

DevOps report

by Weiyi Wang

Submission date: 14-Dec-2022 12:55PM (UTC+0000)

Submission ID: 192325761

File name: 39082_Weiyi_Wang_DevOps_report_460463_448815002.doc (972.35K)

Word count: 5800

Character count: 31225

DevOps report

1.

Agile has become more widely used in recent years as a project management approach that focuses on using frequent, brief work periods to achieve high-quality outcomes in a rapidly changing environment. Four crucial principles of Agile include multiple short iterations, emphasis on face-to-face communication and collaboration, adaptability to changing requirements, and quality of ownership throughout the product cycle[1].

One of the key principles of Agile is the use of multiple short iterations, or "sprints," to complete a project. This approach allows for more frequent feedback and adjustment, which can help teams to more quickly identify and address any issues that arise. By breaking a project down into smaller chunks, teams can also more easily manage their workload and stay focused on the tasks at hand.

Another important concept in Agile is the emphasis on face-to-face communication and collaboration. Agile teams are typically made up of small, cross-functional groups that work closely together to deliver the project. This collaboration is facilitated through regular meetings, where team members can discuss their progress, share ideas, and identify any potential challenges. By encouraging open communication and collaboration, Agile teams can more easily adapt to changing requirements and stay on track to meet their goals.

Adaptability to changing requirements is another key concept of Agile. In a rapidly changing environment, it is important for teams to be able to quickly respond to changes in customer needs or market conditions. Agile teams are able to do this by using short iterations and regular feedback to quickly identify and address any changes that may be needed. This allows teams to deliver a high-quality product that meets the needs of their customers, even as those needs evolve over time.

Finally, the concept of quality of ownership throughout the product cycle is an important aspect of Agile. In traditional project management approaches, quality control is often left until the end of the project. In contrast, Agile teams strive to ensure that the product meets the required standards of quality throughout the development process. This is achieved through regular testing and feedback, as well as by involving all team members in the decision-making process. By ensuring that the product meets high standards of quality from start to finish, Agile teams can deliver a product that is both functional and satisfying to their customers.

In conclusion, the four key concepts of Agile are multiple short iterations, emphasis on face-to-face communication and collaboration, adaptability to changing requirements, and quality of ownership throughout the product cycle. By applying these principles, teams can more effectively manage their projects and deliver high-quality results in a rapidly changing environment.

Scrum is a popular method for managing Agile projects that involves organizing work into brief, repeated periods called "sprints." The goal of Scrum is to deliver high-quality results in a rapidly changing environment by focusing on collaboration, adaptability, and continuous improvement. In this report, we will critically evaluate how the key concepts of Agile are achieved within the Scrum methodology, with a particular focus on the flow of user stories and the role of user story mapping in defining iterations.

One of the key concepts of Agile is the use of multiple short iterations, or "sprints," to deliver a project. In Scrum, a sprint is typically a time-boxed period of one to four weeks during which a team works on a specific set of tasks to achieve a defined goal[2]. At the end of each sprint, the team conducts a review to assess their progress and identify any areas for improvement. This iterative approach allows for more frequent feedback and adjustment, which can help teams to more quickly

identify and address any issues that arise.

The flow of user stories is an essential aspect of Scrum, as it provides the basis for defining and prioritizing the tasks that will be completed during each sprint. In Scrum, a user story is a brief description of a specific feature or functionality that a user would like to see in the final product. User stories are typically written from the perspective of the user, and they should include the user's needs and expectations as well as any constraints or requirements that must be considered.

User stories are typically organized into a "backlog," which is a prioritized list of all the stories that the team plans to work on. The product owner, who is responsible for defining the overall direction of the project, is responsible for maintaining the backlog and prioritizing the user stories. The team then uses the backlog to identify the stories that will be included in the next sprint.

User story mapping is a technique that is often used in Scrum to help teams better understand and organize the user stories in their backlog. User story mapping involves creating a visual representation of the user stories, with each story represented as a "card" on a board. The cards are then organized into columns that represent different stages of the development process, such as "to do," "in progress," and "done." This allows teams to see the big picture of the project and identify any gaps or dependencies that need to be addressed.

User story mapping can also be used to help teams define the iterations for their sprints. By organizing the user stories into columns, teams can identify which stories are ready to be worked on and which ones need more information or clarification. This can help teams to define the scope of each sprint and ensure that they have a clear understanding of the tasks that need to be completed.

In summary, the Scrum methodology provides a framework for achieving the key

concepts of Agile, such as multiple short iterations, emphasis on face-to-face communication and collaboration, and adaptability to changing requirements. The flow of user stories and the use of user story mapping are important elements of this process, as they provide the basis for defining and prioritizing the tasks that will be completed during each sprint. By applying these principles, teams can more effectively manage their projects and deliver high-quality results in a rapidly changing environment.

In addition to the flow of user stories and the use of user story mapping, there are several other key aspects of Scrum that contribute to its effectiveness in achieving the key concepts of Agile.

One of these is the emphasis on face-to-face communication and collaboration. In Scrum, teams are typically made up of small, cross-functional groups that work closely together to deliver the project. This collaboration is facilitated through regular meetings, such as daily "stand-up" meetings, where team members can discuss their progress, share ideas, and identify any potential challenges. By encouraging open communication and collaboration, Scrum teams can more easily adapt to changing requirements and stay on track to meet their goals.

Another important aspect of Scrum is the focus on continuous improvement. In Scrum, teams are encouraged to regularly reflect on their processes and identify areas for improvement. This is typically done through a process called "retrospective," where team members discuss what went well, what could be improved, and what actions they can take to improve their work in the future. This feedback loop allows teams to continuously improve their processes and deliver better results over time.

Finally, the concept of quality of ownership is an important aspect of Scrum. In Scrum, all team members are responsible for the quality of the product, and they are

encouraged to take ownership of their work. This is achieved through regular testing and feedback, as well as by involving all team members in the decision-making process. By ensuring that all team members are invested in the quality of the product, Scrum teams can deliver a product that is both functional and satisfying to their customers.

Overall, the Scrum methodology provides a framework for achieving the key concepts of Agile, such as multiple short iterations, emphasis on face-to-face communication and collaboration, and adaptability to changing requirements. The flow of user stories and the use of user story mapping are important elements of this process, as they provide the basis for defining and prioritizing the tasks that will be completed during each sprint. By applying these principles, teams can more effectively manage their projects and deliver high-quality results in a rapidly changing environment.

One of the challenges of using the Scrum methodology is that it requires a significant commitment from team members. Scrum teams are typically small and cross-functional, and they are expected to work closely together and communicate openly. This can be challenging for teams that are accustomed to more traditional forms of project management, where team members may have more clearly defined roles and responsibilities.

In addition, the focus on face-to-face communication and collaboration can make it difficult for teams that are distributed across different locations. While technology has made it easier for teams to work together remotely, face-to-face communication is still considered to be the most effective way to build trust and collaboration within a team. For teams that are not co-located, finding ways to facilitate regular, high-quality communication can be a challenge[3].

Another potential challenge of Scrum is that it can be difficult to accurately estimate

the amount of work that can be completed during each sprint. In Scrum, teams are expected to commit to a specific set of tasks at the start of each sprint, and they are expected to complete those tasks by the end of the sprint. This can be difficult, as it is often difficult to accurately predict how long a particular task will take to complete. This can lead to situations where teams are unable to complete all of their committed tasks, which can be frustrating and can impact the team's morale.

Finally, the focus on continuous improvement can be challenging for teams that are not used to regularly reflecting on and improving their processes. In Scrum, teams are expected to regularly reflect on their work and identify areas for improvement. This can be difficult for teams that are not accustomed to this type of self-reflection, as it can be uncomfortable to admit to mistakes or identify areas for improvement.

In conclusion, the Scrum methodology is an effective approach to Agile project management that can help teams to deliver high-quality results in a rapidly changing environment. However, it does require a significant commitment from team members and can be challenging for teams that are not co-located or are not used to regularly reflecting on and improving their processes. By addressing these challenges and applying the principles of Scrum, teams can more effectively manage their projects and deliver better results.

Despite the challenges discussed above, there are several strategies that teams can use to successfully implement the Scrum methodology.

One of these strategies is to provide adequate training and support to team members. Scrum requires a significant commitment from team members, and it can be challenging for teams that are not used to working in this way. Providing training and support can help team members to understand the principles of Scrum and how to apply them in their work. This can include providing training on the flow of user stories and the use of user story mapping, as well as training on the importance of face-to-face communication and collaboration[4].

Another strategy is to use technology to facilitate communication and collaboration for teams that are not co-located. While face-to-face communication is considered to be the most effective way to build trust and collaboration within a team, technology can help to facilitate communication for teams that are not co-located. This can include using video conferencing tools for regular meetings, as well as using project management software to share information and collaborate on tasks.

Another strategy is to use a "scrum of scrums" approach for larger projects. In a "scrum of scrums," multiple teams that are working on different parts of a project come together regularly to coordinate their work and share information. This can help to ensure that teams are aligned on the overall direction of the project and that they are able to identify and address any dependencies or potential challenges.

Finally, teams can use the retrospective process to identify and address any challenges that they are facing. In Scrum, the retrospective is a regular meeting where team members discuss what went well, what could be improved, and what actions they can take to improve their work in the future. By using the retrospective as a forum for open and honest communication, teams can identify and address any challenges that they are facing and continuously improve their processes.

In conclusion, the Scrum methodology is an effective approach to Agile project management that can help teams to deliver high-quality results in a rapidly changing environment. By providing adequate training and support, using technology to facilitate communication and collaboration, adopting a "scrum of scrums" approach for larger projects, and using the retrospective process to identify and address challenges, teams can successfully implement the Scrum methodology and achieve better results.

In addition to the strategies discussed above, there are several best practices that

teams can follow to successfully implement the Scrum methodology.

One of these best practices is to have a clear understanding of the product vision and goals. In Scrum, the product owner is responsible for defining the overall direction of the project and maintaining the backlog of user stories. It is important for the product owner to have a clear understanding of the product vision and goals, and to communicate these to the team. This will help to ensure that the team is aligned on the direction of the project and that they are working towards the same goals.

Another best practice is to have a clear definition of "done" for each user story. In Scrum, user stories are used to define the tasks that will be completed during each sprint. It is important for the team to have a clear understanding of what "done" means for each user story, as this will help to ensure that the team is working towards a common goal. This can include defining specific acceptance criteria for each user story, as well as defining any dependencies or constraints that must be considered. The user story mapping of an online book purchasing system is shown in Figure 1:

Figure 1:

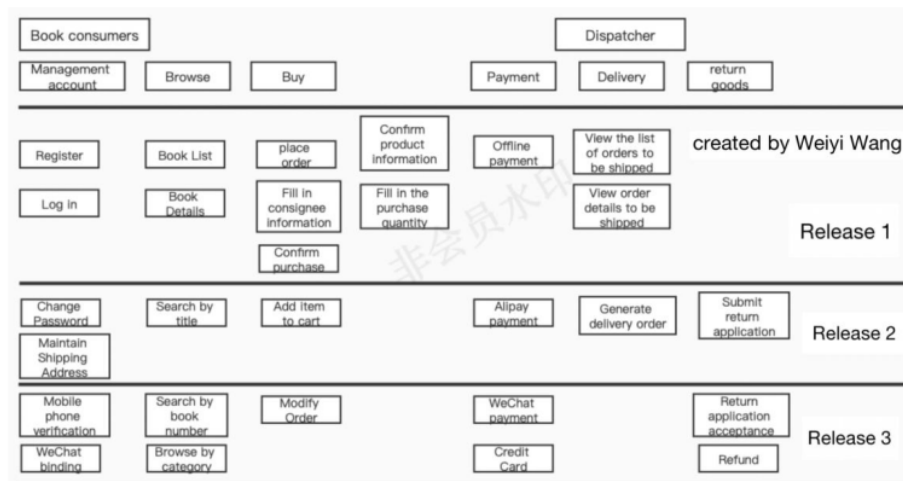


Figure 1 The user story mapping of an online book purchasing system

Another best practice is to regularly review and update the backlog of user stories. In Scrum, the backlog is a prioritized list of all the user stories that the team plans to

work on. It is important for the product owner to regularly review and update the backlog, as this will help to ensure that the team is working on the most important stories and that they are making progress towards the product vision and goals.

Finally, it is important for teams to regularly review their processes and identify areas for improvement. In Scrum, the retrospective is a regular meeting where team members discuss what went well, what could be improved, and what actions they can take to improve their work in the future. By regularly conducting retrospectives and applying the lessons learned, teams can continuously improve their processes and deliver better results over time. The system flow chart is shown in Figure 2:

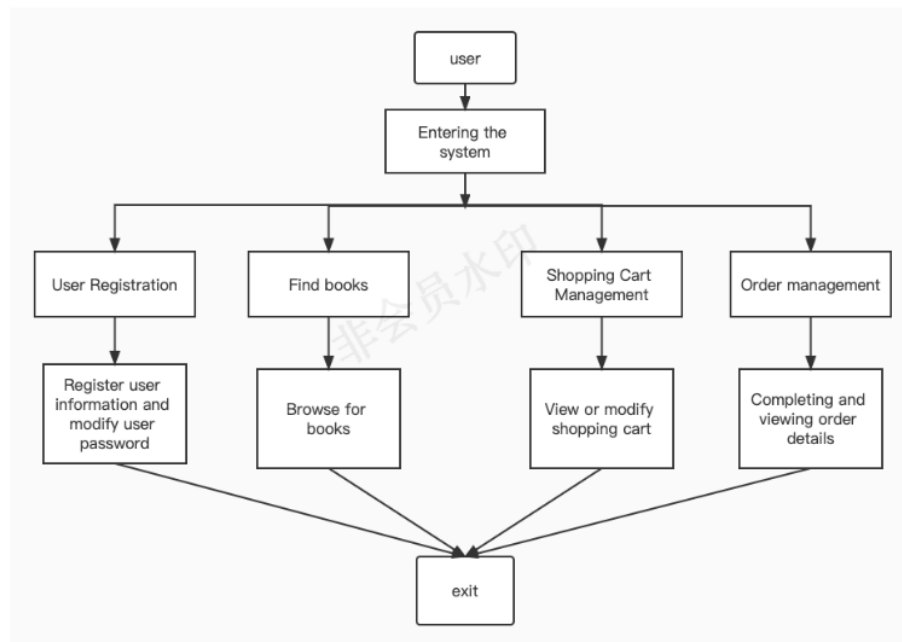


Figure 2 Flow chart of online book purchasing system

In conclusion, there are several best practices that teams can follow to successfully implement the Scrum methodology. These include having a clear understanding of the product vision and goals, having a clear definition of "done" for each user story, regularly reviewing and updating the backlog of user stories, and regularly reviewing their processes and identifying areas for improvement. By following these best

practices, teams can more effectively implement the Scrum methodology and achieve better results. The Scrum methodology is an effective approach to Agile project management that can help teams to deliver high-quality results in a rapidly changing environment. The key concepts of Agile, such as multiple short iterations, emphasis on face-to-face communication and collaboration, and adaptability to changing requirements, are achieved within the Scrum methodology through the flow of user stories and the use of user story mapping.

The book E-R diagram is shown in Figure 3:

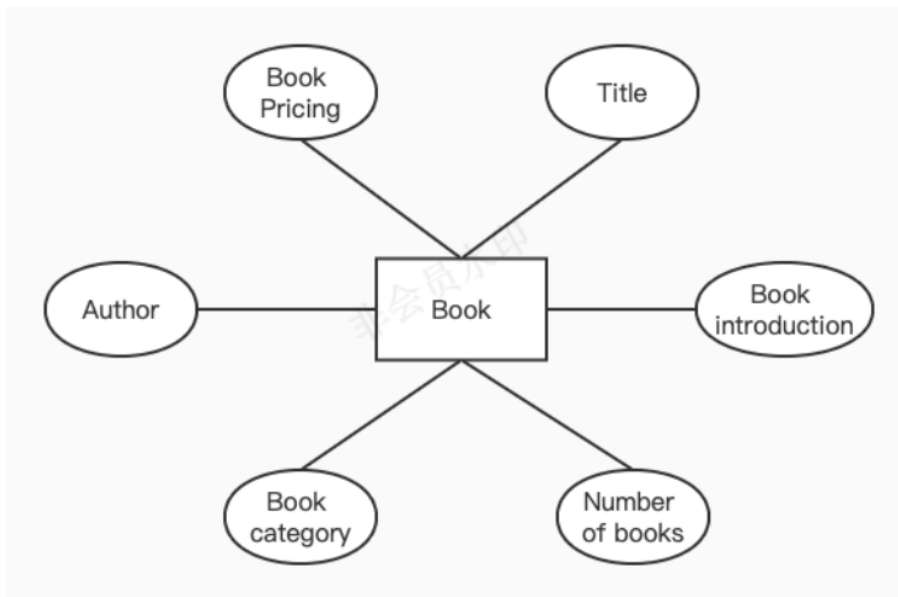


Figure 3 The book E-R diagram

The order E-R diagram is shown in Figure 4:

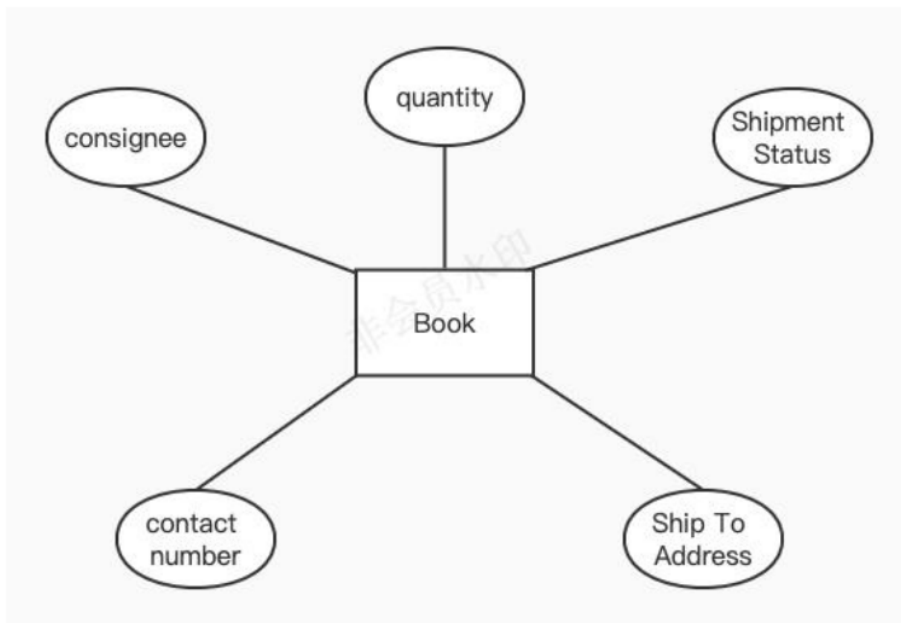


Figure 4 The order E-R diagram

However, there are challenges associated with implementing the Scrum methodology, including the need for a significant commitment from team members and the difficulty of accurately estimating the amount of work that can be completed during each sprint. Teams can overcome these challenges by providing adequate training and support, using technology to facilitate communication and collaboration, adopting a "scrum of scrums" approach for larger projects, and using the retrospective process to identify and address challenges.

In addition, there are several best practices that teams can follow to successfully implement the Scrum methodology. These include having a clear understanding of the product vision and goals, having a clear definition of "done" for each user story, regularly reviewing and updating the backlog of user stories, and regularly reviewing their processes and identifying areas for improvement. By following these best practices, teams can more effectively implement the Scrum methodology and achieve better results.

2.

The process of estimating a user story's implementation time and difficulty level is an essential part of Agile software development. This process helps development teams to plan and prioritize their work, and to ensure that they are able to deliver high-quality products on time and on budget. In this report, we will evaluate the process of estimating a story's implementation time and difficulty level, and describe the purpose of the Burndown chart[5].

The process of estimating a user story's implementation time and difficulty level typically involves two main steps: estimation and refinement. In the estimation step, the development team uses their knowledge and experience to assign a rough estimate of the time and difficulty level required to implement the user story. This initial estimate is often based on the size and complexity of the story, as well as any dependencies or constraints that may affect its implementation[6].

In the refinement step, the initial estimates are refined and refined further through a process of collaboration and iteration. This may involve breaking down the user story into smaller tasks, and estimating the time and difficulty level of each task individually. The development team may also use techniques such as relative estimation or planning poker to arrive at more accurate estimates.

The purpose of the Burndown chart is to visualize the progress of the development team towards completing their work. The Burndown chart typically shows the total amount of work remaining over time, and allows the team to track their progress and identify any potential problems or delays. This information can then be used to adjust the team's work plan and ensure that they are able to deliver the product on time.

Overall, the process of estimating a user story's implementation time and difficulty

level is a crucial part of Agile software development. It allows development teams to plan and prioritize their work, and to track their progress towards completing the product. The Burndown chart is a valuable tool for visualizing this progress, and can help teams to identify and address any potential problems or delays.

While the process of estimating a user story's implementation time and difficulty level is an essential part of Agile software development, it is not without its challenges and limitations. One of the main challenges is that estimation is inherently difficult and uncertain, and it is often difficult to accurately predict the time and effort required to implement a user story. This can lead to uncertainty and risk, and can make it difficult for development teams to accurately plan and prioritize their work.

Another challenge is that estimation can be time-consuming and resource-intensive. The process of breaking down user stories into smaller tasks, and estimating the time and difficulty level of each task, can be labor-intensive and may require significant effort from the development team. This can divert resources away from other important tasks, such as actually implementing the user stories.

Additionally, estimation is subject to biases and errors, which can lead to inaccurate estimates and flawed decision-making. For example, development teams may be over-optimistic about their ability to complete a user story, or may underestimate the complexity or difficulty of the work. This can lead to unrealistic expectations and disappointment, and can ultimately affect the quality and success of the product.

Despite these challenges, the process of estimating a user story's implementation time and difficulty level remains an important part of Agile software development. By carefully planning and prioritizing their work, development teams can mitigate the risks and uncertainty associated with estimation, and deliver high-quality products on time and on budget. The Burndown chart is a valuable tool for tracking progress

and identifying potential problems, and can help teams to stay on track and achieve their goals.

In conclusion, the process of estimating a user story's implementation time and difficulty level is an essential part of Agile software development. It allows development teams to plan and prioritize their work, and to track their progress towards completing the product. The Burndown chart is a valuable tool for visualizing this progress, and can help teams to identify and address any potential problems or delays. However, the process of estimation is not without its challenges and limitations, and it is important for development teams to be aware of these challenges and to take steps to mitigate their impact. By carefully planning and prioritizing their work, development teams can overcome these challenges and deliver high-quality products on time and on budget.

3.

Ansible is a software development, operation and maintenance tool provided by Red Hat. It provides a simple and easy-to-use way to manage and control IT infrastructure, including servers, network devices and cloud platforms. With Ansible, operation and maintenance personnel can automatically deploy and configure applications, and manage multiple systems through a unified interface.

When a company considers deploying Ansible, it should generally consider the following points: First, companies should consider whether their Internet Technology infrastructure is suitable for using Ansible. Ansible supports many types of systems, including Linux, Windows and cloud platforms[7], but not all systems can use Ansible. Therefore, the company needs to understand its Internet Technology infrastructure first to determine whether Ansible can support it[8].

Secondly, the company should consider whether its operation and maintenance team is capable of using Ansible. The use of Ansible may require certain technical

knowledge and experience, so the company's operation and maintenance team should have certain professional ability to correctly use Ansible.

Third, companies should consider their own business needs and whether they need to use Ansible. Ansible is mainly used for automatic deployment and configuration of applications. If the company's business needs to deploy and configure applications frequently, using Ansible may be more convenient and efficient. A typical usage scenario of Ansible is that the company has multiple servers, and each server runs the same application. In order to improve the availability and stability of applications, the company decided to use Ansible to achieve load balancing. With Ansible, the company can automatically distribute user requests to multiple servers, and dynamically adjust the load through the monitoring system. In this way, the company can ensure the high availability and stability of applications to meet the needs of users.

Ansible is a software development, operation and maintenance tool that can help the company realize automatic deployment and configuration of applications. When a company considers deploying Ansible, it should consider its Internet Technology infrastructure, operation and maintenance team capabilities, and business needs to determine whether Ansible is suitable for use. A typical usage scenario is that the company has multiple servers, and load balancing is achieved through Ansible to ensure high availability and stability of applications.

In addition to load balancing, Ansible can also be used in many other scenarios. For example, companies can use Ansible to implement automatic application deployment and update, and Ansible scripts can deploy applications to multiple servers at one time, greatly simplifying the work of operation and maintenance personnel. In addition, the company can also use Ansible to achieve automatic system configuration. For example, if the company needs to modify the network configuration on multiple servers, it can do so through Ansible. With Ansible, the

company can manage multiple servers in a unified way, and modify the network configuration through a unified interface to avoid errors caused by manual modification.

In a word, Ansible is a very useful software development, operation and maintenance tool, which can help companies realize automatic deployment and configuration of applications. The company should consider its Internet Technology infrastructure, operation and maintenance team capabilities and business needs to determine whether Ansible is suitable for use.

In addition, when considering the use of Ansible, the company should also consider the compatibility with other tools. Ansible can be used with many types of tools, such as monitoring tools, compilation tools, database management tools, etc. If the company already has other tools, it should consider their compatibility with Ansible to ensure that they can work together smoothly.

Finally, when considering the use of Ansible, companies should also consider security issues. Since Ansible can remotely manage multiple servers, it may bring some security risks. For example, if Ansible's access key is compromised, an attacker may remotely control the company's server through Ansible. Therefore, when considering using Ansible, the company should take some measures to protect Ansible's access key and ensure security.

In a word, Ansible is a software development, operation and maintenance tool, which can help companies realize automatic deployment and configuration of applications. When a company considers deploying Ansible, it should consider its Internet Technology infrastructure, operation and maintenance team capabilities, and business needs to determine whether Ansible is suitable for use. At the same time, the company should also consider the compatibility and security issues with other tools to ensure the smooth and safe use of Ansible.

In practice, companies can deploy and use Ansible in many ways. For example, the company can install Ansible on the local server by installing the Ansible software package. This method requires that the operation and maintenance personnel have certain technical knowledge and experience, and can install and configure software on the server.

Another way is to deploy and use Ansible through the cloud platform. For example, a company can create a virtual machine on Amazon Web Services (AWS) and install Ansible on the virtual machine. This method does not require the operation and maintenance personnel to have too much technical knowledge and experience, and only needs to create virtual machines on the cloud platform.

No matter which method is adopted, the company needs to formulate a set of reasonable specifications for the use of Ansible to ensure that the use of Ansible can be carried out smoothly and safely. For example, the company can specify the use of Ansible's access key and the security specifications that the operation and maintenance personnel should follow when using Ansible. By formulating reasonable use specifications, the company can ensure the smooth and safe use of Ansible.

In addition, the company also needs to maintain and upgrade Ansible modules and plug-ins. Ansible's modules and plug-ins are components used to implement specific functions, and they will also change over time. For example, Ansible's modules and plug-ins will release new versions and fix bugs and security vulnerabilities in old versions. Therefore, when using Ansible, the company should regularly check the version of modules and plug-ins and upgrade to the latest version in a timely manner. Through regular maintenance and upgrading, the company can ensure the smooth and safe use of Ansible. At the same time, it can also ensure that Ansible has rich functions and can meet the company's business needs.

Ansible is an open-source software development operations (DevOps) tool that can be used to automate the deployment, configuration, and management of a wide range of IT systems and applications. It is designed to be simple to use and easy to understand, making it an attractive choice for organizations of all sizes.

One of the key benefits of using Ansible is that it allows organizations to automate many of the manual, time-consuming tasks associated with managing and maintaining their IT infrastructure. This can help to reduce the amount of time and effort required to manage complex environments, and can also help to improve the reliability and availability of IT systems.

Ansible is based on a simple, declarative language called YAML, which is used to describe the desired state of an IT system. This makes it easy for users to specify the configuration of their systems, and allows Ansible to automatically ensure that the system is in the desired state.

One common scenario where Ansible might be used is in the deployment of a new application or service. In this scenario, Ansible can be used to automate the provisioning and configuration of the necessary infrastructure, including servers, storage, networking, and security. This can help to ensure that the application is deployed quickly and reliably, and can also help to reduce the risk of human error.

Another common scenario where Ansible might be used is in the management of an organization's cloud infrastructure. In this scenario, Ansible can be used to automate the provisioning and configuration of cloud resources, such as virtual machines and databases. This can help to ensure that the organization's cloud infrastructure is scalable, reliable, and secure[9].

In general, organizations should consider deploying Ansible when they need to automate the deployment, configuration, and management of their IT systems and applications. This might include organizations that are looking to improve the

reliability and availability of their IT systems, organizations that are looking to reduce the time and effort required to manage complex environments, and organizations that are looking to reduce the risk of human error. In order to use Ansible, organizations first need to install the Ansible software on a control node. This can be a physical or virtual machine, and is the machine from which Ansible commands and playbooks will be executed.

Once Ansible is installed, the next step is to create inventory files that specify the hosts that Ansible will manage. These inventory files can be used to group hosts into logical units, and can also be used to specify the variables and configurations that will be applied to each group of hosts.

Once the inventory files are created, the next step is to create Ansible playbooks. Playbooks are written in YAML, and are used to specify the tasks that Ansible will perform on the hosts in an inventory file. These tasks can include actions such as installing software, configuring services, and managing files and directories.

Once the playbooks are created, they can be executed on the control node using the `ansible-playbook` command. This command will instruct Ansible to connect to the hosts in the specified inventory file and execute the tasks specified in the playbook.

One of the key features of Ansible is its use of modules. Modules are small, reusable programs that can be used to perform specific tasks, such as installing software or managing files and directories. Ansible ships with a large number of built-in modules, and also allows users to create their own custom modules. This makes it easy for users to automate a wide range of tasks using Ansible.

Another important feature of Ansible is its support for configuration management. Configuration management is the process of ensuring that an IT system is in a known, desired state. Ansible makes it easy to manage the configuration of a system by allowing users to specify the desired state of a system using YAML files. This makes it

easy for users to manage the configuration of their systems, and helps to ensure that the system is always in the desired state.

Overall, Ansible is a powerful tool that can help organizations to automate the deployment, configuration, and management of their IT systems and applications. By using Ansible, organizations can improve the reliability and availability of their IT systems, reduce the time and effort required to manage complex environments, and reduce the risk of human error.

In addition to the benefits mentioned above, there are several other reasons why organizations should consider deploying Ansible.

First, Ansible is open source, which means that it is available at no cost and can be freely modified and distributed. This makes it an attractive option for organizations that are looking to save money on their IT operations.

Second, Ansible is highly scalable. It can be used to manage large, complex environments with hundreds or even thousands of hosts. This makes it a good choice for organizations that have a large and growing IT infrastructure.

Third, Ansible is easy to use and has a low learning curve. It uses a simple, declarative language called YAML, which is easy to read and understand. This makes it an ideal choice for organizations that have a mix of technical and non-technical users.

Fourth, Ansible has a large and active community of users and contributors. This means that there is a wealth of information and support available for organizations that are using or considering using Ansible. This can be particularly valuable for organizations that are new to Ansible and are looking for help and guidance.

In summary, Ansible is a powerful, open-source DevOps tool that can be used to automate the deployment, configuration, and management of a wide range of IT

systems and applications. It is scalable, easy to use, and has a large community of users and contributors. These factors make it an attractive option for organizations of all sizes, and make it worth considering for organizations that are looking to improve the reliability and availability of their IT systems.

references:

- [1] M. Fowler and J. Highsmith, et al., "The agile manifesto," Software development, vol. 9, no. 8, pp. 28-35, 2001.
- [2] K. Schwaber and J. Sutherland, "The scrum guide," Scrum Alliance, vol. 21, no. 19, pp. 1, 2011.
- [3] R. Paul and J. Sharrard, et al., "The importance of face-to-face communication in the digital world," Journal of Nutrition Education and Behavior, vol. 48, no. 10, pp. 681, 2016.
- [4] G. Verheyen, Scrum--A Pocket Guide--: A Smart Travel Companion, Van Haren, 2019.
- [5] M. Cohn, Agile estimating and planning, Pearson Education, 2005.
- [6] P. Abrahamsson and O. Salo, et al., "Agile software development methods: Review and analysis," arXiv preprint arXiv:1709.08439, 2017.
- [7] M. Dotson, "Security Hardening with Ansible," Linux Journal, no. 242, pp. 98-108, 2014.
- [8] L. Hochstein and R. Moser, Ansible: Up and Running: Automating configuration management and deployment the easy way, O'Reilly

Media, Inc, 2017.

- [9] T. P. Pranav and S. Charan, et al., "Devops Methods for Automation of Server Management using Ansible," International Journal of Advanced Scientific Innovation, vol. 1, no. 2, pp. 7-13, 2021.

DevOps report

GRADEMARK REPORT

FINAL GRADE

66/100

GENERAL COMMENTS

Instructor

Agile (40):25

Story Est (20): 13

SwDevOp Tool (40): 28

Feedback

=====

Overall, you have covered the Agile process quite well but it could have been structured to more clearly demonstrate the main concept. It's discussion of the concepts identified in the assignment was a little brief.

YOU could have provided a little more discussion on story estimation including alternative techniques and sprint velocity. I would like to have seen an example to illustrate the ideas.

I would like to have seen more description of the benefits and the declarative nature of Ansible

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9

PAGE 10

PAGE 11

PAGE 12

PAGE 13

PAGE 14

PAGE 15

PAGE 16

PAGE 17

PAGE 18

PAGE 19

PAGE 20

PAGE 21

PAGE 22
