (2005)
5. Leite, L., et al.: A survey of DevOps concepts and challenges. ACM Comput. Surv. (CSUR) **52**(6), 1–35 (2019)
6. https://www.weave.works/blog/15-years-of-git
7. Light, J., Pfeiffer, P., Bennett, B.: An evaluation of continuous integration and delivery frameworks for classroom use. In: Proceedings of the 2021 ACM Southeast Conference (2021)

8. Pingrong, L., Xiaoquan, S., Junqin, Y.: Research on the application of DevOps in the smart campus of colleges and universities. J. Phys.: Conf. Ser. **1883**(1). IOP Publishing (2021)
9. Jokinen, O.: Software development using DevOps tools and CD pipelines: a case study (2020)
10. Agarwal, V., Krishnappa, H.K.: Robot framework integration with Jenkins: a blessing for automation (2021)
11. McKendrick, R.: Mastering Docker: Enhance Your Containerization and DevOps Skills to Deliver Production-Ready Applications. Packt Publishing Ltd (2020)
12. Paavola, E.: Managing multiple applications on kubernetes using GitOps principles (2021)
13. García, B., et al.: A survey of the selenium ecosystem. Electronics **9**(7), 1067 (2020)
14. Batra, P., Jatain, A.: Software Quality Enhancement Using Hybrid Model of DevOps, Intelligent Systems, pp. 281–288. Springer, Singapore (2021)
15. Srivastava, N., Kumar, U., Singh, P.: Software and performance testing tools. J. Inf. Electr. Electron. Eng. **2**(01), 1–12 (2021)
16. da Silva Lima, G.B., et al.: Devops methodology in game development with unity3D (2020)
17. Atesogullar, D., Mishra, A.: Automation testing tools: a comparative view. Int. J. Inf. Technol. Sec. **12**(4) (2020)
18. Jayasri, A.S.V.: An extensive risk-mitigating framework for continuous testing using DEVOPs (2020)
19. Angara, J., Prasad, S., Sridevi, G.: DevOps project management tools for sprint planning. Estim. Execut. Matur., Cybern. Inf. Technol. **20**(2), 79–92 (2020)
20. Kołtun, A., Pańczyk, B.: Comparative analysis of web application performance testing tools. J. Comput. Sci. Inst. **17**, 351–357 (2020)
21. Haver, T.: Cucumber 3.0 and beyond, PNSQC proceedings (2018)
22. Solouki, S.: Knowledge Management Practices in DevOps, Diss. Université d'Ottawa/University of Ottawa (2020)
23. Österberg, G.: A systematic literature review on DevOps and its definitions. Adopt. Benef, Chall. (2020)
24. Parashar, R.: Path to success with CICD pipeline delivery. Int. J. Res. Eng. Sci. Manag. **4**(6), 271–273 (2021)
25. Singh, R.: DevOPS now and then (2020)
26. Uphill, Thomas, et al.: DevOps: Puppet, Docker, and Kubernetes. Packt Publishing Ltd (2017)
27. Vadapalli, S.: DevOps: continuous delivery, integration, and deployment with DevOps: dive into the core DevOps strategies. Packt Publishing Ltd (2018)
28. McAllister, J.: Implementing DevOps with Ansible, vol. 2. Packt Publishing Ltd (2017)
29. Bovina, S., Michelotto, D.: The evolution of monitoring system: the INFN-CNAF case study. J. Phys.: Conf. Ser. **898**(9). IOP Publishing (2017)
30. Mishra, R.M., More, M.: A qualitative study of DevOps. **7**(1) (2021)
31. Kubryakov, K.: Deployment and testing automation in web applications: implementing DevOps practices in production (2017)
32. Subramanian, K.: Introducing the Splunk Platform. Practical Splunk Search Processing Language, pp. 1–38. Apress, Berkeley (2020)
33. Hasselbring, W., et al.: Industrial devops. In: 2019 IEEE International Conference on Software Architecture Companion (ICSA-C). IEEE (2019)
34. https://www.ul.ie/gps/devops-and-continuous-software-engineering-graduate-diploma
35. https://www.lyit.ie/CourseDetails/D202/LY_KDVOP_M/DevOps
36. https://www.torrens-international.com/programs/postgraduate-degrees/graduate-diploma-of-software-engineering-cloud-computing/
37. https://www.wileynxt.com/iit-roorkee-wiley-post-graduate-certication-in-cloud-computing-and-dev-ops
38. https://intellipaat.com/post-graduate-certification-cloud-and-devops/
39. https://www.iiitb.ac.in/executive-post-graduate-programme-in-software-development
40. https://www.simplilearn.com/pgp-devops-certification-training-cours
41. https://www.gminsights.com/industry-analysis/devops-market