

Research on Basic Strategies of Constituting Software Development Method

Yu Gao

Department of Computer Science, Zhejiang Ocean
University, Zhoushan, Zhejiang, PRC 316000
gaoyu@zjou.edu.cn

Zhen-Bo Bi

Department of Computer Science, Zhejiang Ocean
University, Zhoushan, Zhejiang, PRC 316000
bzb136@sina.com

Abstract—In the software methodology, the structure of the software development method is a worthy issue. After studying the structure of software development method, a rule is found. The rule is that every software development method is formed from the basic strategies of development software. Discussed the content of the familiar some basic strategy, these basic strategies are: (1) according to linear order, stages development, (2)iterative development, step-by-step perfection, (3)split up, divide and conquer, (4)reuse, component assembling. The problems of application of the basic strategies are discussed. An application is to build new software development methods, and the other application is to formulate the development program of definite issues.

Keywords—software methodology;software development;basic strategy

I. INTRODUCTION

For years, people have study software development methods [1]. Currently, there are many different software development methods [2], and new software development methods are put forward still. In order to build more suitable software development method, we need to study the structure of software development methods. By studying the structure of software development method, we found that: the current software development is constituted by the basic strategy, although their structure is different, complexity is different. Or rather, the basic strategy of developing software is basic elements constituting software development methods.

What is basic strategy of constituting software development method? What is characteristics and scope of application to each basic strategy? How to use these basic strategies in building new software development methods? How to use these basic strategies in drafting development programs of specific issues?

In software methodology, the research of these issues has important significance. Currently, no one has done thorough research on these issues. We did some research. The following is description.

II. SOME FREQUENTLY-USED BASIC STRATEGY OF CONSTITUTING SOFTWARE DEVELOPMENT METHOD

A. The first basic strategy: "according to linear order, stages development"

"According to linear order, stages development" is a basic strategy of development software. In some software development methods that the waterfall model is representative [3], expression of the basic strategy is very

obvious. Following, by analyzing the waterfall model, we understand the basic strategy.

Waterfall model is a software development method, it is earliest. In waterfall model, software development process is divided into several stages (problem definition, feasibility study, needs analysis, overall design, detailed design, code, test and maintenance). When developing software, according to a linear sequence, from one stage to the next stage, the people carry out development activities. Each stage has a specific target. At the end of each phase, development work has been reviewed. After passing the examination, development work can proceed to the next stage. According to a linear sequence, when the task of each stage is completed, the whole development task is completed. To the other type of such software development methods, development processes are also much the same.

According to the above description, we can see that the main features of this basic strategy are to emphasize stages and linear sequence. Task of each stage is decided, the boundaries between the various stages is clear, each stage is the basis and premise of the next phase. In the development, the developer must strictly follow linear order relationship between the various stages.

When using only this basic strategy to build the software development method, it applies to situation that user needs is clear, and the complexity of problem is low and scale is small. In the early development activities, to face situation is like this. Therefore, in the early development activities, this basic strategy has been widely used to build software development methods, and result is very good.

"According to linear order, stages development" is a basic strategy, although its form is simple, but it is important. It makes that operation step is clear, operation process is standardized and it is beneficial for the monitoring and management of process. It has changed the disorder state to the early software development activities. It made a great contribution, for the ease of software crisis. Although later there are some other basic strategy, and people have built some more complex software development method, but when building any kind of software development methods, this basic strategy should be used.

B. The second basic strategy: "iterative development, step-by-step perfection"

"Iterative development, step-by-step perfection" is a basic strategy. In some software development methods that the rapid prototype model is representative [4], expression of the basic strategy is very obvious. Following, by analyzing the rapid prototype model, we understand the basic strategy.

Based on the waterfall model, for disposing issues that user's needs are ambiguous or incomplete and unceasing change, rapid prototype model is built. When using the rapid prototype model, first, on the basis of basic needs of the user, to create quickly a prototype system, which is a subset of the target software product in the function. Please user to try this prototype system on the computer. After trying the prototype system, users do the comments. Developers quickly modify the prototype system in accordance with user's comments. After modifying the prototype system, developers allow again users to try the prototype system. Users do the comments again, developers modify again. So doing iteration, repeated trial and modified several times, the prototype system has been gradually improved. When the user's needs are met, the iteration stops. Then on this basis, the using "discard" method or "evolvment" method, software development task was eventually completed. Content of "discard" method is discarding prototype system, according to user's needs, from scratch to develop the software. Content of "evolvment" method is gradually evolution to the target software, eventually the prototype is evolved into the target software, to achieve the customer satisfaction.

Above description shows that main features of this basic strategy is to emphasize the iterative process. Iteration is the best way to solve the problem of ambiguity and uncertainty, to improve gradually is the goal and the inevitable result of iteration. This basic strategy accord with laws of understanding things, laws is "practice, knowledge, again practice, again knowledge", after several repeat, knowledge is gradually deepened, target system is gradually improved.

Rapid prototyping model has apply the basic strategy of "iterative development, step-by-step perfection" in the requirements analysis phase. For other software development method of this type, they apply the basic strategy of "Iterative development, step-by-step perfection" can be in the requirements analysis phase, and can also be in the other phase software life cycle.

Analyzing the development process of such software development method, we can see that: it applies to situation that need is not entirely clear, and finishing task is impossible to do once only. For example, the user's need is vague, or there will be many changes in the development process, then developers can apply the basic strategy of "iterative development, step-by-step perfection". Through iterative development, work of this stage is gradually detailed, and the target software is gradually improving.

C. *The third basic strategy: "split up, divide and conquer"*

"Split up, divide and conquer" is a basic strategy. In some software development methods that the incremental model is representative [5], expression of this basic strategy is particularly obvious. Following, by analyzing the incremental model, we understand the content of this basic strategy.

When using incremental model in developing software, a large and complex software development tasks are broken down into several small and simple software development task. Each small and simple software development task is called an "increment". Each "increment" can be achieved one

function of the software system. When developing software, one time to complete one the "increment", after submitting to the user, and then complete the next "increment", so, to submit software products to user in batches. Usually, the first "increment" is the core of the software, which can achieve the most important function of software. When all of the "increment" is completed, the entire software development task is completed.

For other software development methods of this type, although they split-up process may be different, the method of divide and conquer may be different. But the development process can reflect the same basic strategy.

From the above description, we can see which main features of this basic strategy are to emphasize the "divide and rule". Although the settlement of large and complex problems is difficult, but after the problem of large and complex is decomposed into several small problems, because to solve every little problem is very easy, when all the small problems are resolved respectively, then large and complex problem will be solved.

The basic strategy of "split up, divide and conquer" applies to situation of developing software system of scale Large and structure complex. After such software system is decomposed into a number of increments of small and simple, because the development of each increment is relatively easy, so the entire software development is not difficult.

Appearing incremental model is after appearing the waterfall model and rapid prototyping model. When developing each "increment", basic strategy of "according to linear order, stages development" can be used, basic strategy of "iterative development, step-by-step perfection" can be used, or other basic strategy is adopted, there is no strict limit.

D. *The fourth basic strategy: "reuse, component assembling"*

"Reuse, component assembling" is a basic strategy. In some software development methods that component assembling model is representative [6], expression of this basic strategy is particularly obvious. Following, by analyzing the component assembling model, we understand the content of this basic strategy.

Component assembling model is a software development method that built on the basis of object-oriented technology. After appearing a variety of software development method that object-oriented technology is feature, people begin consider reuse of "software components". Narrowly speaking, "software component" can be source code of module. Broadly speaking, in addition to source code, the "software components" can also include requirements specification, software architecture, test plan, user interface, and other software materials. These "software component" is collected together. They are placed in "software component library". In development activities of the future, people can reuse them. When developing software using the component assembly model, in the development process, people can reuse the software component of "software component library", people can use directly, or people can use after slightly modifying. According to structure of the target

software, people can select the software components of "software component library", through assembly, to form a complete software version. Then developer let user evaluation. Thus, after repeated several times, software development work was done.

For other software development methods of such type, although the range reusing software components is different, or the process reusing software components is different, but they all reflect the basic strategy of "reuse, component assembling".

From the above description, we can see which main features of this basic strategy are to emphasize the reuse. Reuse can reduce development costs, and enhance efficiency, and software quality has assurance. The basic strategy of "reuse, component assembling" applies to situation that There are a lot of software components. These software components are stored in "software component library", when developing software, they are reused, through selection in "software component library", assembling into the target software system. Originally the software development is a very complex task, but if the "software component library" as the basis, to apply this basic strategy, then the software development process will be simplified.

III. THE BASIC STRATEGY AND NEW SOFTWARE DEVELOPMENT METHOD

How to build new software development methods to use these basic strategies? Here we analyze.

Why the basic strategy of "iterative development, step-by-step perfection" is used in some software development method? This is because: In software development, when the objective situation has many ambiguities, variability and uncertainty, for the task of certain stage, it can be not complete to do only one. People yet can not wait to eliminate the ambiguity, variability and uncertainty, people need to work. At this point, in accordance with the laws of understanding things, the best approach is to use iterative development approach, people carry out repeated practice, repeatedly recognized, gradual improvement. That is to say, when we are in an environment that is full of ambiguity, variability and uncertainty in software development, then we should consider the use basic strategy of "iterative development, step-by-step perfection" in building new software development method. Such as RUP [7] method and XP [8] method, since they often have to deal with changes in users' needs, so when build RUP method and XP method, people use basic strategy of "iterative development, step-by-step perfection".

Why the basic strategy of "reuse, component assembling" is used in some software development method? This is because: Development efficiency is an important issue, it is emphatically considered. In the development, if existing software components can be reuse, then development efficiency can greatly improve. When there are a lot of software components that can be reused, people can use them, in developing target software. So time and manpower is saved, the increase of developing efficiency is natural. Under such a prerequisite, when building software development method, of course people can consider using basic strategy of

"reuse, component assembling". For example, "class" is a software component, after object-oriented technology is widely used, the number of "class" is accumulated, so the people build some libraries, to store a lot of "class", they can be reuse. On this basis, a variety of object-oriented software development methods have use basic strategy of "reuse, component assembling".

Why the basic strategy of "split up, divide and conquer" is used in some software development method? This is because: This basic strategy is applicable for large and complex problems. In any field it is applicable, software development is no exception. As the scale of target software is increasing, in building new software development method, it is an inevitable choice to use basic strategy of "split up, divide and conquer".

As for the basic strategy of "according to linear order, stages development", when building software development method, any software development method should include it. The reason is: the development activity no matter how complex, development process of any software development method should be divided into several phases, according to linear sequence to do work of each stage. The difference is: Because the type of software development method is different, so the content of each stage is different.

From the above analysis, we can see: each basic strategy has its characteristics and scope of application. How to build new software development methods to use these basic strategies? To do so is better: according to their own characteristics of each a basic strategy, combined with specific development technology and development environment, selecting a few basic strategies, and together combining them, to form a new software development methods.

IV. THE BASIC STRATEGY AND DEVELOPMENT SCHEME OF SPECIFIC PROBLEM

When to design a development scheme for specific problem, how to use these basic strategy? Through a land resources information management system, we illustrate.

For the land resources information management system, the users only put forward a number of basic needs, and detailed needs are vague and incomplete. According to the existing user's basic needs, after repeated discussions, we decided to use the basic strategy of "split up, divide and conquer", "iterative development, step-by-step perfection" and "according to linear order, stages development", for the design of development scheme of land resources information management system.

In the mass, we apply the basic strategy of "split up, divide and conquer". We divide the system classified as five sub-systems. They are: the cadastral information management subsystem, the planning construction information management subsystem, the dynamic monitoring subsystem, the developer information management subsystems, the information dissemination subsystem. First, the cadastral information management subsystem was developed, and then the other subsystems were developed.

For each subsystem, we apply the basic strategy of "iterative development, step-by-step perfection". Based on existing basic needs of user, first of all, we can build a prototype system. Then after several iterations, getting all the user needs, prototype system is gradually improved. Last, the subsystem is complete.

For each iteration development process, we apply the basic strategy "according to linear order, stages development". Each iteration development process is divided into four stages, they are: planning phase, design phase, coding phase, testing phase. According to linear order, from a stage to a stage, development task is completed.

V. CONCLUSION

Above several basic strategies, their formation is based on the technology and ways of thinking of the past. For example, the basic strategy of "reuse, component assembling" is based on object-oriented and a large number of software materials that is accumulate over the years. These common basic strategies are the basic elements to build software development methods. These basic strategies are combined in different ways, forming variety of software development methods existing.

Based on these basic strategies, according to application areas and nature of problem, how to build more suitable software development method? This is an important research question in software methodology. The future, when new development technologies and ways of thinking appear, then

appearing new basic strategy is inevitable. Based on the new basic strategy, we can assemble new software development methods. Therefore, to build new basic strategy is too an important issue that software methodology should study.

ACKNOWLEDGMENT

This research is sponsored by Natural Science Foundation of Zhejiang(No. LY12F02021).

REFERENCES

- [1] LIN Zheng-kui, YANG De-li, "Software Development Model Research: Survey, Problem and Challenge", Application research of computer, vol.3, pp.6-9, 2005.
- [2] Zhang Yousheng, Li Xiong, "Software Development Models: a Survey", Computer Engineering and Application, vol.3, pp.109-115, 2006.
- [3] Ian Sommerville, Software Engineering(6th Edition), MA: Addison-Wesley, 2001, pp.45-46.
- [4] Zhang Haifan , Introduction to Software Engineering (5th Edition), Beijing: Tsinghua University Press, 2008, pp.16-17.
- [5] Roger S Pressman, Software Engineering: A Practitioner's Approach (6th Edition), New York:McGraw-Hill,2005, pp.79-81.
- [6] Wu Jie-Ming,Yuan Shan-Long, Software Engineering Application Practice course, Beijing: Tsinghua University Press, 2003, pp.9.
- [7] Jacobson I, Booch G, Rumbaugh J, The Unified Software Development Process (Third Edition), MA: Addison Wesley, 2004.
- [8] K Beck, Extreme Programming Explained, MA: Addison Wesley, 2000.