

Assignment

by Xu Zhang

Submission date: 14-Dec-2022 02:25PM (UTC+0000)

Submission ID: 192345202

File name: 40921_Xu_Zhang_Assignment_460463_1776458347.pdf (454.14K)

Word count: 5150

Character count: 28032

Assignment

Name: Xu Zhang

Student number: 2200467

Major: Software Engineering and Artificial Intelligence
programme

Email: 2200467@student.uwtsd.ac.uk

Abstract—Nowadays, Traditional software development methods (such as the waterfall model) have been superseded with cutting-edge agile development models, which place a greater emphasis on quality, service, and efficiency. This paper focuses on an agile method Scrum, analyzing the main concepts of Scrum software development method, event evaluation, practical development operation tools application, etc. It is found that this agile method has improved and enhanced the cost, quality, customer and employee satisfaction in software project application. Compared with the inefficient traditional development model, the scrum agile development method is more likely to promote team motivation and improve project productivity, and has some guidance for efficient software development model. And analyzed DevOps a tool to automate software development and automated operation and maintenance, which mainly works on the existence of disconnection between development team and operation and maintenance team. Finally, a platform of DevOps, Ansible, is applied to a practical scenario to further elaborate on the automated operation and maintenance approach.

Keywords: Agile development, Scrum, DevOps, Ansible

I. INTRODUCTION

In the past, a waterfall model [1] was used for managing a large software project. This model required detailed requirements analysis and system design, but as the software industry has evolved, this cumbersome approach has been gradually abandoned by developers. The higher the completion of a software project, the greater the cost of making changes [2]. Therefore, there was an urgent need for a solution to the problem of unclear requirements at the beginning of a software product concept and frequent changes in the design process. The concept of agile was first formally introduced in [3], which also published the "Manifesto for Agile Software Development", identifying four key values: Delivering a usable software product is the primary objective; The use of both iterative and incremental development models; Timely communication and collaboration among all parts of the project team; Using comments during the development phase to continuously improve the product. An agile development approach with these concepts can be adapted to software development projects with rapidly changing requirements. In a global changing business environment where requirements, knowledge, and technology are always in a state of flux, agile development allows for rapid allocation of resources to capture business opportunities [4].

Scrum, introduced in 1993, is one of the most widely used practices in the agile development model [5]. Scrum is not a process or a technique for developing products, but a framework in which processes and techniques can be applied to efficiently and creatively deliver the highest value

products possible. Scrum employs an incremental, iterative methodology to minimize risk and maximize predictability. Transparency, review, and alignment are the three pillars of empirical processes that underpin the implementation of each empirical control process. scrum software development is used to respond to frequently changing user requirements by breaking the product development process into iterative phases. It also establishes a communication platform between developers and users to achieve efficient and high-quality delivery of software products. The Agile Manifesto is mapped in Scrum as 1) Choosing a short development cycle to satisfy customers by consistently delivering valuable software early; 2) Being flexible even in late phases of development and mastering flexibility in the Agile process for the competitive advantage of the customer; 3) Business people and developers must collaborate, and face-to-face communication is the most effective and efficient method of information delivery; 4) Teams regularly reflect on how they can be more effective and therefore adjust their own behavior and performance; 5) Individuals are motivated to build projects with them as the core, providing the environment and support needed, complemented by trust, in order to achieve their goals.

Scrum software development consists of four main concepts:

1. There are multiple short iterations of the process, Scrum development process uses iterative development, the project is divided into multiple iteration cycles according to the size of the project, each iteration cycle should not be too long, usually controlled between 1-4 weeks. Each iteration cycle involves a full set of processes such as requirements communication, requirements analysis, requirements design, code writing, testing, delivery, etc. After an iteration is completed, a sprint review meeting is held to demonstrate the features to customers and collect customer feedback.

2. Scrum emphasizes face-to-face team communication and collaboration, and agile teams emphasize human-centeredness, mutual equality and self-management among team members. The Scrum Director is positioned as a servant leader, providing an agile development environment for developers. Self-managed agile teams have a good, harmonious and trusting working atmosphere, work morale and efficiency are greatly improved, and the maximum energy of the team is brought into play.

3. The Scrum approach is able to adapt to changing requirements, rather than the traditional development approach that requires all the feature list of a software project to be developed and delivered to the customer at once. Scrum development is a joint effort with customers to confirm product feature priorities, so as to develop a sprint development plan, prioritize the completion of a list of features with high market value and clear requirements and deliver them, prioritize the delivery of partially available features for operation after the completion of each sprint cycle, and subsequently conduct incremental deployments

based on the previous completed versions, and finally deliver a fully functional software product that meets customer expectations.

4. Scrum development includes users as an important part of the agile team, and They contribute significantly to the whole process of software product development. In the product requirement analysis and product design activities, customers are fully involved and provide timely feedback, which largely avoids deviations in the project development process. In addition, Scrum's sprint planning meetings and sprint review meetings provide a way for customers to provide comments and feedback, which greatly facilitates faster and better delivery of software products, and also improves the overall quality of the products.

The Scrum software development process is shown in Fig. 1.

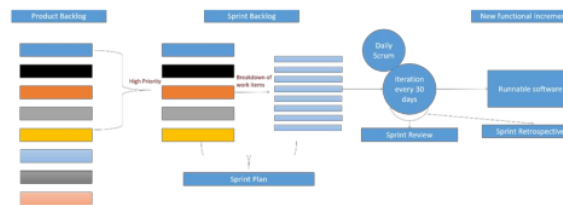


Fig. 1. Scrum Software Development

Regarding the explanation of some terms in the diagram: 1) Sprint Plan: The product owner presents the highest priority product to the team, and the team asks questions about the content, purpose, meaning, and intent of the backlog, and ultimately the team plans the arrangement of this sprint; 2) Sprint Review: Team members present product development in the Sprint cycle to product owners and other relevant people; 3) Daily Scrum: In the daily meeting, each member needs to answer three questions: what did you do today? what will you do tomorrow? what difficulties did you encounter in the realization of the project? 4) Sprint Retrospective: The ScrumMaster will guide the team in making changes to the development process within the Scrum process framework and practices to make it more effective for the next Sprint.

II. RELATED METHODS

A. Agile estimation

When a team develops a product or develops a project, it needs to divide a period of approximate time points that represent the progress of the workload completion. So as with traditional development models, we need to estimate the workload of what we need to do before we start working.

Compared to traditional estimation, agile estimation has the following processes:

(1) In the Scrum development process, the team shares responsibility and collectively commits to each Sprint, so for workload estimation agile teams use a collective estimation approach. The team determines the total workload through a group game, usually based on a  1 game. Each team member has different values of playing cards and gives his or her own cards for different entries in the backlog for workload estimation, and after counting and discussing collectively to reach a consensus.

(2) Relative estimates are generally used to determine the difficulty level of a story point. When agile estimation is done, instead of estimating absolute times and periods, the

size and relative values are estimated, the size of the estimate is expressed in multiples. Using story points as a unit of measurement, a minimum story is defined as a story point by using it as a reference benchmark. The other stories are compared with the benchmark, and having several story points is determined by how many times it is the benchmark.

(3) The speed of each sprint team needs to be recorded. The team rate is the number of story points a Scrum team actually completes in a Sprint, and the team rate allows you to know how fast the team is doing. For a newly started project or product development, or a new team, there is no initial speed, we can do 1-2 Sprint to measure a speed as the initial speed. During the Sprint execution, we have to record the speed of each Sprint to make reference for future plans. Finally get the average rate of a team to complete a sprint.

The flow of estimating story time is shown in the Fig. 2.

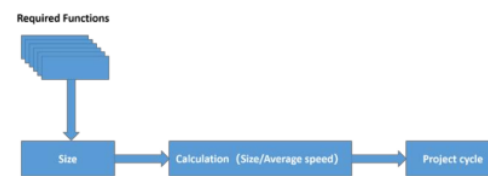


Fig. 2. Estimating story time

B. Burn down chart

The role of the burn down chart is equally important in the agile development process. First described by Ken Schwaber on burn down chart [6], visualize the work that has to be done in the burn chart to finish the project. The burn down chart has a Y-axis (work) and an X-axis (time). A downward slope that "burns out" to zero once the remaining work is finished is the ideal shape for the graph. The burn-up chart is a public view of work progress to project team members and business owners. Two forms of burn-up charts exist, the Sprint burn-up chart is used to show the remaining work in an iteration, and the Product burn-up chart is used to illustrate the remaining work for the entire project.

In general, there are two curves in the burn down chart, the planned curve and the actual curve. A straight line joins the start and finish locations to form the plan curve. The end point of the plan curve crosses the X-axis, signifying that there is no longer any work to be done, and it indicates the total amount of estimated tasks that need to be finished. However, since the planning curve is based on estimates and the actual situation is always variable, it is not very accurate. Another curve, the actual curve, is used to show the actual remaining work in the project or iteration. The planned remaining work and the actual remaining work are equal at the beginning of the project or iteration, but as the project or iteration moves forward, the actual remaining work curve will veer above and below the planned work line. If the actual work line is higher than the planned curve, it means that there is more work left than expected. However, if the actual curve is lower than the planned curve, it means that there is less work remaining than expected and the project is progressing faster than the established plan.

An example of a burn down chart is shown in Fig. 3 [7].



Fig. 3. Burn down chart

The burn down chart's most essential advantage is that it offers up-to-date reporting on the progress and current status of the project, and a visual representation of this crucial information makes sure that everyone is in agreement on the progress. Additionally, keeping the burn down chart visible encourages team members to address any problems in advance and keeps them all engaged in the project. Burn down chart should serve as a focal point for the development team, always generating relevant discussions about the project and its progress. Of course, burn down chart also have limitations such as not being able to present all the information and relying on accurate forecasts. Therefore, using burn down chart wisely is the best way to complete software projects.

C. DevOps

After the advent of agile development methods, around 2008, it was discovered that there were some flaws at both ends of the software development process (development and operations) that had to be figured out in order to reduce cycle times [8]. The shortcomings are mainly focused on the development team's constant pursuit of change and will take continuous delivery as the goal; while the operations team mainly pursue the goal of maintaining stability when running in reality. The inconsistency in the pursuit of goals between the two departments has led to thicker and thicker barriers between them in the long run. At this time a processing method DevOps emerged to solve the problem of barriers between Dev (development team) and ops (operations team) very well. The original DevOps model, shown in Fig. 4, was used to break down the development and operations dilemma.

Flickr company first demonstrated in 2009 how to make software project process management efficient by removing the barriers between development (Dev) and operations (Ops) teams, working together with tools, platforms and cultural changes [9]. This shows that the barriers between the two teams emerge because of the lack of a common deployment and operational view of the system, two key aspects [10]. As the development model evolves, DevOps now addresses not just the development and operations barriers, but the end-to-end collaboration throughout the project process phase, as shown in Fig. 5.



Fig. 4. Original DevOps model



Fig. 5. DevOps End-to-End Model

At the heart of DevOps is cross-departmental collaboration and automation of people [11]. Its three pillars are people, process and platform, as shown in the Fig. 6. People and process form the culture, people and platform form the skill, platform and process form the tool, this shows that the three pillars are interconnected, leading to better collaboration.

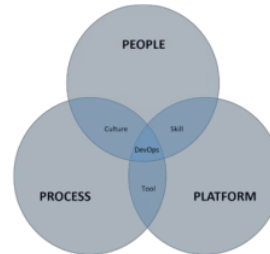


Fig. 6. Three Pillars of DevOps

This article describes DevOps from the following three perspectives.

(1) From the standpoint of a platform, DevOps, based on Lean and Agile, means end-to-end automation in software development and delivery, a process that tightly integrates developers with operators [12]. It is a novel tool for automating software development and operations tasks, including both technical and non-technical processes. By releasing new software frequently and automatically, it can assist businesses with huge software projects become more responsive to consumer requests. The DevOps platform relies on its robustness and stability to enable frequent release of different versions, which reduces the risks associated with deployment and allows for timely feedback on changes to the software application and its configuration (including the environment). Often, this fast and timely feedback is critical for the development of new information that needs to be selected later in the software development process [13]. Because the DevOps all-in-one platform enables different parts of development and operations, even development and testing, testing and product, to break down communication barriers, This allows for better collaboration between teams and ultimately on-time project delivery. In the ever-evolving corporate landscape of today, where most customers demand faster turnaround times and higher quality, the DevOps platform has become a fundamental requirement to meet customer needs. DevOps is an all-in-one platform with numerous tools, such as automated testing tools, JIRA for project management, GIT for development, etc. DevOps bridges the communication gap between software design, development, testing, and operations by leveraging a series of tool chains.

(2) From a process perspective, a process that is non-redundant and valuable helps team members develop a sense of quality and improves overall product integrity. Building on the history of past mistakes in designing software and iterating on the process can reduce the chance of design mistakes while also providing greater clarity on the project goals.

(3) From a people perspective, people are the ones who change, execute, and implement DevOps. As mentioned above, people and processes form the culture, and people and platforms form the skills. Especially in software innovation, people are essentially the culture, so people need

to be trained and empowered to understand the concept of DevOps, master the use of the platform, and gradually form an enterprise-specific culture, so that they can continue to deliver effective value.

In summary, in order to solve the enterprise's improvement of the entire software delivery process, maintain high throughput, high delivery quality, and high stability, and achieve the delivery purpose of quick and effective user value, DevOps is generally to start from the whole picture of software project process management, continuous optimization of the process, overall innovation of the system architecture, and continuous upgrading of the organizational culture. Its basic principles are fast flow, fast feedback and continuous learning, the main advantages of which are 1) efficient and continuous delivery, shortening the software development cycle; 2) fast access to feedback, improving software quality and stability; 3) improving the company culture, promoting continuous learning and communication. Of course, DevOps also has some shortcomings, before the implementation of DevOps management need to invest too much in tools, too dependent on the existing toolset, etc.

D. DevOps application platform

Many platforms exist for DevOps to be applied, such as drone spinnaker ansible terraform, etc. This article focuses on automating DevOps operations and maintenance based on the ansible platform. On its website, Ansible is described as follows: "Ansible is a radically simple IT automation engine" [14], this shows the simplicity of Ansible tools. Ansible is a Python-based cross-platform automated operations and maintenance solution that combines the benefits of other server cluster operations, batch system setup, batch program deployment, and batch run commands [15]. Ansible works on a module basis and has no batch deployment capability per se. The modules that Ansible works on are what actually provide batch deployment, and Ansible only acts as a framework. It has a rich set of built-in modules, such as command, shell, cron, yum, service, and many others, and new modules are being added rapidly as Ansible grows. ansible also has an open API interface that can be modified by companies and individuals who follow the GPL agreement, and can release modified versions of it. Automation of APP deployment, administration of configuration items, continuous delivery, and management of (AWS) cloud services are the objectives of the Ansible platform. Ansible's primary function is to assist staff members who work in operations and maintenance in automating these tasks in order to decrease human error, boost business automation, and raise operations and maintenance efficiency. Its main components include the following.

(1) Connection plugins: in charge of coordinating with the monitored end; ansible connects to each host based on the connection plugins, by default ansible uses SSH to connect to each host, but it also accepts different connection protocols, such as local, zeromq.

(2) Host inventory: hosts specified for operation, is a configuration file inside the definition of monitoring hosts; Host Inventory (host group), management nodes through various types of modules to achieve the corresponding operation; where we might refer to the batch execution of a single module or command as ad-hoc.

(3) Various modules: core modules, command modules, custom modules, core modules. By calling core modules,

core modules are used to accomplish management activities. Custom modules support many languages and perform tasks that core modules are unable to.

(4) Plugins: plugins complete functions such as logging emails.

(5) Playbooks : When playbooks perform multiple tasks, they allow nodes to run multiple tasks at once. A collection of multiple tasks is implemented through playbooks to achieve a class of functions, such as installation and deployment of web services, bulk backup of database servers, etc.

after entering Ansible's Ad-Hoc command set or playbook on the server terminal while using Ansible or Ansible-playbooks. Ansible will adhere to the pre-programmed guidelines to separate the playbooks into plays one at a time, then group the plays into tasks that are specific to Ansible. Then invoke each module and plugin needed for the job. The task specified in the form of temporary files or commands transferred to the remote client to run and return the results of execution will be done in accordance with the list of hosts defined in the Inventory using SSH. If it's a temporary file, it will be erased after execution if it is. The workflow is shown in the following Fig. 7.

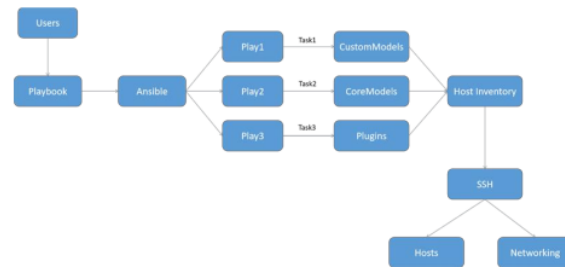


Fig. 7. Ansible Workflow Chart

Ansible has the following features: 1) no agents: no need to install any client on the controlled host; 2) no server: no server side, just run commands directly when using; 3) modules in any languages: module-based work, can use any language to develop modules; 4) yaml, not code: use yaml language custom script playbook; 5) SSH by default: based on SSH work; 6) strong multi-tier solution: can achieve multi-level command.

To summarize the benefits of using Ansible: 1) Because it is lightweight and only requires one master node to be installed and setup, users will find it to be quite user-friendly; 2) Python is the foundation for Ansible's tool development., and DevOps is becoming more and more popular in the IT world, which is a wider range of Python language applications, and the difficulty of operations and maintenance personnel to develop their own operations and maintenance tools is reduced, which also makes the secondary development of Ansible more convenient; 3) clientless, is based on SSH [16] protocol, which can be used once the management side is configured; 4) There are a ton of built-in modules, and some businesses are even creating relevant functional modules for businesses; Ansible is thus a potent tool for managing system configuration as well as for system deployment. You may easily finish the deployment and configuration management of the system using this effective tool.

At the same time, Ansible tools also have some shortcomings that need to be improved, such as the design

and positioning of Ansible is not a complete operations and maintenance management system, so the official web UI interface is not provided accordingly; Ansible support for Windows systems is not very good, requiring the installation of SSH protocol and other software; Ansible is flawed at large scale concurrency and can suffer from low efficiency and hang easily.

III. Examples of Practical Ansible Applications

A. Application scenario description

When deploying the cloud platform on site, there are relatively more servers deployed, and the difference between the contents of each server deployment is not very big, and each server needs to do the corresponding test after deployment to determine whether the deployment is correct. Such a process will take a long time, and there is no guarantee that all the deployed ones are correct, which leads to low efficiency, long deployment cycle time, and the system cannot be updated in time. This kind of operation and maintenance method consumes a lot of human and material resources, and also provides trouble for users. So, there is a great need for an automated operation and maintenance system to simplify the complexity of the cloud platform in the deployment process, reduce the work pressure when deploying the cloud platform, and reduce the operation and maintenance costs.

B. System function description

Deploying the system is becoming more and more challenging as cloud computing advances. In order to improve the efficiency, it is necessary to use a perfect automated deployment tool to install and deploy the system. Since it is a cloud platform automated deployment system based on the Ansible tool, it should not only inherit the tool's current functions but also create new ones tailored to the specific situation. The study and research of Ansible and other automated deployment tools found that Ansible does not have a good web UI interface and is not suitable for remote operation of staff, so it is also necessary to develop a set of web UI interface for automated deployment of the cloud platform, which is convenient for operation, and also convenient for interfacing with other people in the future, and more intuitive deployment of the system. After improvement, the services provided by the automated deployment system of the cloud platform can be roughly divided into the following service modules: 1) basic function services; 2) large file transfer services; 3) remote web cloud desktop services.

1. Basic function services

These basic functional services are mainly based on the services originally provided by the Ansible platform and are improved on top of them to suit the actual application system. The basic services include the following services: 1) remote command execution services; 2) remote file modification services; 3) remote small file distribution services; 4) database file creation, import and export services; 5) service management services; 6) scheduled task management and other functions. Finally, it also includes some services such as the determination of server network connection and the prompting of server failure.

2. Large file transfer services

Some improved algorithms are used to speed up the transfer of large template files over the network, so it is possible to perform a large file transfer service based on the improved algorithms.

3. Remote web cloud desktop services

Although the web UI interface is provided for operation, the operating system is Windows, for example. Since Ansible tools do not support Windows OS very well, it is not possible to provide good services to Windows users by providing web UI interface only. So, the automated deployment system for the cloud platform needs to provide simple desktop services to the users. For the sake of simplicity, the system provides a web version of the desktop services for users to facilitate deployment and reduce labor and material resources.

C. Overall system architecture

Based on the requirement analysis and functional analysis of the system done before, the overall architecture of the automated deployment system is shown here. The structure of the whole system is roughly divided into the following layers: user presentation layer, data processing layer, intermediate plugin layer, service layer, and controlled object layer. As shown in the Fig. 8.

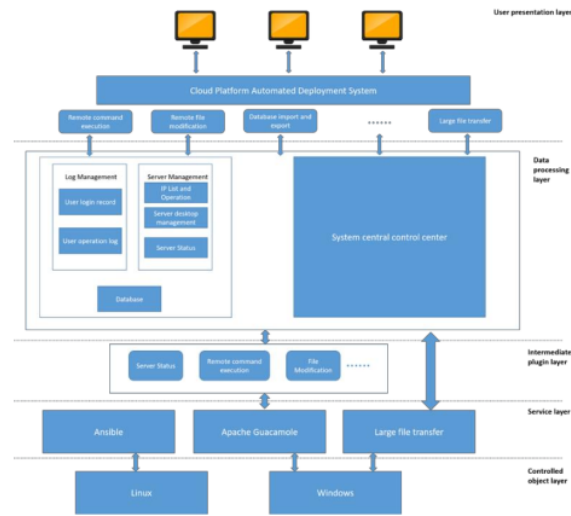


Fig. 8. Overall system architecture diagram

User presentation layer: The user presentation layer is presented to the user through the web UI, which is more intuitive and convenient for the user to operate. Users can login to the system by user name and password, and enter the management page of the system, and can make corresponding operations according to the needs of users.

Data processing layer: The data processing layer consists of two major parts: management of user data information and user operation information, and data processing of services provided to users. The main task of user data information and user operation information management is to record and save user's personal information and user operation information to a database file for future management personnel to query. The data processing part of the service for users is through a central processing system, which also receives instructions from users and reacts accordingly. For example, the user sends a command to inquire about the status of the server, and the central control center receives the command and starts processing the command.

Intermediate plugin layer: The purpose of the intermediate plugin layer is to better extend the system in the future, for example, when Ansible tools have a new feature, we also need to use this feature, at this time we need

to change only two places a user-oriented web UI, one is our intermediate plugin layer, we only need to add a plugin to provide this service to the user. The intermediate plugin layer is the key to its next layer of consulting services, but also the core of the entire system.

Service layer: The service layer is the core of the whole cloud platform automated deployment system, which provides various services for users. For example, Remote desktop services gateway Apache Guacamole, large file transfer services, etc.

Controlled object layer: The controlled object layer is mainly controlled servers or virtual machines of various operating systems, such as Debian OS, Windows OS, etc.

IV. CONCLUSION

This paper consists of the following main parts:

(1) This paper introduces the development process of agile development methods, then describes one of the most widely practiced applications of agile development models, the Scrum model, and analyzes the four main concepts of Scrum software development and the software development process.

(2) The process of estimating the workload using agile development involves teamwork to estimate the workload for each story point, determine the difficulty of work for each story point, and obtain the average rate of team work. Further it is possible to obtain the completion time (Size/Average speed) for each story point. It also describes the importance of the burn down chart to visually represent the project completion progress.

(3) From the development of the DevOps development model to the introduction of the three core pillars, we analyze the effectiveness of DevOps in eliminating the barriers between the two teams through continuous optimization of processes, overall innovation of system architecture, and continuous upgrading of organizational culture from a global perspective.

(4) Describes in detail the Ansible platform, an application platform for DevOps. This includes the main components of the Ansible platform, its features, and its benefits.

(5) The last part is a practical application of the platform. Based on the Ansible platform, an automated operation and maintenance system is built to solve the problems of low work efficiency and long deployment cycle time when deploying a cloud platform with multiple servers. The functions and overall structure of the system are analyzed, and the process of deploying an automated operation and maintenance system is broadly explained.

The above section summarizes the process of agile software development from learning to practice, and I believe it will be a great help for future work practice.

REFERENCES

- [1] Royce W W. Managing the development of large software systems: concepts and techniques[C]//Proceedings of the 9th international conference on Software Engineering. 1987: 328-338.
- [2] Boehm B, Grunbacher P, Briggs R O. Developing groupware for requirements negotiation: lessons learned[J]. IEEE software, 2001, 18(3): 46-55.
- [3] Manifesto A. Manifesto for agile software development[J]. 2001.
- [4] Rico D F. Capability Maturity Model Integration–CMMI Cost Model[J]. David Rico, 2001.
- [5] Sutherland J. Distributed Scrum[J]. Agile project management with outsourced development system, 1993.
- [6] Beck K, Beedle M, Van Bennekum A, et al. The agile manifesto[J]. 2001.
- [7] https://en.wikipedia.org/wiki/Burndown_chart.
- [8] Dörnenburg E. The path to devops[J]. IEEE Software, 2018, 35(5): 71-75.
- [9] Allspaw J, Hammond P. 10+ deploys per day: Dev and ops cooperation at Flickr[C]//Velocity: web performance and operations conference. 2009.
- [10] OZKAYA I. The deployment view[J]. IEEE Software, 2020, 37(3): 3-5.
- [11] Wettinger J, Breitenbücher U, Kopp O, et al. Streamlining DevOps automation for Cloud applications using TOSCA as standardized metamodel[J]. Future Generation Computer Systems, 2016, 56: 317-332.
- [12] Lwakatare L E, Karvonen T, Sauvola T, et al. Towards DevOps in the embedded systems domain: Why is it so hard?[C]//2016 49th hawaii international conference on system sciences (hicc). IEEE, 2016: 5437-5446.
- [13] Gupta V, Kapur P K, Kumar D. Modeling and measuring attributes influencing DevOps implementation in an enterprise using structural equation modeling[J]. Information and software technology, 2017, 92: 75-91.
- [14] Ansible[OL]. <https://github.com/ansible>. Feb. 15, 2017.
- [15] Mohaan M, Raithatha R. Learning Ansible[M]. Packt Publishing, 2014.
- [16] Ebert C, Gallardo G, Hernantes J, et al. Dev Ops [J]. IEEE Software, 2016, 33(3):94-100.

Assignment

GRADEMARK REPORT

FINAL GRADE

83 / 100

GENERAL COMMENTS

Instructor

Agile (40): 31.5

Story Est (20): 14.5

SwDevOp Tool (40): 37

Feedback

=====

Good description of Agile and the important concepts are discussed. I would have liked a little more detail and examples demonstrating the user story process.

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

Very good discussion of Ansible, I would like an example scenario to demonstrate the benefits but you have identified the benefits.

PAGE 1

PAGE 2



Comment 1

You need to define this term

PAGE 3

PAGE 4

PAGE 5

