**北京邮电大学软件学院工程硕士研究生**

**专业英语成绩评定表**

**学号： 2011127136**

**学生姓名： 赵炜**

**导师姓名： 吴国仕**

**报告题目：**CONSTRUCTION OF SOFTWARE

**评分：**

**2016 年 7 月 15 日**

|  |
| --- |
| **CONSTRUCTION OF SOFTWARE**  **Process of software building**  Program here refers to the source program, that is, code. A closer look at the code, they do are some algorithms that based on the data structure. Programs also operate on the data, some of the data is static (software icon, Prompt information, etc.), some are dynamic (such as random digit generate by program, data downloaded from the internet by the program, the text or voice input by the users, etc.). But only code and static data is not enough, the engineers need to change them into executable code for the machine. Structuring is not only Cc and link command, to build a complex software not only need a reasonable software architecture, software design and implementation and Debug, but also need all kinds of files and data to describe the various procedures dependencies between files, need compiler parameters, link parameters, and so on. These are the software building process.  Members of the software team are modifying various sources every day, how to ensure that the software modification process to continuously improve the quality, to maintain at least the previous quality, not go so far as to collapse? Sometimes, we have to write some special functions for a requirement, and soon after to make all these functions merged back into the main version. Some programs you want to configure different interface, running on the operating system in Chinese, English or any other language; some programs there are 32-bit versions, 64-bit version and so on. This is the Source Code Control problem - sometimes called Software Configuration Management.  We also have a range of tools and procedures to ensure the correctness of the program, these tools and program itself should be more correct, in order to ensure the quality of other software. This is the Quality Assurance, the specific verification process is called Software Testing. A software or service you want people to buy, you have to find customers, customers have a variety of needs, some make sense, some do not; some is easy to achieve, some is difficult. Software team starts from Requirement Analysis, sort out the appropriate needs, and then gradually expand the follow-up work, such as design (software architecture), implement (write data structures and algorithms), testing, till the final software releasing. There is also employee turnover in a software team, a new member should be able to read, understand the design of the program, which is called Program Comprehension.  During operation, problems of some sort will occur, maybe we need to make a patch to the software from time to time, or to maintain a large number of servers, the new and old members of the team to work together to fix a variety of problems, it is called Software Maintenance, or service operation. This series of process is the Software Life Cycle, SLC, some people may be responsible for software Project Management. A good software, even if not very different from the similar software in function, but it will make people feel very easy to use. This is the software User Experience. User experience and data structures, algorithms are not directly related, but a lot of very successful software win in this regard.  **Problems of software development**  Software development process have any particular problems? Scholars have summarized the following five points:  1. Complexity: software is arguably the most complex type of system created by human beings. Large software (operating systems, office software, and search engine) has over one million lines of source code, tens of thousands of different files. The software engineers usually only see 30-80 lines of source code (equivalent to a display screen), their intelligence, memory is similar to the most ordinary people. There are various explicit or implicit dependencies between the various software modules, increase with the growth of the system and modules, the number of these relationships often increase at geometric rate.  2. Invisibility: Software engineers can directly see the source code, but the source code is not the software itself. Software operates at high speed in the form of machine code, also may run simultaneously on several CPU cores, designers cannot "see" their how the source code being executed on the user's machine specifically. Commercial software error has occurred, the engineer can see some traces (error code, rough object code position, and the error message) left by the program in the wrong moment, but it is almost impossible to reproduce the complete program in the end what went wrong.  3. Changeability: looks very easy to modify the software to modify the software to modify than hardware easier. It is natural to expect the software can be in the following two situations "change": a) let the software do new things; b) allow the software to adapt to the new hardware. But at the same time, the right to modify the software is a very difficult thing.  4. Conformity: software cannot exist independently, it is always on top of the hardware to run, it should be subordinated to the system requirements for other components, and it is also subject to the requirements of users, the system requirements of the industry (such as changes in bank interest rates).  5. Discontinuity: It is easier to understand continuous system: increase the input, we will see a corresponding increase in output. But many software systems have no such characteristic, sometimes small changes in input will cause great changes in output.  **The areas of Computer Science**  Computer Science (Computer Science) this academic field can be divided into the following areas: computational theory (Theoretical Computing) information and coding theory (Information and Coding Theory) algorithm and data structure (Algorithm and Data Structure) form method (Formal methods) programming language (programming language)  Partial areas of practice: computer architecture (Computer Architecture) parallel computing and distributed systems (Concurrent, Parallel and Distributed System) real-time systems and embedded systems (Real Time and Embedded System) OS (Operating System) computer network ( Networking) scientific computing (scientific computing) security and cryptography (security and cryptography)  AI (Artificial Intelligence) in this area covers a number of related fields such as pattern recognition (Pattern Recognition), machine learning (MachineLearning), data mining (Data Mining), information extraction (Information Retrieval) and the like. Computer Graphics (Computer Graphics), computer vision (Computer Vision), multimedia (Multimedia) and large-scale data processing database (Database and Large Scale Data Processing) WWW (World Wide Web) natural language processing and speech (Natural Language Processing and Speech) HCI (Human Computer interaction) software Engineering (software engineering)  To move forward the human civilization is inseparable from thinking, discovery, build. I used to work at Microsoft Research Asia, the Ministry of technological innovation over seven years, and the engineering team where I do a lot of projects together computer scientists in different fields, these projects have their own characteristics:  o Build To Learn: developing software build system is designed to do further tests to try to find some objective law or explore the pros and cons of the method. These projects are often the basis for the work of research papers.  o Build To Show: In order to highlight the role to show a technology demonstration to develop some software for the purpose of these projects is very eye-catching, often get news, but the function may not be comprehensive.  o Build To Serve: To serve a range of target users and build tools, and sometimes in the form of public release of the SDK.  o Build To Win: to win customers in the market and to build for the target software. This is also a variety of scientific discoveries, technological breakthroughs best touchstone.  **Team manage**  A team needs some processes to manage development activities, the work of each engineer in the software life cycle should also be made to have a process, and this chapter will introduce the PSP (Personal Software Process, individual software development process). How can responsible for their own module functions defined as clear as possible, change the internal module does not affect the other modules, and modules can be stable quality, quantitative guarantee? Unit testing is a very effective solution.  Ability Carnegie Mellon University (CMU) Maturity Model (CMM and CMMI), is used to measure the ability of a team set of models. CMU experts for software engineers also have a model, called the Personal Software Process (PSP), PSP, and, like any other methodology, are not achieved overnight.  Modern software industry after decades of development, a complete software by one person alone, has rarely met, the software is done in mutual cooperation. Cooperation is the smallest unit of two people, two engineers together, to do most things is the "look at the code," everyone can see "someone else's code," and express their views. But every one of what is "good" code specifications may not agree, then it is necessary to give us a baseline - what is good coding standards and design specifications. Programmers to write code posters, or to the machine to see? See also, the machine could not see, but ultimately people watching. Our code to make "bystander" to see clearly.  Do a commercial value of the project, or work in a team, code specifications is very important. "Code norms" can be divided into two parts:  1. Code style norms. The main provisions of the text on the seemingly superficial, in fact, very important.  2. Code design specifications. General principles of programming involves the relationship between modules, and other aspects of design patterns.  Code-style principles: concise, readable, and unambiguous. Tip: Here to talk about the style of one of the words, in case of dispute, the key to the spirit of "keeping concise, to make the code easier to read" principle, take a look at the dispute in the code specifications can allow programmers to better understand and maintain program.  It is to use the Tab key to good, or 2, 4, 8 spaces? Conclusion: The four spaces in Visual Studio and some other editing tools can be defined Tab key to expand into several spacebar. Tab key is no reason, the Tab key will display different lengths in different situations, severe interference reading experience. From the four spaces from the readability, the just  Line width must be limited, but some 80 lines of characters before the document specified width is too small (previous computer / typewriter display line width of 80 characters), and now the times are different, can be limited to 100 characters.  If we do the project is real, specific and changing needs, there are contradictions schedule, quality and resources, team members each level, the goal is also inconsistent, and so the team cannot be without internal contradictions. However, the contradiction is not the beginning of the outbreak, it has its own life cycle, and there are different stages of development. Love, two cooperation projects have similar phases, following our life ballroom dancing, for example, describe the various stages of the two cooperation.  1. Infancy (Forming) to dance, for example, when the two first met, cautious and courteous. This phase of the phenomenon: They just know each other, then we are polite and generally a lot of exchange, everyone wants to accept each other, trying to avoid conflict and opinions prone challenges. Dance forthcoming, there are different expectations, but the two sides do not understand each other.  2. The run-in phase (Storming) began to dance, beginning step on the foot. After the contact, just feel helpless, his eyes do not know where to look, to feel each other original dance is so ...... so clumsy.  3. Specification phase (Norming) gradually harmonious dance, step, team members made a lot of things the same. Some written or unwritten rules and gradually built up. A gesture the man gently, the woman will know how to rotate.  4. The creation stage (Performing) into one dance duo, art and dance (she sent a buzz and desire). Not all cooperation to achieve this stage, running too much, we may also enter the "disintegration of the stage."  5. Disintegration stage (Deforming) disband, go their difficult path, back to the dorm holding bench dance, or find another partner.  As can be seen, these teams have a common characteristic: 1. The team has the same collective goal, the team to accomplish this goal together. A team member does not have to work at the same time, such as a relay race. (King Estates moving bricks "non-team" members is not the case, how much each person would like to move much on the move, I did not want to leave the settlement wages.) 2. The team members have their own division of labor, dependence mutual cooperation and common tasks. (King Estates moving bricks "non-team" members of their action is independent of the task is completed, some people walked out, no real impact on other people moving bricks.)  **Development model**  Waterfall model (Waterfall Model) When the period of the software industry is still young, it is from other mature industries (hardware design, construction) borrowed a lot of experience and models. In those "hard" in the industry, the products are mostly follow [Analysis → design → implementation (manufacturing) → Sales → Maintain] this process. Since the "hard" once the mass production of products in the industry, returns again to modify it very difficult, if not impossible. Therefore, this model describes the one-way, irreversible processes.  Rational Unified Process Unified Process (RUP) various models from the beginning of the waterfall model have one thing in common: weight program, heavy pre-designed, re-expression of the document. This type of approach to be considered a master of the Rational Unified Process (Rational Unified Process, RUP) [Note 5]. RUP all phases of the software development integration in a unified framework. To complete a complex software project, various members of the team to do different things at different stages, these different types of work called the procedure (Discipline) or in the RUP workflow (Workflow).  Owner-driven processes (Boss-Driven Process) and the author of some Chinese companies software developers exchange when heard many references to the development process in fact led by the executive leadership or by the company's owner-driven, we shall say it is named for the owner-driven processes (boss-driven process).  Progressive flow delivered (Evolutionary Delivery), MVP and MBP this process is Steve McConnell (Steve McConnell) in 1996, summed up, but it is already very close and now we are talking about more iterative development process. When the main demand and the architecture of the system clearly, software team into an evolving evolution cycle: [development → News → → listen to feedback to make improvements based on feedback  Agile development  Agile development principles  1. The early and continuous delivery of valuable software to meet customer needs  2. Agile processes Welcome changing requirements, and take advantage of this change to improve the user's competitive advantage  3. regularly publish available software, publishing intervals from weeks to months, can be as short as short  4. Business people and developers in the project development process should work together every day  5. enterprising artificial heart project, the full support of their trust  6. Whether the team inside and out, face to face communication is always the most effective way to communicate  7. The available software is a key measure of the project's progress  8. Agile processes should be able to maintain sustainable development. Leadership, team and users should be able to continue at the current pace continue cooperation  9. Only constant attention to technology and design, in order to more agile  10. Keep simple - as simple as possible workload skills - extremely important  11. The only self-managing teams to create outstanding architectures, requirements and design  12. always summarize how to improve team efficiency, and put into action  Lessons agile processes here are some practitioners lessons:  1. Agile Manifesto suggests some priorities, you do not have to be as controversial decree or dogma.  2. Scrum Master is not an officer, but a no executive powers communicators, like as PM Microsoft. He / She would also like to do specific work in a team. Directly to the original "manager" into Scrum Master, most of the work.  3. Some projects require a lot of secret operations and political wrangling to get, Scrum will these contradictions are placed in the daylight. This is good, there are risks.  4. In a complex project, let the front-line team members to make a decision.  5. The start-up companies in fact are often running team in Scrum modes (except everyone is too busy, no time to argue their own in the end how Scrum).  6. In estimation Scrum planning stage is not a "contract", the leaders do not regard it as a contract. Always inaccurate estimate. Insist on short-term Sprint, so that even if there will not be big inaccurate estimate of the damage.  7. Do not talk about management and "process", they only care about "result."  8. In large teams, regional teams, or complex projects, Scrum and no perfect answer, Scrum's founders admit it.  In addition to the software team to write code, test code to do the design and drawing of members, there is a class role do these things above, but also very important, and we call them project managers --PM. PM of M is the Manager, but there are these types of P: Product Manager, Project Manager, Program Man-ager, in different industries and companies, their roles are different. This chapter introduces Microsoft's program manager --Program Manager. Product Manager: Product Manager - to do the right product. At present, most of the company PM refer to this post. Product manager for one or more products or product lines is responsible, and Internet products related to these aspects: product positioning, market development, demand analysis, operations, marketing, marketing, business cooperation. Product Manager across these sectors, find the resources to continue to promote the product. With the development of products, different companies, the PM requirements will be different. Core requirement is based on market and customer needs, coordination of various departments resources, product positioning and correctly grasp the direction, to solve customer pain points, continue to optimize ours.  **Project Manager**  Project Manager: Project Manager - to do the process correctly. In some companies, this position separately with separate product manager. They are in charge of the project process, namely on-line project from the project to be completed on time. Proper coordination of internal and external teams, the deployment of the various departments of resources and time, effective risk management, to ensure the smooth knot according to plan a project, a project manager is a core value of  Program Manager: Microsoft's job title. Microsoft product teams are assigned the role of the three pillars of PM, development, testing. PM is responsible for everything except for product development and testing. In a sense, the first two characters of synthesis. Microsoft usually have a special product planning (Product Planner), full-time staff and their market sector together, responsible for long-term development and marketing of products.  With the business development and growth of the team, the following two questions stand out:  1. The sharp increase in the cost of communication among team members  2. There are many things to develop and test other than the person responsible needs  Some students said, I do not have to write code to test, I do not think in addition to development and testing, as well as what to do. We look at what Microsoft has several types of PM. There do functional design of PM; some features or products require a deep understanding of the various branches of computer science expertise to do a good job. Such as Visual Studio in a variety of computer languages, frameworks, project manager TFS project, SQL Server, Windows Server, Azure, Bing Search algorithm core team of some PM. PM needs of business and the customer has a strong ability to understand, for example, Office software some PM requires extensive experience and knowledge, as well as business development capabilities, such as MSN Internet division of PM and some are driven processes PM, for example, hundreds of people pushing to complete a version of the development team, and if the guarantee Windows Phone release can also have specialized in a particular area of the PM depth in dozens of different hardware, such as internationalization / localization (Globalization / Localization) as well as software and researchers responsible for cooperation, wondering how cutting-edge technology into the mainstream products, do technical transformation PM  Become a qualified PM, which need capacity?  1. Observation, understanding and fast learning ability PM to be able to quickly get started in a new field. PM to be able to understand the user, to the user's perspective to consider the issue, observed that implication needs of the user inarticulate, understand the team members, listening to the boss / client / stakeholder overtones. We must be able to understand other people's situation, mental ability motivation - empathy. Usually a PM might be able to get along a lot of high-tech tools, but when work needs, he / she can suddenly transform itself into a completely non-technical novice user, from the user's perspective.  2. Analysis of daily management of a multitude of things happening in the project, PM to be able to analyze the priorities, find the priority, to make a judgment, decision......  3. Be sure of professional competence If anything, professional competence, then, PM is the professional understanding and expression, you can understand the different human psychology, needs and implication? Can you help of text, graphics, sketches, and even the code to clearly and accurately express their thoughts? PM should always be able to enthusiastically sell the product to the user, we want to sell the team. PM also usually write code, to Fun Excel, PPT, Visio, Gantt charts, will be PS, have writing skills, writing blog someone to love to read, anyway, must have learned the recipe for it! Needless to say, there are also a lot of reading on the IT industry, user psychology, society must have a broad understanding.  4. Introspective ability. You can set a PM racking our brains to do the first project period, each of the offerings promise last shot ass leave (who was young it did not), but after the failure to have the ability to self-reflection and self-improvement. In life can exercise ability PM it? Of course, such as decorating the house, organize a large-scale activities with their children, with three days to help a neighbor child, and so on.  In one project, the specific tasks PM What is it? Their task is to:  Objectives 1. Lead the team formation team / vision of the abstract goals into actionable, concrete, beautiful design;  2. The specific functions of the software life-cycle management (demand / idea / design / implementation / test / edit / publish / upgrade / migration / phase-out);  3. Create and maintain the software specification, let it become a developer / tester timely and accurate guidance, rather than an obstacle;  4. The representative of the interests of customers and users, the initiative to collect user feedback, the new user demand is expected. Coordinate and determine the needs of priority;  5. Analysis and led other members of the defect / change needs a consensus, and ensure implementation;  6. lead other members to ensure the project hold function / time / resources in a reasonable balance, track project progress, to ensure customer satisfaction team released software;  7. The various data collection team project management and software engineering, an objective analysis of the advantages and disadvantages of the project implementation process, project members to promote continuous improvement, boosting morale.  **UI&UE**  In fact, computer software UI (User Interface, UI) and user experience (User experience, UX) is a country with rich content of academic fields, software engineers work in the long term and has accumulated a lot of relevant experience  Norman further clarified the three levels of design and the corresponding product characteristics: instinct (Visceral) level design - shape behavior (Behavior) level design - fun and efficient use of reflection (Reflective) level design - self-image, personal satisfaction, recall three levels of factors are intertwined, and the combined effect of the user experience. Most software engineer is primarily concerned with "use efficiency", this is just a small part of the user experience design.  1. To provide tactile feedback as soon as possible to have the system status feedback, the waiting time to the right. The program is now what is happening, it should be clearly marked in one unified place. A target user is able to rely on the main feedback software to perform basic operations without first learning to use the manual. Feedback system can be visual, auditory, tactile (for example, the phone vibrates). However, to avoid repetition of simple tips.  2. System interface match the user's real practice (Familiarity, Avoid Surprise) to communicate with users, the software system to use the language of the user rather than the developer language, the concept used to be close to real life, rather than the concept of academic term or developers. We are talking about real life, real life experience is the best target user. For example, the network used by the patient registration system, it should not be used only to medical workers are familiar with the terminology and interface (the worst results were obtained using software engineer was familiar with the terminology and interfaces, and the medical staff and patients of this very familiar) to avoid feedback software gives the user a surprise - for example, the user does not expect the dialog box, the software at odd angles from the pop-up dialog box or prompt the user to "find the object."  3. The user has control over operating errors can be rolled back, let the user can exit the software (some software does not exit the menu, which is leading a major cause of resentment user). Users can customize the amount of information displayed, you can customize common settings.  4. The consistency and standardization. In the software, for a term representing the same thing and the same operation, should be consistent throughout. For example, a dictionary software function of "to help users gather and recite the words". This function must be clear and consistent called, instead of be mingled with "the word of this", "Glossary", "Word List", "Word Book", "word document" ...... and so on.  5. Suitable for all types of users. Our software provide customized design for novices and experts. Some mode of operation, such as fast operation, the user can adjust. We should also provide a degree of convenience for certain impaired users (color weakness, color blindness, the blind, hearing impaired users, inconvenient keyboard mouse user, etc.). For long-term use of a software user, the software should be able to adapt to the user's habits, allowing users to More with the more smoothly, and finally generate goodwill and loyalty emotion.  6. Help users identify, diagnose and repair errors. In the software key operations there should be confirmation prompt, in order to help the users to eliminate misuse as soon as possible. Note that the use of plain language to express the error message. Need to give prompt for the next operation after Error Message (I'm wrong, then what to do for the next step). Provides detailed information to help when necessary, and help users easily restore their work from their mistakes. All users can submit feedback via e-mail or form. Some programs use a simple pair of smiley / sad face symbol to encourage users to submit feedback, which is a good solution.  7. There is necessary tips and help files. If the users can freely use the software without documents, of course better, if necessary, provide online help. If the software relates with the user's work (rather than a simple game), then the basic tips and help documentation is necessary, but also offers a convenient search function. Describe the specific steps from the user's point of view and not too lengthy. Some software will show when you first start the usage of some new functions by way of illustration or animation, or guide the user through some basic settings (for example, when you first use the input method, allowing users to select the number of candidate words, font size, and many more). These are a good way. In the PC desktop software era, software team always have to wait until the stabilization phase of the project began to write "Help documentation" because the previous software interface and features may change a lot, and then you need to write quickly, to publish with software. In the Internet era, offline help document progress to "Online Help" page; in the help of large bandwidth and active user community, we can see the videos of how to use of various software efficiently generated by users. This should give the team a lot of inspiration - how to use all sorts of forms of "help file".  **The End**  We are familiar with computer and IT field, all the "cool" things that we see are innovation results by several generations of people, many teams’ continuous innovation. Like a jigsaw puzzle, a lot of smart people vaguely see the final image splicing piece by piece, but usually the person who find the last piece get the greatest honor. However if there is no accumulation of their predecessors, did not own a solid skill, there is no "last piece" waiting for you to splice. Another corollary is - do not start with thinking to find and fight all the puzzle pieces that can create a huge innovation. Peter Drucker (Peter Drucker) said: Those entrepreneurs who start out with the idea that they'll make it big - and in a hurry - can be guaranteed failure.  Of course, at the moment must also recognize the danger. Excellent team sometimes complacent, team members feel good, too close may also lead to excessive self-interest, do not attach importance to cooperation with other teams, do not pay attention to customer needs and so on. Recent IT industry has repeatedly verified the story of the "great company are only eighteen months away from bankruptcy" the law was not to be taken lightly by outstanding team leader.  **学生签名：**    **导师签名：** |