Software engineering reflection essay

This semester has studied the software engineering course. Through the teacher's diversified teaching, not only let me master the software engineering in theory, but also from different examples, the theory and practice have been well combined. Throughout the semester, I have learned a lot of things in general. There are many places that are worthy of recognition. In fact, in my opinion, software engineering is not so much a course as it is an idea. It is a process of how to analyze and deal with problems. It should be said that its scope is far more than just a course, and it has become a comprehensive collection of ideas that can solve problems.

Topics covered in the required courses for this course

First of all, I will describe what we have learned. First, we will analyze the development of the software engineering discipline and the social environment. Then we will analyze the software opening process and mode, software project management, and computer engineering in the following chapters. , build use cases, requirements analysis, structured analysis modeling, and UML object-oriented analysis modeling. Then I will give a detailed introduction to my understanding of the knowledge points of this course: software is an executable computer program that

can perform predetermined functions and functions, and the data needed to make the program execute normally, plus describe the operation and use of the program. Documentation. The characteristics of the software: 1 software is a logical entity, not a specific physical entity, so it is abstract. 2Software is a product that transforms knowledge and technology into information through people's intellectual activities. 3 software becomes a product, its production is just a simple copy, unlike hardware manufacturing. 4 Maintenance process is more complicated than hardware, and even lead to new errors. The software life cycle can be divided into eight stages: 1 problem definition; 2 feasibility study; 3 demand analysis; 4 overall (summary) design; 5 detailed design; 6 coding and unit testing; 7 comprehensive testing;

Technical challenges and solutions for developing software products

Our team developed a web-based online recruitment system this semester. For our group (including myself), we also encountered a lot of technical issues. Below I will introduce some of the problems I encountered and my solution.It takes a lot of time to reconfigure the system architecture. Since the customer is required to run the system in a distributed environment, the development is often on a single machine, and there is not enough time to sketch and slowly design. The work

arrangement is often The layout of the system is sometimes full of adjustments. If the architecture is not clear and the ideas are not rigorous at the beginning of development, the whole project will be chaotic and the system architecture will not be able to withstand the system architecture. Refactoring, of course, the architecture refactoring here is more of a minor adjustment. If it is really a big adjustment, it means that the architecture at the beginning is very unsuccessful. The project is not developed by one person. If it is a personal development project, it is better to say that I want to adjust and adjust. Now it is a multi-person development project. Although it can't be compared to an aircraft carrier, at least like a frigate, it is not so easy to turn around. My solution is to determine the refactoring, then the goal should be clarified, that is, the boundary conditions of the refactoring, clearly outline the points that need to be completed in this refactoring. The goal is to quantify the data (such as the number of lines of code is reduced to In the past half, code execution time has been reduced to 30 percent in the past, etc.), and the refactored code can be effectively tested. Before refactoring, there needs to be a process of requirement analysis. If the requirements are not clear, the reconstructed PRD cannot be clearly written, and the team responsible for refactoring will have difficulty in having a clear goal. In particular, when the refactoring work is performed by a team, all members must agree on the goals of the refactoring.

The most practical technology I have learned this semester

For me, in the study of software engineering in this semester, the most important technology learned for our team is the demand analysis, because the early stage of system development needs analysis is very important, it is to effectively solve the user's problems. For engineering activities, the requirements that need to be considered are functional requirements, data requirements, performance requirements, and interface requirements. The developer assumes the analysis task and the core is the user. There are three steps: 1 to obtain customer needs, customers refer to a person or institutional department, etc. The general method is to investigate, including interviews, seminars, questionnaires, follow-up and collection of information, and the requirements specification can express the user's software value. 2 Establish a demand model, which is a graphical representation of user requirements. Some commonly used models are: business tree diagram, use case diagram, and activity diagram. Used for structured requirements modeling, system business examples, and system workflows. 3 Perform requirements verification. The main contents to be verified are: validity verification, consistency verification, integrity verification, reality verification and verifiability verification. On the other hand, I think that the technology we have learned this semester is very practical and will definitely play a very important role in our future career.

Define the choice of a new project

According to my experience, although the project we are doing now is basically completed, it will have many shortcomings, which deserves our repeated thinking and improvement in the future. For example, in the early days of the project, some of the system use cases and functions we created have not been implemented. Although we will not redefine our project, we will fix and upgrade system bugs on the basis of the current and future learning. Of course, new projects will be developed and defined after we reach a good level.

Challenges encountered

As a member of the team, to be honest, this is a difficult semester, and I have encountered a lot of problems with our system. Fortunately, we did not give up and tried our best to persist. For example, in the process of creating a database, there will be some thinking, mature database design routines. In fact, database design is also a matter of learning. It seems simple. If you really want to design well, you need to have hard work, and you need skill and skill. superb. The database is basically an indispensable part of the current development of various management systems, and even now is the cornerstone of a stable management information system, so whether the database design is reasonable, at least 30-40% of the project is smooth and stable. On the other hand, the code is

also racking its brains, because the code specification, code quality check, often also bear the post-maintenance, and even part of the running work, if the project's code quality is not good, there will be endless pain in the late stage, put some problems Killing in the cradle is better than cultivating the problem, and then eliminating the trouble and the head. Therefore, the code must be strictly checked in the middle and late stages of the project. Many of them have SQL injection vulnerabilities, code with repetitive functions, and naming. Unreasonable code and so on will still be found a lot.

Personal experience benefits

I am very grateful to this semester of software engineering for my experience, it goes without saying that such personal experience will add a touch of color to his life. Next, I will simply talk about my simple experience in this class. We are junior students. When I was in class, I still listened very seriously to what the teacher said. I think his thoughts and The understanding of the comprehensive quality of computer students that I have always come to is still coincident to some extent. Secondly, this course teaches us the general procedures and processes when we complete an actual project. I think this is a very practical teaching content. When we graduated, this is a very useful skill that we actually need to use, and it is not limited to the scope of software

engineering. Even if we are engaged in other industries, we must not start from the demand, and always have a reasonable understanding. Is the completion of the final product? It should be said that this is the value of this course.

Other knowledge that is helpful for project development after learning

As our students, we should take on the main responsibility. There are too many basic courses in college, most of the basic courses are boring, maybe we can still be fresh in the first semester, but after 5 semesters, it can be said that it is more difficult to have a sense of freshness. Things, we have all begun to become dull. Second, I did not realize the value of this course. The value of this course I have already said above is self-evident. But not every student will be engaged in the computer industry after graduation, and not every student knows that the significance of this course is not limited to the computer category. Maybe some people think that this is not the direction of development in the future, and they don't care about this course. I personally feel that this course is really good. If you study hard, you will learn a lot. The practical thinking and the logical thinking from the whole general analysis system will be obviously enhanced, no matter what kind of work you will do in the future. For the future, it is a great hidden wealth.

Course evaluation

In general, the software engineering curriculum emphasizes the understanding and mastery of basic concepts and basic knowledge, as well as the basic skills of analysis, design, implementation and maintenance of software projects. Pay more attention to the combination of "point" and "face". I still like this course very much. Through the study of this course, I realized that theoretical study is very important, practice is more important, practice is the only criterion for testing truth. Only by combining theory with practice can we play our role better. The role of learning knowledge can directly create benefits, society and the state contribute.