Advanced Engineering Forum *Online: 2012-09-26*
ISSN: 2234-991X, Vols. 6-7, pp 3-8
doi:10.4028/www.scientific.net/AEF.6-7.3

Factors to be Considered When to Design Software Development Plan

Yu Gao[1, a], Xiangzhong Feng[2,b]

[1]Department of Computer, Zhejiang Ocean University,
Zhoushan, Zhejiang, P.R.China 316000

[2] College of Mechanical and Electrical Engineering,Zhejiang Ocean University,
Zhoushan, Zhejiang, P.R.China 316000

[a]gaoyu@zjou.edu.cn, [b]fxzfeng@yahoo.com

Keywords: devise, software development plan, software engineering

Abstract. It is an important prerequisite to ensure the success of software development that designing a reasonable and suitable software development plan. In the past, when discussing the design of the software development plan, people focus limited to a specific development tasks, so the lack of universality. In order to guide design work of software development plan, starting from the perspective of software engineering, the main factor to be considered has been studied when designing software development plan. These factors are: the type and size of the software, the experience of use for reference to predecessors, difficulty level to obtain users' needs, development techniques and tools, situation of development team, development risks, the software development methods that can be chosen. This study results can improve the rationality and applicability of the software development plan.

Introduction

To face a task of software development, people need to design software development plan. Then, the software development activities are carried out in accordance with the software development plan [1,2]. The importance of software development plan is well known. If the software development plan is reasonable and applicable, then a good foundation was laid for the success of software development. Otherwise, if the software development plan is unreasonable or inapplicable, even people put more effort and spent more the cost, then software quality is not guaranteed, and even software development is unsuccessful. Therefore, the design of the software development plan needs to be done seriously. For software development programs related to the factor, people must think carefully.

When discussing software development plan, previous research articles are usually to aim at a specific development task. These articles discuss the design for specific software development plan [3,4,5]. The contents of discussion are often limited to a specific development environment. The comprehensive study has not carried out in these articles. Therefore, these research results lack of universal significance. From the perspective of software engineering, for the design of the software development plan, we did comprehensive study. We strive for a universal significance for the results of the study, so that to guide the generic development work. On this basis, we propose a number of primal factors to be considered, when designing a software development plan.

The following describes the content of our research.

The main factors to be considered

When designing a software development plan, many factors are involved. Under the guidance of software engineering theory [6], and after combing, we consider that main factors to be considered should are the type and size of the software, the experience of predecessors, the degree of difficulty of obtaining users' needs, the development techniques and tools, the condition of development team, development risks, software development methods to be selected. In the following, we discuss these factors.

F**irst Factor to be Considered: The Type and Size of the Software.** Confirming the type and size of the target software has important significance. The reason is that difference is very large between software of different types and sizes. Their architecture is different, and their amount of data store is different, and their pattern processing data is different, and their operating environment is different, and their serving objects are different, and their development methods, techniques and tools are different. Therefore, to software of different types and sizes，software development plan has very big difference. Only after the type and scale of the target software has been determined, other design factor can be considered. For example, how to obtain the users' needs? How to organize the development team? How to avoid the risk of development? How to choose a software development method? And so on.

So, when designing a software development plan, the type and size of the target software should be foremost factor to be considered. That is to say, people should first analyze the type of goals software, and estimate the size of the target software. Only after the type and size of the target software is make clear, and other follow-up work can take place.

If the designer of software development plan is unfamiliar with the development field, in order to understand the type and size of the target software, then the designer of the software development plan must consult experts in the field. In fact, in many cases, the designer of the software development plan is not clear for the expertise of development field. After consult experts in the field, under the premise of understanding the expertise of development field, we can analyze clearly the type and size of the target software, and can reasonably design software development plan.

Second Factor to be Considered: The Experience of Predecessors. So far, the large number software has been developed. Type of software is various. When developing software, in order to improve development efficiency and quality, we can take example by the experience of predecessors. Therefore, after to determine the type and size of the target software, we should investigate situation of developing similar software, and collect the software development plan of similar software and other information. Take example by previous experience can improve efficiency and avoid detours.

Note: Take example by previous experience is not a simple re-use, and is not copying someone else's software development plan. The general approach is: to collect as much as possible of similar the software development plan, analyzing and comparing them, to analyze clearly the advantages and disadvantages of each software development plan, and then combined with the specific situation of the target software, in the basis of these software development plan, a better and suitable software development plan was formulated.

Third Factor to be Considered: Degree of Difficulty of Obtaining User Demand. For software development succeed, one of the important condition is able to obtain real and detailed user requirements. Therefore, when designing software development plan, how to obtain the user demand is important factor, this factor must carefully think about.

In general, when to obtain user's needs, there will be three cases:

(1) The user's demand information is stable and clear, developers spend a relatively short period of time obtaining all user demand information. To obtain all user's demand information is one-off, repeat of work is unnecessary.

(2) The user's demand information is stable, but is not clear. To obtain user's demand information can not be once in a short time to complete. Developers need to work several times, each time to obtain the user's part of the demand information, and developers can gradually obtain the user's all demand information.

(3) User's demand for information is the instability, uncertainty and change frequently. In a short period of time obtaining all user demand information one-off is not possible. The user's demand information may change. Developers need to work many times, and repeat work.

To be based on the objective situation the designer of the software development plan should predict the degree of difficulty to obtain user demand information. Then, take example by previous experience, the designer design a practical software development plan.

In the before developing software, due to any case are likely to appear, so, for each of the above, the designers should give the response measures when the design of software development plan. In either case, there is a solution. Especially, for the aforementioned second and third, the designer should consider carefully.

Fourth Factor to beConsidered: The Development Techniques and Tools. When the design of the software development plan, in the after to determine the type and size of the target software, designers need to think about an important factor. This important factor is the techniques and tools of development.

On the techniques and tools of development, designers should first think about which the techniques and tools of development may be used when developing the target software. In the system analysis phase, system design phase and system implementation phase, all techniques and tools of development are included.

Secondly, for a variety of techniques and tools of development, designers of software development plan should be aware of the status of each technique and tool of development. For example, for the systems analysis phase, these should be clear: what kinds of techniques and tools of development may be used? What are the advantages and disadvantages of each technique and tool of development? What is the scope of each technique and tool of development? Similarly, for the system design phase and implementation phase, the status must be clearly.

Thirdly, Techniques and tools that may be used for each stage development should not only know its current situation, but also think about its future development trends. The reason is: Techniques and tools of development are constantly evolving. Prospect of some techniques and tools of development are better, future their vitality is relatively strong. Prospect of some techniques and tools of development are not optimistic, and may even be eliminated. We should apply those techniques and tools of development that prospect is good and vitality is strong.

Last, on how to select the techniques and tools of development, in the design of software development plan, designer should put forward a reasonable proposal, and explain reason of proposal.

Fifth Factor to be Considered: the Development Team. No matter how good a software development plan, if there is not a strong development team, then, software development plan is empty. Therefore, after to determine the type and size of the target software, an important factor to be considered is the condition of development team.

Type and size of software is the most important factor determining manning of the development team. That is to say, to be based on the type and size of the target software, to consult past experience of developing similar software, we determine manning of the development team, as well as number of people of the development team. For example, if the target software is a financial information management system, then the development team need person to know the knowledge of financial information management. If the target software is network software, then the development team need person to know the knowledge of network software development.

Designer of software development plan must be aware of the status of existing development team of company. How much development technical staffs have the company? How many people have the company to do system analysis? How many people have the company to do system design? How many people have the company to write programs? How many people have the company to test? Has each developer the experience to develop similar software? Do they know well the condition of development field? How their ability? Have they mastered the development technologies and tools to be use? Do they need to be training?

If the number of developers of the development company is insufficient, then cooperation with other development companies needs to be considered or hiring some developers. This ensures that the number of the development team is sufficient.

Sixth Factor to be Considered: Development Risk. In the software development process, software development risk is the loss or injury caused as a result of uncertainty. To development risk that may exist, designers of software development plan should able to predict, so that future to identify the development risk, to assess and to control development risk.

According to the software development process, to combine with the specific situation of the target software, designers of software development plan should predict development risk for each stage.

On the analysis phase of the target software system, the main task is problem definition, feasibility study and needs analysis. At this stage, the possible risks include: Problem definition is unfaithful. To describe the function of the target software system is incomplete or contradictory. The development plan does not accord with the actual situation.

On the target software system design phase, the main task is to design the software system architecture, to design algorithms and data structures of each module, and so on. At this stage, possible risks include: The design of the system architecture is unreasonable. Due to the lack of customer participation and expert review, the problem to appear is not promptly to corrected. Due to the lack of change control, the design is free modify.

On the implementation phase of the target software system, the main task is coding and testing. At this stage, possible risks include: The wrong design can not be encoded to achieve. Due to the ability of developer is insufficient, resulting in the encoding can not continue. The modules can not be integrated. Sudden major the change of project appear. The test found a significant error, resulting in a lot of work needs to re-do.

In addition, for each stage of the above-mentioned, there is risk of rate of progress, financial risk and human resource risk. Such as: Scheduling of arrangement may be unreasonable. The time limit for a project may have been delayed. Capital budget may be unreasonable. The funds may not be on time in place. Developer may be not enough, Development team may be unstable.

The impact degree of development risk can be divided into different grades. According to the actual situation, the designer of the software development plan should predict the likelihood of presence and impact degree of various risks, and give to pre-arranged planning of reducing, avoiding and eliminating the risk at the same time.

Seventh Factor to be Considered: Software Development Methods to be Selected. At present there are a variety of software development methods [7, 8]. When developing software, selecting software development method is absolutely not simple and casual. When developing the target software, for the designers of the software development plan, an important factor must be thinking. Important factor is: In the existing software development methods, which development methods are that you can choose? In the development methods that you can choose, which one is the best software development method?

How to choose a software development method? Selecting software development methods based on a number of factors, the more important factors include: the type and size of the target software, condition obtaining user demand, the main development technology. The application scope and field of each software development method is different. For example, the RUP method is applicable to the development of large-scale software, agile methods are applicable to the development of small and medium-sized software, formal methods are applicable to the case of particularly high reliability, rapid prototyping method is suitable for that user demand is vague and variable. On the basis understanding content of various methods, we can choose a few of the more suitable software development methods. After comparing them, we can determine one most appropriate software development method.

After determining one most appropriate software development method, we can design software development plan, according to content of the software development method. The question to be noted is that the contents of the software development method must not be completely copy. This is because: for a particular software development method, which is based specific techniques and tools at that time, which is based practical activities at that time, its contents only is a summary at that time. A number of years have passed, the situation has undergone many changes, its some content may not be applicable, and therefore can not copy the contents of the software development methods.

In addition, for the arrangement of the development process, any a software development method is not detailed, designers of software development plan need to refine the development process, to combine with specific development problems.

Practical Application

The following is a brief introduction to the practical application of the above studies. Certain company needs to develop software, when we take part in design of the software development plan, we think carefully about the seven factors listed above.

On the type and size of software, after the analysis of user requirements and further investigation, we determined that it is a network database management software, it is a small-scale software. On the previous experience, to the software of this type and scale, and there are many such software, and there are many successful experiences. To collect their data is easy. On the degree of difficulty of obtaining user demand, after communication with the user, we determine that user's demand information is stable, but is not entirely clear, we can gradually obtain user's demand information. On the development techniques and tools, on the basis of type and size of the target software, comparison of development techniques and tools to be selected, we confirm using Dreamweaver and Access to develop software. On the development team, for software companies to undertake development tasks, the number of developers is sufficient, and it has all types of developers. On the development risk, this software company has developed this type of software, so there is no technical risks, the possible risk is the human resources risk (developer removed from or illness, resulting instability in the development team, the progress of the development may be delayed.). On the software development methods to be selected, on the basis of condition of various aspects, we determine the use of Extreme Programming method, on the basis of Extreme Programming method, to do some modifications, to form specific software development methods. On the basis of thinking carefully these factors, we designed a software development plan.

Conclusion

In the above, we discussed seven factors to be selected in the design of software development plan. The seven factors have associated with each other. When the designers think about these factors, their order should be noticed. In the above, the order that we discussed to various factors should be the order of thinking of designers of the software development plan. The type and size of the software should be first factor to be selected. It is the basis of other factors. It is the most important factors affecting the design of software development plan. After thinking clearly to the seven factors, when we design a software development plan, the rationality and applicability of the software development plan clearly can raise.

Acknowledgment

This research is sponsored by Natural Science Foundation of Zhejing(No. Y12F020079).

References

[1] WU Yuan-bin, A Methodology for Agent-oriented Analysis and Design, MINI-MICRO SYSTEMS, 24(2003)456-459.

[2] Lu Xudong Wan J iancheng, Study of Development Methodology in Web Applications, Computer Engineering and Application, 13 (2006)97-101.

[3] GU Zhen-li, Research on prototype development model in multi-media instruction software, Computer Engineering and Design, 27(2006)1828-1830.

[4] Wang Jicheng, Research on Agent-Based Network Management Modeling, Computer Applications and Software, 21(2004)57-59.

[5] Li Chao, Zhang Jing, Liu Lu, Research on Modeling of Agent-Oriented Intelligent Procurement System, Computer Engineering and Application, 23(2003)221-223.

[6] Roger S Pressman. Software Engineering: A Practitioner's Approach. Sixth Edition, McGraw-Hill, New York, 2005, pp52-126.

[7] Zhao Chi-Long, etc., Practical Software Engineering, Electronic Industry Press, Beijing, 2009, pp7-16.

[8] Yin Ren-Kun, etc., Practical Software Engineering, Tsinghua University Press, Beijing, 2010, pp13-26