Design and Practice of Software Architecture in Agile Development

Ding Yong1,*

1College of Information Engineering, College of Arts and Sciences, Yunnan Normal University, Kunming City, Yunnan, China, 650228

*Corresponding author e-mail: yongding345@ynnu.edu.cn

Abstract. Software architecture design is a very critical part of the software development process. In order to be able to respond to changes in user requirements in software development in a timely manner, a convenient and lightweight development model should be adopted. Agile development adopts a human-oriented, step-by-step development process that can quickly respond to changes in software requirements, adjust the design of software architecture in a timely manner and adapt to software development in a dynamic environment. Based on the analysis and comparison of traditional development process and agile development process, this paper analyzes the key technologies in agile architecture design and proposes a software architecture design method based on agile development mode.

Keywords: Design, Agile, Software

1. Introduction
Software development usually needs to complete the four stages of requirements analysis, software design, code writing and software testing based on the definition and planning of the problem. The process is a complex and even cyclical process. Traditional development methods usually require determining all of the user's needs before proceeding with specific development and then develop a detailed plan that spans the entire software project development cycle, after which the development process is based. The advantage of this development model is that it can maintain the consistency of the entire software design. The disadvantage is that once the situation changes, the framework structure needs to be adjusted. This detailed plan may be invalidated, resulting in a large number of complexities with no application value. Documentation, unnecessarily increases the cost and difficulty of software development. In order to avoid the drawbacks of these traditional heavy-duty development methods, a new lightweight software development method-agile development method has emerged in recent years. This method is a typical light software development method, which integrates many light software development methods. The advantages, emphasis on people-oriented, highlight the characteristics of "adaptive", can quickly adjust according to various changes in the software development process, to minimize the cost and risk of software development [1].

Based on the advantages and disadvantages of agile development and traditional development methods, this paper analyzes the core ideas and design concepts of agile development model and
discusses the design method of software architecture based on agile development mode, including the adopted design technology. Build ideas and implementation specifications [2].

2. Comparison of agile development model and traditional development methods
Agile development is a recently-recognized software development method. It is an iterative-based software development method that drives the implementation and advancement of the entire software development process with human core. It is a new method and new management software development process. Ideas. In agile software development, the software project is divided into several sub-projects, which are completed through multiple iterations. Each iteration has a clear goal and can quickly deliver runnable software. Agile development focuses on the design of concepts and software architectures to simplify the detailed design of the software, leaving room for adjustment later. When adopting agile-developed software projects, the initial design of the software architecture is just to meet the demand. Later, according to the understanding and update requirements of the software requirements, the reconstruction technology is used to gradually adjust the design. Agile development is an organic whole formed by a combination of simple but interdependent practice steps, highlighting the relationships that exist between "people", including communication between programmers, feedback between development teams and customers and attention to both parties. The courage of innovation and the simplicity of the software system. Usually, from the beginning of the software project, it emphasizes the need for periodic feedback through software testing. The programmer submits the first draft of the software to the customer as soon as possible and cooperates with the customer to discover the vulnerabilities through the software. The first draft is optimized while responding to new customer demands for software. The key points in the design process are shown in Table 1.

Table 1. Key points of agile development technology

| Design | Mode | Method | Destination |
|--------|------|--------|-------------|
| Ratio  | 56.5%| 26.5%  | 17.0%       |

Agile development has improved the four stages of traditional software development, blurring the concept of "stage", avoiding the cumbersome and rigid traditional software development methods, making it more flexible and responding to the latest needs of customers in a timely manner. Dynamically adjust the development process and software architecture and continuously and iteratively refine, optimize and improve the simple and few development steps to the most satisfactory results for users [3].

3. Software architecture design based on agile development model
Through the previous analysis, the architecture design based on agile development includes three core elements: one is the overall idea of agile architecture design; the other is the key technology used in agile architecture design; the third is the standardized management of design and implementation process [4].

3.1. Design ideas
The outstanding advantage of agile development lies in the rapid and incremental development method, the first time to deliver the workable software to the customer and then according to the communication with the customer, feedback the use of the software, adjust the software structure according to customer needs. It is a human-centered, iterative sublimation, step-by-step development approach that runs through all aspects of agile software development. This principle is also followed for the design of the software architecture. The development process of agile software is also a process in which the software functions are gradually improved and the version is gradually upgraded. In other words, the architecture design in agile development adopts an evolutionary design method, that is, in the whole cycle of software development, through repeated iterations to modify, improve and enrich the design, the architecture is optimized. Maximize customer demand for software. It should be noted
that the evolutionary software architecture approach should follow three principles: 1) The current iterative architecture should be designed to avoid harming the architecture and functionality already implemented; 2) the actual iterative architecture should be consistent with the neighborhood model. Consistent, avoiding misunderstanding of neighborhoods and increasing development costs; 3) Architecture design should be complete and all levels of the architecture model should be unified [5].

In agile development, the architecture design process of each iteration needs to go through six steps. First, the requirements in the current iteration are extracted according to the user's overall requirements, then the neighborhood modeling is performed and then the conceptual architecture design is performed according to the model. If the design meets the customer's needs, the next step is to refine the software architecture and then the user's verification of the architecture design. If the user's needs change, the first phase of the demand analysis is re-executed.

3.2. Key technologies
There are two aspects of key technologies in agile development. One is refactoring technology. In agile development, it is a rapid adaptation to constantly changing and frequently changing design environments through refactoring technology. The so-called refactoring is to make full use of the existing functions of the software, improve the software quality and performance through partial adjustment of the overall architecture and program code, make the design pattern of the software architecture more rational and improve the software extension. In essence, refactoring is to adjust the internal structure of the software as much as possible to preserve the existing functions of the software and reduce the software upgrade cost. Refactoring technology runs through the entire process of software development, including architecture reconstruction, design refactoring, code refactoring and business refactoring. The second is the design pattern. In the software reconstruction process, the design pattern is usually used to improve the existing design. The design pattern is actually the skills and design experience that many software developers have summarized during the development process, which can be used repeatedly. In agile development, the code refactoring phase can be used to make the program more reliable and understandable.

3.3. Process management
According to the basic principles of agile development, such as simplicity, communication, feedback, courage and quick delivery of workable software, a successful agile architecture requires the development team to work together to complete the following four steps: First, the product owner develops a product list and Sort the user stories in the list according to their priorities, then select a group as the current design goal and list the subtasks. Second, the architect establishes the initial architecture, including the establishment of the architectural vision, the positioning of the architectural style and Third, the architect and team incremental maintenance architecture, that is, when the customer has new requirements for the delivered product or the demand changes, the response should be timely and the architect should communicate with the team continuously. Promote the understanding of the entire team and the overall architecture, incrementally maintain the architecture through refactoring; fourth, determine the incremental content of each iteration architecture, obtain feedback on the architecture through the communication between the architect and team members and obtain new Increased architectural content.

In general, all modeling is "sufficient" modeling done on the whiteboard. Through the above four steps, the model will gradually grow and gradually improve with each iteration and finally meet the user's needs.

4. Application of agile development mode in the software architecture of "online teaching system" project
The online teaching system is a small and medium-sized software development project for higher vocational colleges. Due to the short development cycle of the project and the unclear user requirements, the customer demand of the system is likely to change. In this case, the adoption is
simple and fast. The light and agile development model with strong adaptability is suitable. The specific application of the agile software development method is described below in conjunction with the development of the project. The time sequence of the specific application mode is shown in Figure 1.

![Figure 1. Specific application mode](image)

### 4.1. Initialization function module determination

Agile software development emphasizes the involvement of on-site customers. In the software development process, developers are always ready to communicate with the major customers of the software, such as teachers and students, about business issues, report on the progress of the project and get feedback and support. According to the user needs of this project, the initial user story is determined and the online test sub-module is selected as the functional module that is implemented preferentially.

### 4.2. Iterative development

According to the initial software architecture proposed by the architect, the software development team will design, code and test the software together by programmers and testers. The whole process follows the principles of simplicity, refactoring and collective ownership, which is convenient for optimizing the internal structure of the system. Eliminate redundancy and improve code quality and readability. Developers should implement the initial architecture as soon as possible and deliver it to customers for trial and feedback. According to the feedback information of the user, in the "online teaching system" project, the online test sub-module needs to add system security design content such as identity verification module, input validity verification and information encryption storage.

### 4.3. Small release

Agile software development requires a combination of business and technology to quickly deliver a workable product and determine the scope of the next release, small hair. Combined with the development time requirements of this system.

### 5. Conclusion

This paper analyzes the design method of software architecture in agile development mode in detail and systematically expounds the application of agile development actual software development case "online teaching system" project, which fully shows the simplicity and flexibility of agile development.

### Acknowledgments

2018 Yunnan Intelligent Tourism Science and Technology Innovation Team.

### References

[1] Crescencio Lima, Wesley KG Assunção, Jabier Martinez, William Mendonça, Ivan C Machado, Christina FG Chavez. Product line architecture recovery with outlier filtering in software families: the Apo-Games case study[J]. Journal of the Brazilian Computer Society, 2019, 25(1).

[2] Rakesh Kumar Jha, Burhan Num Mina Llah. Software Defined Optical Networks (SDON): proposed architecture and comparative analysis[J]. Journal of the European Optical Society-Rapid Publications, 2019, 15(1).

[3] Shraddhha Narasimha, Emma Dixon, Jeffrey W. Bertrand, Kapil Chalil Madathil. An empirical
study to investigate the efficacy of collaborative immersive virtual reality systems for designing information architecture of software systems[J]. Applied Ergonomics, 2019, 80.

[4] V. Huber, A. Huber, D. Kinna, G. F. Matthews, G. Sergienko, I. Balboa, S. Brezinsek, P. J. Lomas, J. Mailloux, P. McCullen, Ph. Mertens, F. G. Rimini, S. Silburn, D. Valcarcel, K.-D. Zastrow, JET Contributors. The software and hardware architecture of the real-time protection of in-vessel components in JET-ILW[J]. Nuclear Fusion, 2019, 59(7).

[5] Emilia Rosa Jimson, Kashif Nisar, Mohd Hanafi Ahmad Hijazi. The State of the Art of Software Defined Networking (SDN) Issues in Current Network Architecture and a Solution for Network Management Using the SDN[J]. International Journal of Technology Diffusion (IJTD), 2019, 10(3).