

Assignment

by Huazhi Lin

Submission date: 14-Dec-2022 01:00PM (UTC+0000)

Submission ID: 192326876

File name: 40822_Huazhi_Lin_Assignment_460463_18721361.pdf (759.34K)

Word count: 5582

Character count: 31301

Assignment

Huazhi Lin
2200376
(of Affiliation)
Software Engineering and Artificial
Intelligence program
2200376@student.uvwsd.ac.uk

Abstract—All IT organizations depend on the development of software because it now controls the globe. The process of developing software is quite complicated and demands multidimensional expansion. Agile development is particularly helpful for the creation of specialized products because a single model, such as a waterfall or prototype, is insufficient to meet the needs of the product. Agile is the greatest method for efficient and speedy software development because of its adaptable nature, early delivery, and flexible life cycle. Additionally, according to the study, agile adaptation helps an organization mature and achieve CMM levels. While there are many different agile frameworks, some of them may be too complicated or confusing for an agile newcomer, therefore among them all, scrum offers a simple way to apply agile. Scrum is currently the most popular method used in software development, as well as in industries like finance, research, etc. Scrum can surely become the technique of choice for hassle-free development if we can overcome the few backlogs it has.

Keywords—project management, agile development, scrum framework

I. INTRODUCTION

Current Internet technology is evolving quickly, and software development is now an essential component of internet businesses [1]. In the software industry, finding ways to swiftly release products has emerged as the top problem. The quickly shifting market demands can no longer be accommodated by the conventional software development methodology. Agile software development, which emphasizes the prompt delivery of high-quality software that satisfies customers, was introduced in response to this situation [2].

Information technology is employed extensively in the new economy's framework and always evolving into new stages. Cloud computing, 5G, AI, and other new information technologies have received a lot of attention since the start of the twenty-first century. By the end of 2019, China's software industry revenue had reached \$7,207.187 billion US dollars, of which the revenue for the information technology services Class had reached \$4358.034 billion US dollars, according to data from the 2020 China Statistical Yearbook. Numerous software companies have been founded as a result of the advancement of science, technology, and Internet technology, which has also given rise to several potent new goods. New products' intelligence and ease have gradually displaced the conventional working paradigm. The complexity of software systems is growing, the standards for quality and safety are rising, and the rate of function upgrades is increasing due to the software products' flawless functionalities. a crucial step in capturing market possibilities.

Even though the associated theory of project management has a long history of development, there will still be a gap between theory and practice when it is used to run a project. The primary cause is due to the project's distinctive qualities, including its transience and diversity, as well as the fact that numerous stakeholders and management disciplines are involved in its implementation. Software businesses, particularly small and medium-sized ones, have scarce resources in all areas. A challenge to the growth of businesses has always been how to provide goods on time, cheaply, and of great quality. At the moment, the traditional waterfall project management model is used by the majority of domestic software companies. The linear and standardized waterfall model has drawbacks, including a lengthy delivery cycle, expensive management costs, and the inability to react to changing requirements, yet it can effectively finish the development work of some standardized projects. The old software development model (such as the waterfall model) has been superseded by the cutting-edge agile development model in the context of software development that prioritizes quality, service, and efficiency. Software firms have gradually come to understand agile software development, which is represented by Scrum and Extreme Programming, and more and more software organizations are starting to implement agile software development. They use an incremental and iterative approach to software development that puts people at the center. This allows them to quickly produce high-quality working software while maximizing advantages.

II. TRADITIONAL WATERFALL MODEL

Software project management in waterfall style (Fig 1) follows the steps of "requirements analysis design-coding-testing-delivery," with significant attention paid to the design phase. Outline design, detailed design, database design, page design, etc. are all included in the design phase. This pattern aims to address the majority of issues during the design stage. However, given the speed of software iterations and the frequency of requirement changes, it is challenging to achieve the requirements with such a cumbersome management approach today. First off, the project implementation process lacks flexibility due to an overreliance on the order of nodes before and after. Second, the waterfall paradigm is unable to adapt quickly to requirements changes. Before the project moves into the development stage, the waterfall approach requires that client demands be clarified. After beginning the development phase, it is extremely difficult and expensive to changing customer requirements. The waterfall model also heavily relies on project documentation. Client requirements documents communicate project construction objectives to developers. When parsing requirements papers, developers frequently make errors. Products cannot live up to client expectations when they are built using deviating specifications [3].

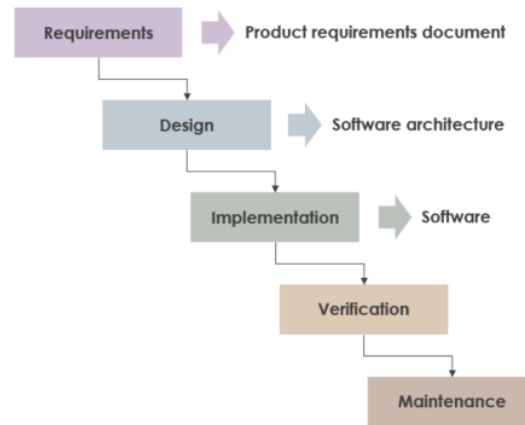


Fig. 1. Traditional waterfall model.

Consequently, the agile development model emerged. The agile development paradigm may finish project iterations rapidly and reacts quickly to varied changes. More and more software project developers are embracing it.

The Standish Group's most recent report covers initiatives they investigated between 2013 and 2017. The following chart compares the overall success, difficulty, and failure rates for waterfall versus agile projects throughout this period. Agile initiatives are around 2X more likely to succeed and 1/3 less likely to fail.

Method	Successful	Challenged	Failed
Agile	42%	50%	8%
Waterfall	26%	53%	21%

Fig. 2. Agile vs Waterfall – Project Success Rate

III. AGILE DEVELOPMENT

Agile development is a method for iterative, sequential, human-centered development. Test-driven development or business-driven development is at the heart of agile development, which concurrently works on both development and testing. Agile development, in contrast to the conventional waterfall paradigm, enables users' needs to change continuously, and the development team can react and adapt fast to these changes. The agile approach will encourage the team to regularly review and modify work-related content that promotes collaboration, self-organization, and the quick delivery of high-quality software. It will also support the standardization of project management procedures. These are the fundamental principles of agile development.

(1) People and interactions above procedures and equipment. The agile methodology emphasizes the importance of "people" and holds that people, not processes or tools, are what finish projects. The success or failure of a project can be affected by a person's qualities, the team's skills, and effective communication. Tools and processes can either help or hinder.

(2) Thorough documentation falls short of working software. Customers place a considerably higher value on the development team's consistent delivery of usable software than they do on their submission of copious amounts of exhaustive documentation. Spending a lot of time and money on creating and keeping documentation is not cost-effective. For development work, a modest amount of documentation is required. To ensure that the presented working software is the most beneficial, the agile technique maintains communication with clients through minimal documentation and frequent delivery.

(3) Scrum changes to meet evolving needs. After each sprint, Scrum will continue to provide increments. The fresh increments have to be given to clients, "finished," and meet the Scrum team's concept of completion. Because each iteration will have a product for people to use, this may quickly adapt to changes in user needs, making the product more in line with their actual needs. To create sprint development plans, scrum developers integrate the product function priorities agreed upon with customers, giving top priority to finishing and delivering the list of functions with a high market value and unambiguous needs. Some available functions will be made available for use following the conclusion of each sprint cycle. Eventually, a software product with all of the desired features and in line with customer expectations is delivered once the finalized version is incrementally deployed.

(4) Following a plan is less important than adapting to change. Change is necessary since the objective world is constantly evolving. Agile welcomes change and holds that it may improve software products. Agile modifies plans as needed, but this does not imply that plans are unnecessary; on the contrary, it necessitates more thorough planning. Agile development is appropriate

for projects with high levels of uncertainty because it allows for the fast gathering of knowledge through execution, feedback, and adjustments.

IV. SCRUM MODEL

A. Scrum concept

Scrum is one agile management strategy. This technology compares software development to a rugby team because the word "Scrum" is a rugby term. To solve issues that arise during the growth process, the players work together, use their skills, share a common purpose, trust one another, and cooperate. to the difficulty.

Currently, one of the most popular agile development methodologies is the Scrum approach. It is a framework for producing software products, and the method is gradual and iterative. A sprint's cycle typically lasts no longer than 30 days. The team members construct the Sprint to-do item list throughout an iteration cycle by choosing the tasks from the product backlog list that need to be finished this time based on priority. Each iteration cycle ends with the development team delivering and confirming incremental products.

Generally speaking, the SCRUM framework includes 3 roles, 3 artifacts, 5 activities, and 5 values [4].

3 roles: Product Owner, Scrum Master, and Scrum Team.

3 artifacts: Product Backlog, Sprint Backlog, and Burndown Chart.

5 events: Sprint, Sprint Planning Meeting, Daily Standup Meeting, Sprint Review Meeting, Sprint Retrospective Meeting.

5 values:

Commitment: Willingness to commit to goals

Concentration: use your mind and ability to do the work you committed to

Open: Scrum makes everything in the project open for everyone to see

Respect: Everyone has his unique background and experience

Courage: having the courage to make commitments, keep them, and accept the respect of others

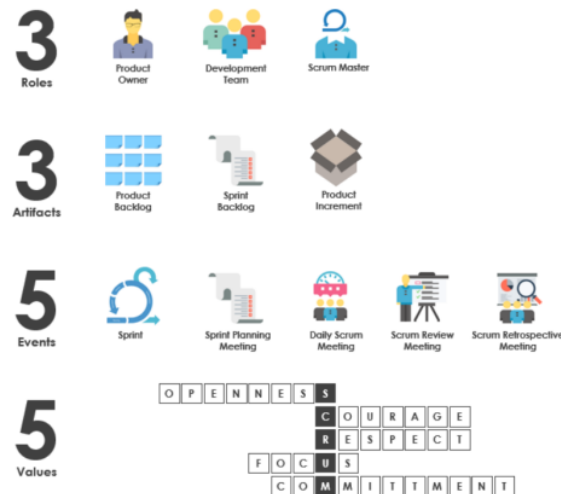


Fig. 3. 3355 Scrum Framework

B. Scrum Estimation, Estimating User Stories

Similar to conventional development approaches, Scrum development calls for an effort estimate before the project's launch. Narrative points, an abstract measurement of the work necessary to implement a given user story, are a statistic used in agile project management and development to gauge the difficulty of doing so. A narrative point can be thought of simply as a number that indicates to the team how tough a certain story is. Depending on the complexity, risk, and effort required, there may be difficulties. A story point estimate is a relative estimate that is typically made at a product backlog grooming meeting, where the team in charge of the actual development and testing work evaluates the product backlog.

Since each Sprint's work is shared among the team members in the Scrum development process, the agile team's workload is estimated using the collective estimation approach. Teams make collective estimates by participating in an estimating game, such as planning poker, which is frequently used in collective estimating. The most productive and intriguing method for estimating

effort in Agile is called planning poker. Each deck of playing cards comprises 4 sets of these Fibonacci numbers, allowing for the usage of 4 persons. The set consists of the following numbers: 0, 0.5, 1, 2, 3, 5, 8, 20, 40, ? and ∞ .

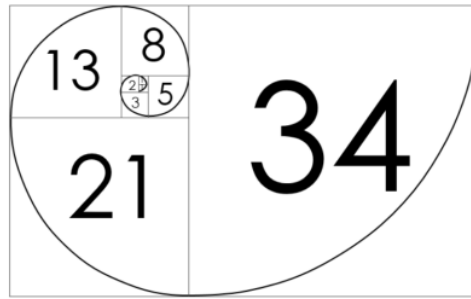


Fig. 4. Story points and Fibonacci numbers

Each participant receives a set of 12 cards totaling 0, 0.5, 1, 2, 3, 5, 8, 13, 20, and 40. The team members will hear a description of the feature from the product owner. Team members talk over the feature and, if necessary, ask the product owner questions. After the group has completed talking, each participant selects a playing card to symbolize the anticipated story points, then simultaneously reveal the cards, if teams use various estimates. Before each participant selects a new playing card, those who select the highest or lowest value should explain their selection to the others. Following the discussion, another round's estimation can be made, and the team must concur

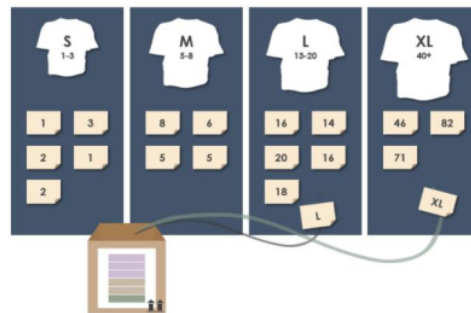


Fig. 5. Story Point Estimation

Team members give each tale a point value when estimating narrative points. Raw values are less significant than relative values. The number of stories for a story with two story points should be twice that of a story with one story point. Additionally, two-thirds of the story, or three plot points, should be included. It should also be mentioned that the most logical situation is when the single user story point does not exceed 8 and the user story has to be split again when the evaluated single story point is larger than 21.

C. Scrum development process

Scrum divides the project's whole development process into numerous iteration cycles (Sprint) [5]

(1) The product owner first decides the product demand list (product backlog), which is then arranged by priority by the formed Scrum team, which then sets the workload based on the Product Backlog list.

(2) Choose one of the goals that must be accomplished in this iteration of the cycle of 1 to 4 weeks through the sprint planning meeting (sprint planning meeting-ing), and then reduce the Story to produce a SprintBacklog finished by the Scrum Team. Then, for each participant, shorter activities that can be finished in two days are created.

(3) A daily scrum meeting [7] will be held for around 15 minutes during which the Sprint Backlog will be chosen. During this time, everyone will have the opportunity to speak. The meeting's main purpose is to provide an update on the work performed yesterday. Commit to the job that has to be done today, and then bring up any issues you run into in the meeting. Once everyone has had a chance to speak, update your Sprint burndown chart [6]. (spirit burn down).

(4) A highly significant demo meeting (sprint review meeting) will be held at this time, and participants will include product owners and customers. This meeting takes place when a Story is done, which means the Spirit Backlog is completed,

which means a Sprint is completed. Every member of the Scrum Team will take part in the presentation of the finished software products, which will take up the majority of the meeting.

(5) The sprint retrospective meeting is the penultimate stage of the development process. Each person takes a time to speak, summarizing their experiences and lessons, listing the information that needs to be corrected, and providing supplementary material for the upcoming Sprint requirements.

D. burndown chart

A burndown chart (Fig 4) plots the amount of work left to perform against the amount of time. can be used for any project requiring demonstrable advancement over time. Work completed and time is typically plotted on a burndown chart's vertical and horizontal axes, respectively. When estimating when all work will be completed, is helpful. The Development Team maps the day's remaining tasks in the Daily Scrum and updates the Sprint Burndown. For Scrum teams, a burndown chart is practically a must-have tool because it tracks project scope creep, keeps the team on schedule, and compares planned work against team progress.

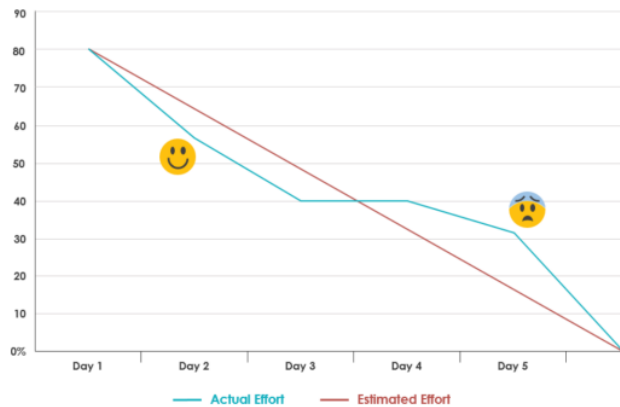


Fig. 6. Burndown Chart and Sentiment

Calculate the amount of work left by adding the Sprint Backlog estimations for each day of the Sprint to build a burndown chart. The total amount of Remaining Work across the whole Sprint Backlog is the Sprint Remaining Work. Create a graph illustrating the amount of work left to be done over time by keeping track of these sums day by day.

E. Advantages of Scrum

Software development teams can work rapidly and adapt to changes thanks to Scrum, an agile development methodology that takes planning and adaptability into account. Scrum is still the most popular agile development approach in the agile project framework, according to the 14th annual agile status report published by the agile enterprise VersionOne in 2020. At least 75% of the 1,100 respondents who participated in the 14th annual State of Agile research claimed their companies used Scrum or a hybrid that did. The following benefits of the Scrum development methodology are the major reasons why it is so popular.

(1) Roles and iterations in Scrum are well-defined, and the meeting structure is more formal and standardized. Multiple brief iterations are used to complete a project, and brief iterations are used to update user requirements to accommodate shifting demands. Customer satisfaction is given first importance in Scrum, which also emphasizes production effectiveness. Priority is no longer a primary consideration when incorporating User Stories in iterations, making Scrum more flexible and advantageous to the advancement of the development process. The earliest fixed-cycle delivery of working software that satisfies the criteria can even be sped up to one week under Scrum, which restricts the iteration period to be less than or equal to one month. Because each little iteration provides a short-term objective that is simple to execute and leaves a strong sense of accomplishment, rapid iteration can encourage active members of the development team. The work completed on Monday might be released on Friday, and users would receive feedback right away. Then continually repeat this procedure. The team members will have a strong desire for each product release after the rhythm has been established.

(2) Scrum emphasizes direct communication. The daily stand-up meeting gives team members a forum for dialogue and idea sharing. Together, everyone discusses challenging issues, which simplifies daily tasks and, to some measure, lessens complexity. The team members will first stand face to face during the daily regular meeting to exchange information, to understand what everyone did yesterday and what they plan to do today, to determine whether there are any difficulties and obstacles, whether they need assistance, and to determine whether their work plans need to be adjusted. Second, the PO will present the sprint's objective to the team members during the sprint meeting. However, it will also listen to their input and, if necessary, make any necessary adjustments. The team must then show what we have accomplished in this sprint to the

interested PO, SM, management, customers, etc. during the sprint retrospective. The development team must demonstrate in this meeting that the tasks for this sprint are "done," and the PO must vouch for their accuracy. The remaining backlog and next steps are also discussed, and changes to the product backlog are made as appropriate. Finally, during the sprint review meeting, the PO, SM, and members of the development team will talk about the areas that may be improved, as well as what worked and what didn't. Face-to-face communication is without a doubt the most effective and efficient approach to delivering information.

(3) Scrum changes to meet evolving needs. After each sprint, Scrum will continue to provide increments. The fresh increments have to be given to clients, "finished," and meet the Scrum team's concept of completion. Because each iteration will offer a product for users to use, this may quickly adapt to changes in user needs, making the product more in line with their actual needs. If the product function in this iteration is not what the user wants, we will change it in the subsequent iteration.

(4) Scrum has high levels of client involvement. Users are seen as an integral component of the agile team in scrum development, contributing to every step of the creation of a software product. Customers actively participate in and provide timely feedback on tasks like product demand analysis and prototype design, which primarily prevents deviations in the project development process. Additionally, Scrum's sprint planning meetings, sprint review meetings, etc. give consumers a platform to express their thoughts and feedback, considerably promoting the quicker and more effective delivery of software solutions.

(5) The documentation for Scrum is brief. In contrast to the conventional project management methodology, scrum development is a new approach. It promotes open communication and face-to-face interactions. At the same time, it introduces more user-friendly presentation features including burndown charts and Kanban. It is less reliant on project paperwork and merely keeps track of the more crucial project management details. Scrum development places greater emphasis on the finished product than on words.

(6) The group is motivated by relationships and self-organized. Scrum gives the development team a lot of weight. Agile teams have a strong emphasis on people orientation; there is no hierarchical leadership structure, and all team members are equal and capable of managing themselves. Scrum masters act as servant leaders and offer an agile development environment to developers. The self-managed agile team works in a positive, peaceful, and trusting environment, which has a significant positive impact on the team's productivity and morale.

In conclusion, the key benefits of Scrum can be seen in its ability to quickly adapt to users' changing needs, lower the cost of change, and steadily enhance product features through layer-by-layer iteration.

F. Scalability of Scrum

Each agile methodology has a few advantages and a few, if not all, drawbacks. Even though each agile technique has a separate implementation technology suited for particular situations, Scrum appears to be the winner in general. A major benefit is its capacity to be easily utilized in large-scale projects due to its scalability.

With the growth of Scrum's application domain, complicated large-scale projects may now be managed to utilize the management approach of managing small teams as a foundation [9]. These projects can be cross-departmental, cross-team, and even extend beyond conventional bounds. Scalability of Scrum

The Scrum of Scrums also referred to as "program management," is the key to the project's success (program management). The Scrum of Scrums can be used to expand the daily stand-up meeting, which is designed to help agile teams communicate and work with other teams when numerous teams are involved in the project or product development process. The SOS meeting can be elevated to a higher level in an organization with several Scrum teams operating concurrently by holding a Scrum of Scrums meeting, where one representative from each daily Scrum will be sent to the second Project-level SoS sessions.

LeSS's suggestion proves that Scrum's scalability is not at all an issue. LeSS is Scrum itself; it is not a system that is integrated into Scrum. LeSS's guiding idea is to facilitate scrum meetings and approaches as much as feasible. By adopting LeSS, you can lower the cost of using roles and processes. LeSS can also collaborate well with dispersed teams[8].

G. Automated testing and continuous integration

(1) Automated testing

The application of automated testing is crucial in agile development. Both the number of iterations and the number of requirements are constantly rising in Scrum. The scale of the project is expanding along with the number of iterations. The features that were finished in the previous Sprint must also be tested in the upcoming Sprint in addition to any new functionalities. This is how Regression tests need to be run frequently after iterations. It will take a long time to complete

manual testing alone, which goes against the agile development philosophy. Automated testing is crucial for ensuring that existing functionalities are of high quality during each iteration. The testers will select the most fundamental use cases with a high usage rate in each iteration and then include them in automated scripts. The automated scripts will be regularly enhanced and refined as the iterations advance, enhancing the effectiveness of automated tests. Coverage. This reduces regression testing since testers may use automated running to check the functions created in each iteration of the project rather than having to spend extra time manually checking those existing functions. Testers can devote more time to other exploratory tests to guarantee that product quality is improved with less time.

(2) Constant improvement

Testing doesn't have to wait until all features have been created thanks to the software engineering practice of continuous integration. Continuous integration seeks to minimize time wasting by locating issues as soon as feasible and fixing them quickly. To achieve quality inspection, continuous integration can continuously track developer code submissions, pull codes, compile, and test during project iterations.

V. PROBLEMS WITH SCRUM

Question 1: 94% of respondents, according to the 2017–2018 Scrum Status Survey Report, will adopt Scrum in agile development. Although there appear to be many users, only 16% of the 94% of users will be single. Scrum is widely utilized, and 78% of respondents said they would combine Scrum with other development approaches or frameworks. The cause is that Scrum is insufficiently strong and unable to handle all forms of software development on its own. To get greater results, it must be used with other development strategies.

Question 2: There are still a few small issues with the Scrum implementation, such as a lack of workflow and testing automation, poor management support, and inexperienced team members. Obtaining further assistance is crucial when putting Scrum into practice; without the support of the Scrum Team organization, or if they do not lead the team as a whole, no significant change can be made.

VI. SCRUM IMPROVEMENTS

The team members and process managers can be taught after the Scrum team is constituted, given the lack of greater management support and experienced team members in the Scrum implementation. Given the absence of Scrum's functions, it is possible to improve the separation of traditional and innovative jobs as well as the processing of sudden-innovative tasks separately. Traditional Scrum iterations forbid modifying the Scrum requirements.

This will significantly restrict Scrum's flexibility and make it challenging to handle unexpected changes in demand. The requirement to deal with unexpected change can be increased by including innovation tasks in iterations and daily standups.

Before beginning a task, the Scrum team's primary creators receive training, and the addition of novel tasks makes the execution of Scrum easier and produces better outcomes.

VII. SPINNAKERS

A. Introduction to Spinnaker

Spinnaker is an open-source multi-cloud continuous delivery technology that aids in the speedy and steady distribution of software modifications. Application management and application deployment are two of the primary services that Spinnaker offers.

application administration. Fig 5 depicts the model of the application created by Spinnaker. The environments in which microservices are deployed, such as development, production, and pre-production environments, are represented by ServerGroup in Spinnaker. Cluster is a collection of all instances of the application, and ServerGroup represents each environment. Cluster is the collection of ServerGroup. All environments for the current service are contained in a cluster. The firewall is the security group entry for the cloud provider, and the load balancer LB is equivalent to the Kubernetes Service.

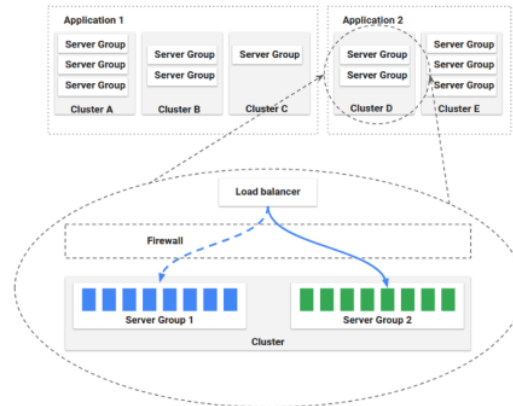


Fig. 7. Application Management Model

deploying an application. The pipeline is Spinnaker's central component. As depicted in Fig 6, a pipeline (pipeline) has various stages (stage). The phases may proceed simultaneously. The stage can be flexibly configured, for instance, if two stages are assigned to the same dependence stage, then other stages will be executed in parallel when the stage is complete

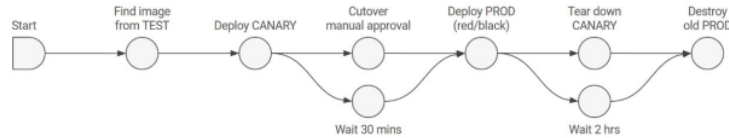


Fig. 8. pipeline process

B. When to deploy Spinnaker

Spinnaker is the solution for businesses wishing to standardize their release procedure and raise quality. Here is a waze example. With Spinnaker, developers can concentrate on creating business logic rather than learning every aspect of the various cloud platforms. To organize handoffs between development and QA teams, teams can track the lifetime of deployments using a variety of notification channels, including email, Slack, and SMS.

The most valuable resource for users in Waze is their time, and the cloud-based service the Waze team provides aids users in choosing the optimum route to reduce travel time. But the quality of a cloud service depends on its accessibility. To increase the resilience of our production systems, Waze uses a variety of cloud service providers. We are better able to withstand a DNS DDOS attack, a regional failure, by deploying an active-active architecture spanning Google Cloud Platform (GCP) and AWS. We need the ability to roll that code back or correct it as soon as feasible since sometimes a defect in the routing or ETA estimates sneaks into production. That is more difficult to accomplish in a multi-cloud scenario. For instance, our real-time routing solution covers more than 80 managed instance groups and autoscaling groups spread over two cloud providers and various countries. Spinnaker delivery is useful in this situation.

Cloud deployments on a large scale might be challenging. Fortunately, Spinnaker abstracts a lot of the specifics of each cloud provider, allowing developers to concentrate on improving Waze for customers rather than worrying about the specifics of many cloud providers. We can keep crucial principles for continuous delivery like canaries, immutable infrastructure, and quick rollbacks throughout.

Jenkins is treated as a first-class member of Spinnaker, thus after code is committed to git and Jenkins creates a package, that package initiates the main deployment pipeline for that specific microservice. This pipeline keeps running any automated testing phases while baking the package into an immutable machine image across several cloud providers. Without delving too deeply into the specifics of each platform, the deployment moves from testing through staging utilizing the blue/green deployment technique, and then to production. To ensure that each cloud's deployment operations occur automatically and accurately without the need for any configuration, take note that Spinnaker automatically determines the relevant image IDs per cloud provider.

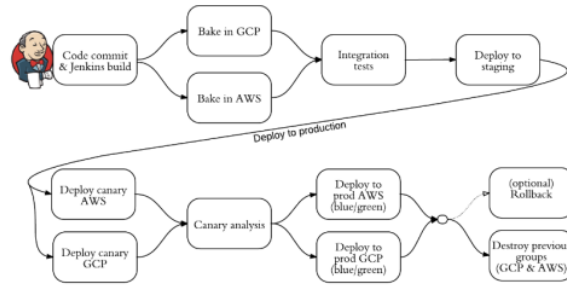


Fig. 9. Example of a multi-cloud pipeline

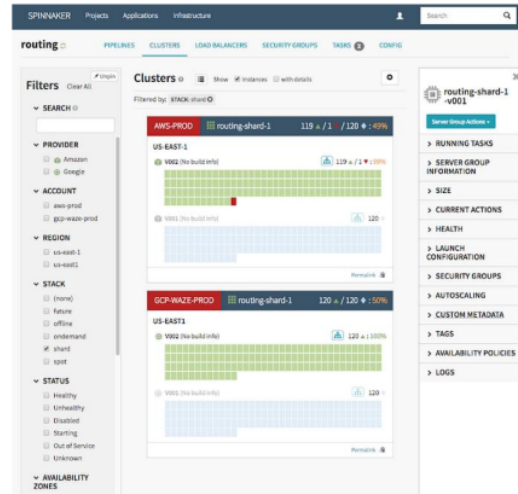


Fig. 10. Multi-Cloud blue/green deployment using Spinnaker

Spinnaker allows developers to concentrate on creating business logic rather than learning every aspect of every cloud platform. Teams can organize handoffs between developer and QA teams by tracking the deployment lifecycle using a variety of notification methods, including email, Slack, and SMS. Developers may make informed judgments about the condition of their deployment with the use of tools like canary analysis and rapid rollbacks. Since Spinnaker was created to be a self-service platform, developers may complete all of these tasks with little assistance from the operations team.

Waze works hard to provide users with new features and issue fixes as soon as possible. Waze can accomplish this because of Spinnaker, which also makes multi-cloud deployments and rollbacks straightforward, convenient, and dependable.

VIII. CONCLUSION

The waterfall model was first introduced before agile project management. It is a simple paradigm for project development. The agile development framework most frequently employed is scrum. Compared to the conventional waterfall model, its quick iteration and small-step development idea are more suited for software project management with frequent and small-scale requirements changes. Its iterative and incremental development approach can increase the productivity of software creation, collect consumer input through frequent delivery, and produce goods that are better suited to market and customer demands. It offers the following benefits:

- (1) When adopting Scrum, high-quality, high-value products may be delivered to clients in a shorter period than when using traditional waterfall development, which necessitates a lengthy development cycle.
- (2) It can significantly reduce the time it takes to release a product, and the business can focus its efforts on implementing high-value features, cutting time expenses, and increasing productivity. improved client satisfaction.

(3) The Scrum agile methodology reduces the number of time-consuming tasks and document creation, giving development team members more time to focus on their preferred tasks.

(4) The requirements can change frequently during the Scrum agile method implementation, and the order of the work needs to be altered to maximize the delivery of worthwhile products and make the alterations more significant.

(5) Compared to the waterfall development approach, the Scrum agile methodology is simpler and provides for more detailed project planning and program tracking. Hold daily sprint meetings and use the burndown chart to display the daily work progress to ensure that the project manager always has a complete understanding of the project's status.

In general, Scrum relies on flexible organizational characteristics and effective member participation, realizes flat management and a good interactive feedback mechanism, and resolves the conundrum in the traditional software development process question. Scrum is a key practice project in the development process of agile project management methods. In addition to making process project management more tangible as a great agile project development management technique, controllable software's self-organization and self-management working approach also enable all participants to actively cooperate and participate throughout the entire process. These project developers must modify the idea of how projects are run in the future and optimize and improve based on project experiences.

REFERENCES

- [1] Fan Shaofen. Discussion on Agile Testing Based on Scrum[J]. Intelligent Computer and Application, 2017,7(5):111-112+ 116.
- [2] MARTIN R C. Agile software development, principles, patterns, and practices[M]. NJ, USA: Prentice Hall, 2002.
- [3] Guo Lianming. Talking about the Waterfall Model and Its Limitations[J]. Science and Technology Outlook, 2016,26(06):172.
- [4] El-Khatib H T, Williams ML, MacKinnon L M. Using a dis-tributed approach to retrieve and integrate information from heterogeneous distributed databases [J]. Computer Journal, 2002,45(4):381-394.
- [5] Shang Zhiwei. Agile Development Research [J]. Electronic World, 2018(7):51-52.
- [6] Wang Deyan, Li Ping. Design of Software Comprehensive Training Room Based on Software Project Development Process[J]. Journal of Wuxi Commercial Vocational and Technical College, 2016,10(3):76-78.
- [7] Herig Recalde, Santiago Soria, Diego Vallejo-Huanga, "Internal Search Engine and Recommender System with Natural Language Processing in PaaS", 2022 IEEE/ACIS 20th International Conference on Software Engineering Research, Management and Applications (SERA), pp.63-69, 2022.
- [8] Mirko Farina, Marina Ivanova, Maxim Korsunov, Alexander Krivososov, Artem Kruglov, Nataliya Matrosova, Giancarlo Succi, "Musical Practices in Software Development: Insights from Gary Marcus's Guitar Zero", 2021 International Conference "Nonlinearity, Information and Robotics" (NIR), pp.1-7, 2021.
- [9] Anthony Bouzakis, "Scaling Challenges of High-Tech SMEs developing Internet of Things Technology", 2021 International Conference on Engineering Management of Communication and Technology (EMCTECH), pp.1-5, 2021.

Assignment

GRADEMARK REPORT

FINAL GRADE

77 / 100

GENERAL COMMENTS

Instructor

Agile (40):30

Story Est (20): 17

SwDevOp Tool (40): 30

Feedback

=====

A very good discussion of the Agile approach. The assignment asked you to consider 4 important concepts, and you have not really addressed these in sufficient detail. Also you should have provided more examples of the process

Good explanation of story estimation, the example could have been a bit more illustrative of the idea.

I wanted a little more discussion of how Spinnaker worked and its main purpose. The example provided was appropriate and this is quite a good effort.

PAGE 1

PAGE 2

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

