

The Practical Application of Software Engineering to Graduation Project (Thesis)

Ping Xu

School of Information Science and Technology
Taishan University
Taian, China
taianxp@163.com

Deyun Yang

School of Information Science and Technology
Taishan University
Taian, China
yangdeyundisk@163.com

Abstract—This paper introduces the application of software engineering to graduation project (thesis), and starts with the discussion about topic selection, thesis proposal, system design and testing, the compiling of software documentation, thesis writing and other aspects, so as to ameliorate the current problems existing in graduation project (thesis) and improve its quality.

Keywords—*Graduation project (thesis); Practical teaching; Software engineering; Software life cycle*

I. INTRODUCTION

Graduation project (thesis) is the last important step of the teaching practice in the training program of computer specialists. It serves as a comprehensive review of the comprehensive application of computer knowledge and the last practice assessment of college students before they go to the workplace; it is also an important evidence for recognizing students' graduation and degree qualification, as well as an important approach to the discovery and recommendation of talents.

In recent years, with the expansion of colleges and universities, graduation project (thesis) is exposed to a variety of problems. The quality of graduation project (thesis) declines year after year. Because students are anxious to find a job or prepare the postgraduate entrance examination, they pay insufficient attention to graduation project (thesis), which results in the lack of demand analysis, defective design result, chaotic software development process, lack of document writing training, etc. In view of the above problems, how to reform and improve the practice of graduation project (thesis) in the new situation and enhance its quality is an urgent issue to be solved.

II. SOFTWARE ENGINEERING

A. Software Engineering

In general, software engineering is an engineering discipline that guides the development and maintenance of computer software. It adopts the engineering concepts, principles, techniques and methods to develop and maintain software, and combine the time-tested and correct management technology with the best available technology at present to economically develop high-quality software and conduct effective maintenance of it[1].

Software engineering methodology consists of three elements: methods, tools, and process. Methods refer to the technical methods to complete various tasks for software development, and it answers the question of "how to do"; tools refer to the automatic or semi-automatic software engineering support environment provided for the usage of methods; process refer to the framework of a set of tasks that needs to be completed to obtain high-quality software, and it specifies the steps to complete the tasks. Currently, the most widely used software engineering methodologies are the traditional methodology and the object-oriented methodology respectively[1].

B. Software Life Cycle

Software life cycle consists of three periods, i.e. software definition, software development and operation and maintenance, each of which is further divided into several stages. The definition period includes three stages, namely, problem definition, feasibility study and demand analysis. The development period contains four stages: overall design, detailed design, coding and unit testing, and integration testing. The former two stages are also known as system design and the latter two system implementation. The main task of the maintenance period is to enable the software to meet the needs of users permanently.

The whole process of graduation project (thesis) can be seen as the process of developing large-scale software. In order to improve the quality of graduation project (thesis), it is necessary to be guided by software engineering, carefully complete the tasks in various stages of the software life cycle, so as to ameliorate the current problems in graduation project (thesis). The

corresponding relationship between the various stages of software life cycle and that of the graduation project (thesis) of School of Information Science and Technology of Taishan University is shown in Fig. 1.

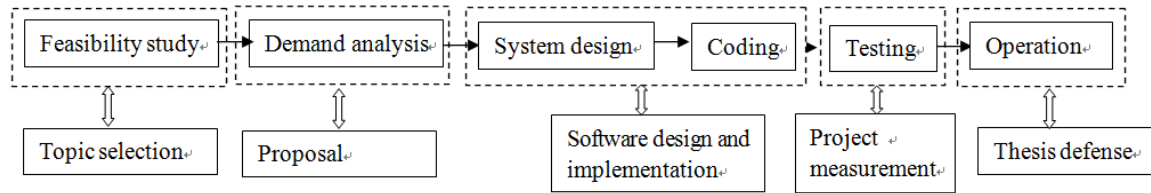


Fig. 1. The relationship between software life cycle and graduation project(thesis)

III. PROBLEMS IN THE GRADUATION PROJECT (THESIS) OF COLLEGE STUDENTS

A. Ignoring Feasibility Study and Choosing Inappropriate Topic

The task of the feasibility study stage is not to solve the problem, but to study the scope of the problem, explore whether the problem is worth to address, and whether there is any viable solution. During this stage, high-level logic model of the system should be derived. On this basis, the scale and objectives of the project need to be determined more accurately and specifically. However, many students ignore this stage, believe that the feasibility study is of little significance, and eventually make improper selection of topic, which mainly shows in the following areas:

1) *The scope of topic is too broad:*The system modules that are to be implemented by the developed software are too numerous. As a result, many students only make a few pages of the system or only implement part of the module.

2) *Repetition of topics:*Currently, many students, not fully aware of the importance of graduation project (thesis), pay insufficient attention to graduation project (thesis). They only consider the step of graduation project (thesis) as a necessary part of college life, and prepare graduation project (thesis) just to complete their studies. Under this premise, the phenomenon in which different instructors guide students to do the same subject (or some contents are the same) emerges. What's more, there are some students who just make slight changes in others' research for their own use in order to save time and energy.

3) *Lack of objectives:*Since many students do not have a deep understanding of the meaning of graduation project (thesis), when choosing topics, they are often perfunctory to finish the task itself, without any specific objective.

4) *Lack of practical value:*The selected topic, which departs from practice, fails to be in line with social applications. Since many students lack of practice and cannot participate in the actual research projects, they can only choose some virtual topics. Therefore, the system business logic that is completed is very simple, without any practical value.

B. Ignoring Demand Analysis

In order to develop the software products that truly meet the needs of users, we must first access to their needs. A deep understanding of the demands on software is a prerequisite for successful software development. The software that does not really meet the needs of users will only make them disappointed, regardless of how well the design and coding work is done.

However, many students think that the most important step in the process of graduation project (thesis) is programming. Therefore, they are unwilling to carry out demand analysis of the system; research is not sufficient in the stage of demand analysis; students do not comprehensively and accurately access to users' needs, or only focus on the needs for the function of the software, ignoring the performance demands, the fact of which leads to poor availability of the software.

Some students skip over the step of demand analysis and directly enter the stage of system design. The entire system process lie in a chaotic state of "design - analysis - modify - design", which causes that the overall system is chaotic and cannot be developed further[2].

Other students write procedures while analyzing demands. Once there emerges any demand which is not identified in advance but in the process of preparing the program, students have to rewrite the program, which undermines the original structure of the program and brings great difficulty to the operation and maintenance of the program.

C. Ignoring System Design

The basic task of system design is to make scientific and reasonable overall design and detailed design of system on the basis of demand analysis according to the requirements of the logic model, so as to provide the necessary technical information for the realization of system in the next stage. In terms of graduation project (thesis), most of the students do not make an overall planning of the system in advance, without clear objectives, thus, they just carry out the work randomly according to what come into their mind. Some students even start the work of the next stage before finishing that of the previous stage, and ignore the basic principles of system design. This disordered design can only produce chaotic system structure, repetitive development of the program code, and ultimately lead to the poor integration performance of system. It is only too late when the problem has been found. Any little change will make the system collapse.

D. Ignoring System Testing

Some students always think that graduation project (thesis) completes when the program is finished, without thinking that a complete graduation project (thesis) also need to be strictly tested after the system has been implemented. Because students do not attach importance to the testing process, the system errors occur constantly, to the extent of irreparable situation. Some students only conduct the unit testing for module, believing that the high cohesion of module is bound to result in stable and reliable integration. Other students randomly select a small number of test data and usage cases, and when they find that the results of the program match with what is expected, they thought that the program is a success.

Still other students directly ignore the test after the completion of coding. During thesis defense, when the system runs, different errors occur in many places, or even the program cannot run, which greatly reduces the quality of graduation project.

E. Ignoring the Preparation of Software Documentation and the Writing of Graduation Thesis

Software not only refers to the program, but also needs to be provided with relatively complete documentation. Software engineering requires that the documents of the system development process must match with the software that has been developed. It should clearly explain what has been done for the system development in a certain stage, and what should be done in the next step, for example, demand specification should be produced in the demand analysis stage, and the corresponding design specification should be formed in the design stage. At the same time, documentation is also the determinant of software maintenance, because unqualified and incomplete documentation brings a lot of inconvenience to the internal communication between developers at work. More seriously, if there is no accurate documentation, a lot of work, such as modifying errors in the system, improving and adding new features, porting to other software environments, etc. will be very difficult to do.

However, in the process of graduation project (thesis), a considerable number of students fail to do this work. When errors occur after the program runs, they do not know where and how to modify. Some students think that the preparation of document is a waste of time, and they are more willing to spend energy on how to debug a procedure. In addition, while completing the graduation project, students should write a thesis under the guidance of the instructor. But many students do not begin writing until the final stage of graduation project. With lack of invested time and energy, the thesis is of low quality and the problem of plagiarism is serious.

IV. APPLICATION OF SOFTWARE ENGINEERING

In order to improve the quality of graduation project (paper) and cultivate the software talents meeting the needs of the community, it is necessary and urgent to apply software engineering to graduation project (thesis) and to guide and standardize the graduation project (thesis) through applying software engineering[3].

A. Application in the Process of Topic Selection

The selection of the topic of graduation project (thesis) should be based on solving practical problems. In terms of difficulty, it cannot involve unfamiliar and difficult application areas, while in terms of time, it should be controlled to complete in 12 weeks or so.

In order to better conduct the feasibility study, we have reformed the way of selecting topic. When students make selections, the instructor does not give a specific topic, only to give the direction of the study and the number of students that are to be guided. Therefore, students can determine their own topic, which gives them more autonomy and stimulates their innovative thinking. At the same time, innovation has been made in the content and form of the topic, for example, for students who take internship in enterprises, it is suggested that they choose a topic related to the training content; for students who make graduate design (thesis) in school, they are encouraged to participate in scientific research team, college students science and technology competition and all kinds of innovative experimental projects, all of which provide students with a good practice platform. No matter what kind of topic selection is, it serves to allow students to have a more clear understanding of the meaning and scope of the topic, in order to better complete the graduation project (thesis).

B. Application in the Proposal Process

After the topic of graduation project (thesis) is selected, students should, under the guidance of demand analysis, further elaborate and analyze the task and objective to be undertaken by the topic, collect and collate data, and clarify users' demand. In addition, as the project progresses, they also need to consider the changes in demand. There will be some new demand or some changes in the original demand. For the changes in demand, first is to make classification, and determine whether the change will have an impact on the overall design. During the system development, try to avoid the changes affecting the overall design, because it needs to modify the system architecture, re-design and implement the system, which will extend the development time and cycle of system.

When obtaining the demand, it need to model the process of demand analysis, during which students can understand the system, and sometimes it is necessary to construct the prototyping of the complex system interface, man-machine interface, etc. At present, in terms of graduation project (thesis), the traditional methodology and object-oriented methodology can be used to achieve this

goal. Through abstraction and generalization, derive the target system's logic models which are shown graphically to be identified by the user. The system can be described through these models. In fact, model is not only the basis of software design but also the basis of preparing the requirements specification.

C. Application in the Design Process

System design, which determines how the software system works, is the technical core in the software engineering process, as well as the basis for the follow-up development and maintenance. The system design includes the overall design and the detailed design, mainly focuses on the software architecture, data structure, user interface and implementation of the design, etc. At this stage, it also needs to write the corresponding design specifications.

In the overall design stage, it needs to draw the data flow diagram according to the requirements of the software engineering, make a detailed description of the input and output data flow, establish the data dictionary in line with the software engineering specification, and design the software structure diagram according to the data flow diagram. During this period, the instructor is responsible for questioning the design that does not meet the requirements and the unreasonable structure, and ask the students to solve the problems independently. In the detailed design stage, students should specify the solution, and design the module's internal details; when making the design of object by using the object-oriented method, they have to determine the algorithms of the class and correlation of solution space, the form of interface and the implementation of service.

D. Application in the Test Process

The purpose of software testing is to systematically identify the potential errors in the software with minimal effort and effort. When the results of detailed design are translated into the source code, testing should start as soon as possible. For graduation project, there are mainly three forms of testing: unit testing, integration testing and system testing.

In terms of unit testing, the programmer, after completing the coding of a module, should conduct the corresponding white-box test and black-box test for this module. Only after passing the test can the module code be marked as completed. Integration testing is a kind of test in which the modules which pass through the unit testing are assembled to form a component or subsystem by a selected strategy in order to be tested, to ensure that the new component or subsystem can be properly implemented. System testing is mainly used to test whether the overall function of the entire system is correct and whether it conforms to the user's original requirements. Only when the software passes through the system testing can it be the final software product.

When testing, it needs to carefully design the test case for the program to be tested, and select the test data needed by system requirements, design documents, program code, boundary data and some random data, taking into account the reasonable and unreasonable inputs. During the testing process, students should pay attention to the wrong cluster phenomenon, focus on testing the modules for which many errors have been found out. Students are suggested to use the automated testing software tools as much as possible. If the test fails to find any error, the test plan and the test case need to be reconsidered and redesigned. After the test, analyze and compare the measured results with the expected results, debug the errors occurring and modify the corresponding documents. The modified program should be tested again until it is satisfactory.

E. Application in the Process of Documents and Thesis Writing

Many students think that graduation project equals to code writing, and the writing of document is not necessary. This understanding is incorrect, because software documentation is a guide to the specifications of software development, and graduation thesis is the summary and improvement of graduation project, which enables the readers to have a comprehensive understanding of the design principles, solutions, key technologies, etc. of the subject. The first step of thesis writing is to determine the topic, review relevant data, then write a proposal, make the system design, complete the first draft of the thesis, revise and present the final thesis. In the process of graduation project (thesis), students are instructed to write the standardized software documents by stages[3]. On the one hand, it is helpful in planning and organizing the software development process scientifically. On the other hand, it can extract the essence to be presented in the relevant part of graduation thesis.

V. CONCLUSION

Graduation project (thesis) is the last important practice of college students. The application of software engineering to graduation project (thesis) through strictly abiding by the basic principles of software engineering greatly ameliorates the existing problems in current graduation project (thesis), and improves the quality of graduation project (thesis). At the same time, it facilitates the transition of "software engineering" curriculum from theory to practice, enable students to make a relatively complete generalization and summary of the knowledge learned about computer before graduation, and accumulate certain experiences in project development.

REFERENCES

- [1] H.L. Zhang and Y.M. Mu: Introduction to Software Engineering (Tsinghua University Press, China, 2013) , In Chinese, p.5.

- [2] W.J. Zhang, H.X. Lu, P. Yang, L.G. Zhang and Y.F. Liu: Journal of Agricultural University of Hebei(Agriculture and Forestry Education), In Chinese , Vol. 9 (2007) No.4, p.70.
- [3] H. Ye and B. Jiang: Computer Education, In Chinese , Vol. 9 (2014), p.77.