

Yi Li's Report

by Yi Li

FILE	17851_YI_LI_YI_LIS_REPORT_227501_1179445591.DOCX (24.03K)		
TIME SUBMITTED	21-JAN-2020 04:12PM (UTC+0000)	WORD COUNT	3676
SUBMISSION ID	118853429	CHARACTER COUNT	18618

Agile Software Engineering Principles

Yi Li's Report

Abstract :

In the software industry, the delivery process of software products is often delayed, its expenses often exceed budgets, often missing software features required by customers, and nearly 20% of projects are ultimately undeliverable or cancelled. These software developments often cost a lot of money, manpower, and time, but a large part of the product features delivered to users are not used by users, or they are not able to help users solve problems. Based on these existing problems, this article elaborates the definitions of Behaviour-Driven Development (BDD), Feature-Driven Development (FDD) and Test-Driven Development (TDD), their respective advantages and disadvantages, and their respective best application environments. Hope to apply these ideas to the actual software development process.

Behaviour-Driven Development (BDD)

Definition

BDD is a user behavior-oriented development method. When product managers or business personnel put forward product requirements, the development team needs to communicate effectively with them, clarify the functions of software development and customer needs, and throughout the development process. Timely feedback enables the other party to follow up the product development status and progress in time, so that the other party can intervene and adjust the entire product research and development according to the needs, so that the development team continuously clarifies the characteristics of the product during the development process, thereby further improving the development efficiency and Degree of completion.

Feature

Behaviour-Driven Development is very different from traditional software development methods. The traditional development method uses waterfall development. From software concept to requirements analysis, the analyst then designs the software architecture and then passes it to the coders for work. In this process, as long as there is a little carelessness, the user's needs will be omitted or tampered with by some link in the middle. In this way, the completion of the project will be reduced, and it will be difficult for the development team to deliver on time to meet customer needs Software product.

Practical application

The most suitable development scenario for Behaviour-Driven Development technology is the collaborative work of project analysts, software developers, and software testers. During the development process, the developer repeatedly collaborates with the R & D personnel to confirm. After the test unit code that can be tested is completed, it is converted into an automated detection script, and then the developer

simultaneously tests whether the code he wrote complies with the characteristics of the product. At the same time, the feature file of the software product is generated, so that the R & D document and the R & D code are synchronized, so that the developer, tester, and customer can understand the project progress at the same time, thereby indirectly accelerating the product development progress and reducing In order to reduce the pressure caused by software product planning or missing features.

Feature-Driven Development (FDD)

Definition

Feature-Driven Development (FDD) is another development method in the software development process. Its core content is Feature, that is, the ultimate goal of this development method is to achieve a certain function. In order to achieve a specific function, in this development mode, a function is usually broken down into several sub-function sets, and then the sub-functions are divided into functional functions with a more single function, which is divided step by step to divide into specific Functions.

Feature

The development process of Feature-Driven Development (FDD) can be divided into five steps, each step is around how to achieve a specific function. Under this development method, the staffing of the team and BDD are also different. Among them, the main developer is responsible for the supervision and management of the macro project, and the personnel belonging to the class are only responsible for a certain class, that is, a certain class is only composed of It is done by a developer. This is a very effective development method, which avoids inconsistencies in the code on the one hand and improves the developer's sense of honor on the other. The third is a functional team. A functional team is usually composed of relevant people who can implement the function, so it is temporary and reusable. Temporary means that once the function is completed, the team is dissolved and reused. ~~Sexuality means that the same developer~~ can belong to different functional teams. In this way, the division of labor of the entire team is quite clear. In addition, when using the FDD development method, it can perform function updates usually once every two weeks. Its extremely fast update speed has a strong ability to respond to the market, can respond to market demand in a timely manner, and is extremely strong for the contemporary high-speed development society. Popularity.

Practical application

The development process of Feature-Driven Development (FDD) can be divided into five stages. The first step is to develop the overall model, which is mainly to determine the macro model of the entire set of software, confirm the functional requirements of the product, and the relationships between the various classes, classes, and classes. The second step is to refine the functions and convert the function model that has been determined in the previous step into a specific function set. In the design process of the second step, some explanatory documents should also be completed, including the function set description and design outline text . The third step is to formulate a work plan according to different functions, sort the functions according to

their priorities, set a completion date, and assign each function to each person. It should be emphasized that the iteration cycle of the code is two weeks. The implementation cycle of the function exceeds two weeks, it should be further detailed. At the same time, it should be noted that the R & D tasks of this cycle are core functions. The fourth step is to further improve the function, and at the same time formulate the task plan for the next cycle. The fifth step is to test and update the version according to the function that has been implemented until the function is implemented.

Test-Driven Development (TDD)

Definition

Test-Driven Development (TDD) is another development technology that is agile, and it is also a methodology. The simple concept of TDD is to write and correct failed tests before writing new code (before development). This helps avoid code duplication, because we only write a small amount of code at a time to pass the test. (Tests are just the conditions we need to test to meet their requirements). Test-Driven Development is the process of developing and running automated tests before the application is actually developed.

Feature

Generally speaking, programmers are willing to reflect the function perfectly in the code, but sometimes it is unsatisfactory, and it is inevitable to worry about it. Can my code meet the business needs? But in TDD, the test case is written first, and then the class or user interface is developed. Because test cases need to be developed first, the developer must be clear about the purpose of the test, so that TDD ensures that the code of the project matches the required business. The main goal of TDD is to make the code pass the test without bugs.

Unlike other development methods, TDD starts with building and testing. First, developers need to know what kind of tests to develop, what data to test, and how to pass code tests. In the normal testing process, the code needs to be generated first, and then tested. In TDD, this approach may fail. Because the test was developed before the code was developed, in order to pass the test, the development team must refactor and modify the code. Refactoring the code means changing the implementation of some code, but does not change its final function.

There are many advantages to using TDD. The first point is that code hidden defects can be found in time. Through continuous testing, the recorded data can build automated tests. In this case, developers can collaborate with any other development People re-run the test; the second point is that this development method can help developers design clean, leaner, and more conducive to expanding code. At the same time, in the process of writing testable code, due to the fragmentation of test units, the functions to be tested tend to be single, making it easier for developers to understand the code's architecture and specific information.

The third point is that when using TDD to develop, there is a temporary absence of team members in a timely manner, and it will not affect the progress of the entire project, because other members can also easily obtain and process this part of the code, because

TDD is unique ,the advantages make code sharing between teams simple and help deepen the entire team's understanding of the project.

Practical application

What TDD emphasizes is to pass the test, everything else is no longer important. Any "redundant" function is a function that is not covered by the Unit Test Case, so there is no value in implementation and processing. Writing such processing code is considered inexplicable because they will not be tested and there is no such Unit Test Case. As a real programmer, you will not tolerate low-level code that is difficult to understand, and then rebuild Perfect code based on what test cases. Because the first thought is usually serious, the code written for the first time is the most rigorous, and it cannot be modified later. Instead, the subsequent code is more for the green light, and the efficiency and beauty are gone. It's a bit like slandering someone who really refactors the code seriously? Actually not, if a person is working seriously, why not think carefully at the beginning, and use the time of writing the so-called Test Case to write your first version of code, and then execute it carefully? Because such a cost is actually inevitable when debugging, if you really want to completely (note that it is completely) rely on passing the test, this is also a serious development efficiency problem.

Can version upgrades and extensions really rely on TDD? First we assume that upgrades and continuation changes are limited, otherwise rewriting will not be included in this discussion. Tests rely on test cases. Tests, such as code comments and development documents, also have the problems of being outdated and out of sync with the implementation, which all require maintenance. And the unit test case written in ugly is the same as the misuse of comments and documents. In the future, it was useless and no one was willing to maintain and modify this muddy water. The real guarantee of version upgrade and continuation is the stability of Product Team personnel. If the main staff has been working steadily from design to implementation and even testing, it will be difficult to make new designs and changes unstable after implementation. Is this TDD capable? Some memorable notes, a bunch of inexplicable development documents, and a tedious Unit Test Case, I don't know how the newcomer starts his continuation work after facing it.

Programming and Software Testing

Yi Li's Report

High cohesion

1. Definition

In the process of Java programming, High cohesion is a requirement for a certain functional module. That is, whether a functional module has a high cohesion

is a criterion for judging whether a function is good or bad. The functions mentioned above refer to a certain object, method, class, module in the process of programming. The higher the degree of cohesion of a work, the less things it needs to perform, so it has a higher degree of concentration and a higher execution efficiency. High cohesion has the following characteristics.

2. Feature

(1)legibility

In the process of Java development, the most basic thing is to write code. Whether it is for development work or later debugging, a piece of easy-to-understand code can make it easier and faster to understand the meaning of this code, which is conducive to the development work. For example, when we call functions between different classes, if we cannot clearly understand the meaning of each function name, it will cause great obstacles to the next development work. Therefore, when we declare a function, we must use a simple, easy to understand, convenient and direct way to name the function. In this way, for each class, we can clearly understand its meaning, as well as the properties and functions and objects contained in the class, it will be more convenient to call, reducing the possibility of producing errors.

(2) Reusability

When developing Java projects, the simplest way to reuse is to copy the code. However, this way makes the code look verbose and complicated. Therefore, a smart developer will encapsulate the functions that are commonly used or need to be called as a whole into an object. When you need to use this function, you can call this object directly. For example, common functions for getting strings and printing strings. Of course, code reuse is also a manifestation of high cohesion, because only functions that meet this feature can be easily called in various projects.

(3)Renewability

When a Java project is developed and put into the market, the project will continue to be updated with the market demand. The workload of version update is directly related to the level of cohesion. Generally speaking, code with high cohesion nature has advantages when performing function updates and modifications, because functions tend to be singular, while changing only one function, the impact on other functions is low, so the code can be greatly improved efficiency.

3. Practical application

In an actual application environment, high cohesion does not mean that a certain piece of code implements only one function, but is determined according to actual engineering needs. In general, if a code to achieve a certain function exceeds 100 lines, then this function can be broken down into several sub-modules as needed. In this way, whether it is during debugging or when the function is updated, it is even more Advantage.

Loose Coupling

1.Definition

Compared to high cohesion, the description of loose coupling is more focused on the measurement between different functions. Coupling refers to the association between different elements. Elements can include methods, objects, modules and functions. Loose coupling points out the principles in which the program is designed.

2.Feature

Advantage

When the program design follows the principle of loose coupling, the degree of correlation between different functions is very low. In this case, whether it is a function call or data update, it will not involve too many functions. In other words, the operation of element A does not depend too much on the existence of element B. Among them, we should understand the word "excessive". It does not mean that there cannot be a dependency relationship between different elements. From a practical point of view, this is impossible. The loose coupling refers to the low degree of dependency between different elements, that is, in programming In the process, we only connect the necessary dependencies, and those unnecessary dependencies should be avoided as much as possible.

Disadvantage

The disadvantage of Loose coupling is that it increases the memory consumption. For some large projects, it is thought that reducing the coupling degree will cause low operating efficiency.

3.Practical application

In summary, Loose coupling is a concept that needs to be weighed repeatedly in practical applications. While loose coupling brings convenience in use, it also increases the difficulty of design. It is in contradiction with high cohesion, and in the development work, avoiding excessive design can maximize the importance of these two principles.

Information Hiding & Data Hiding

1.Definition

Information hiding technology refers to the method of embedding certain information into the host information stream in a specific way to achieve the hiding of information. The purpose of information hiding is not to hide information, but to protect it. With this technology, information Reduces the risk of exposure, thereby reducing the possibility of information being attacked.

2.Feature

(1)Security

When we are in the process of programming, we will inevitably involve the function call. When function A calls function B, A may need to access only one or several elements in B. In this case, If there is no information hiding technology, then all elements in function B can be read by function A, which greatly increases the risk of information exposure. Conversely, if function B only grants access to specific elements of function A, then for function A, all elements that it needs to access can be successfully acquired, while functions that it does not need to access are not authorized to acquire. The result is greatly improved data security. For example, in my program, I set local variables in some classes to private properties, such as the size of the board, and set objects that act on the entire game to be public, such as the attributes of the chess piece and the player. In addition, even if the program happens unexpectedly, the scope affected by private properties is limited to a small range, which greatly improves the efficiency of debugging code.

(2)Simplicity

Speaking of simplicity, it involves the simplicity brought by the information hiding in the program design process and the simplicity brought by the program maintenance process.

In the process of programming, we can reduce the complexity of the code through information hiding technology. That is to say, in the process of calling a function, we only need to focus on the objects and variables we need, and we don't need to care about the elements that are not needed temporarily. This reduces the possibility of code errors. In addition, for example, in my program, I need to define the size of a chessboard. I tentatively set a square board with a value of $15 * 15$. However, in the process of writing the code, I found that this value is too large. Modify it, there will be a lot of code needs to be modified, so I adopted the idea of information hiding, and defined the size of the board as a variable "boardSize". In the process of writing the code, whenever I need to call the size of the board or When a conditional comparison of this value needs to be designed, I can directly call this variable name, eliminating the need to modify a large amount of data each time, and improving development efficiency.

During program maintenance, the benefits of information hiding are even more obvious. For example, in the process of debugging my code, there is a variable in my code called "chessBoard", which belongs to the class "PlayChess". At the same time, I also have a class name also called "ChessBoard", so, "chessBoard" It is a local variable, and its scope of influence belongs to a certain class. The class "ChessBoard" I defined refers to the board interface of the entire game. It is a global variable. In the process of calling and debugging, it can only be set in advance. The influence range of an object can make the program run reasonably and efficiently.

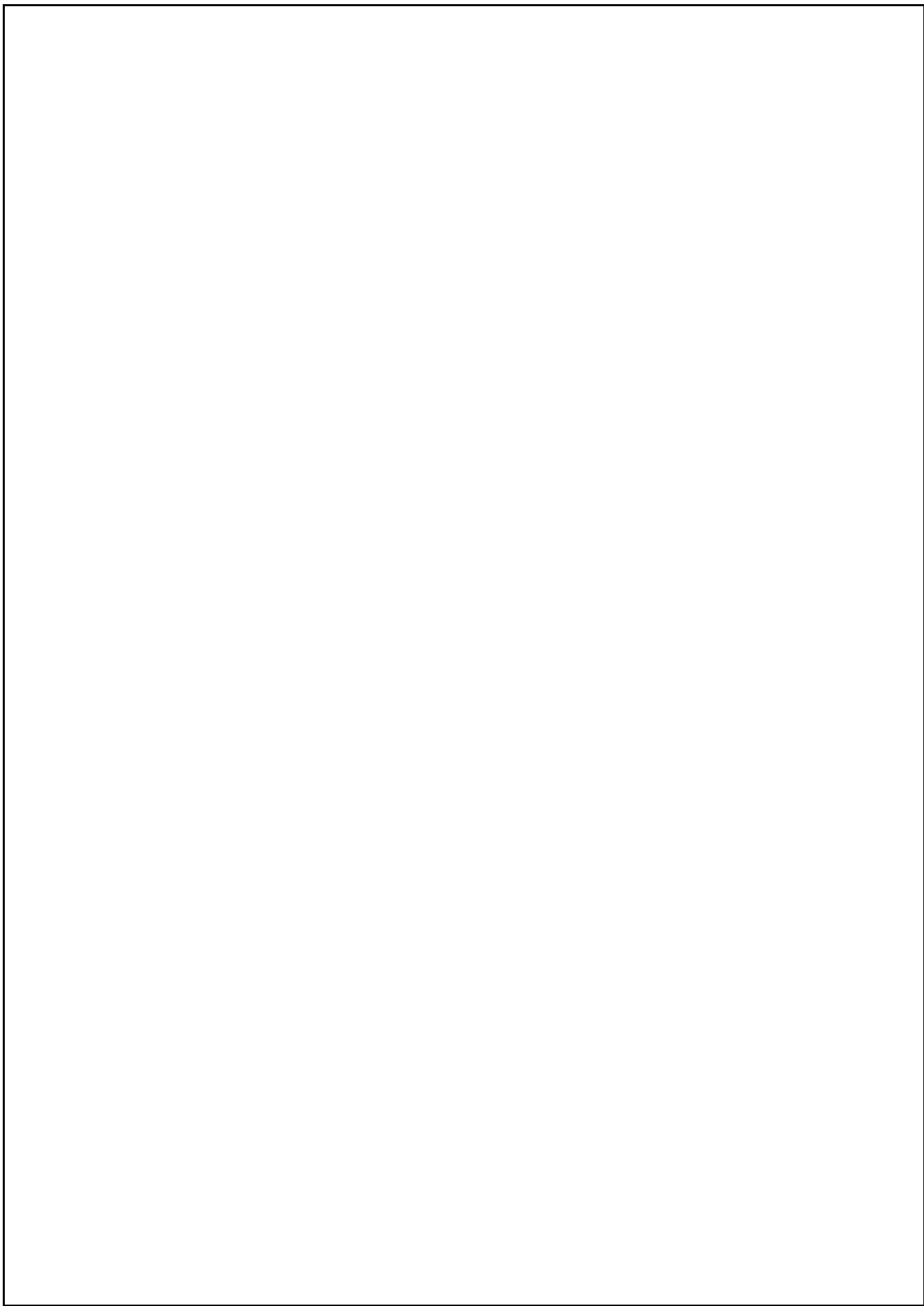
Conclusion

In summary, high cohesion and low coupling are the basic principles of program design. We should always keep in mind when programming, and make repeated trade-offs to increase the focus of the function as much as possible, so that the function of

each module tends to Simplification, at the same time, reduce the dependencies between modules as much as possible, making function calls more convenient. Information hiding and data hiding are the prerequisites for good code. Global variables can be continuously changed throughout the project, while local variables can continue to play a role in the subclass scope. The two complement each other to produce efficient and safe code.

Reference

- [1]张扬,徐亮. TDD/FDD 语数分层研究及总结[J]. 科技经济导刊,2019,27(35):168.
- [2]李文秀,王景全. TDD&FDD 融合优化组网方案研究[J]. 信息与电脑(理论版),2019,31(21):179-184.
- [3]陈秋菊. 基于 BDD 算法的工程 BOM 准确性研究[J]. 内燃机与配件,2019(16):211-213.
- [4]夏宝平. TDD/FDD 融合组网负载均衡策略综述[J]. 电脑知识与技术,2019,15(22):61-62.
- [5]Jiang Yangyang. Research on Application Value of Computer Software Development in Java Programming Language[A]. Institute of Management Science and Industrial Engineering Proceedings of 2019 7th International Conference on Machinery, Materials and Computing Technology(ICMMCT 2019)[C].Institute of Management Science and Industrial Engineering: 2019:4.
- [6]陈子文,赵峰,陈小强,王英. 基于 BDD 考虑共因失效的接触网系统可靠性分析[J]. 中国安全生产科学技术,2019,15(02):170-174.
- [7]肖苏,王洪春. 改进 BDD 算法在故障诊断中的应用[J]. 重庆师范大学学报(自然科学版), 2019,36(02):81-85.
- [8]Tong Qiang. The Design of Game Platform Based on Java[A]. 上海工程技术大学、香港机械工程师协会.Proceedings of 2018 2nd International Conference on Artificial Intelligence Applications and Technologies (AIAAT2018)[C].上海工程技术大学、香港机械工程师协会:,2018:6.
- [9]Xin-hua YOU. Brief Discuss the Application of Object-oriented in Java Language Programming Course[A]. Advanced Science and Industry Research Center.Proceedings of 2018 3rd International Conference on Automation, Mechanical and Electrical Engineering (AMEE 2018)[C].Advanced Science and Industry Research Center:,2018:5.
- [10]Zhiqiong Liang. The Application of JAVA Programming Language in the Development of Computer Software in the New Era[A]. Information Engineering Research Institute,USA、Asia Pacific Human-Computer Interaction Research Center,Hong Kong.Proceedings of 2018 4th International Conference on Mechatronic Systems and Materials Application(ICMSMA 2018)[C].Information Engineering Research Institute,USA、Asia Pacific Human-Computer Interaction Research Center,Hong Kong:,2018:5.



Yi Li's Report

GRADEMARK REPORT

FINAL GRADE

50/100

GENERAL COMMENTS

Instructor

Task 1 report (35%). 15

Task 1 use of git (15%) 10

Task 2 report (20%) 13

Task 2 code implementation (30%)

12

Comments

=====

Basic concepts appear to be covered. The English needs to be addresses, much of this report is quite difficult to read.

Discussions of cohesion, etc., are easier to read.

Code contains a number of errors including an infinite loop. Needs far more testing.

Git usage lacks branches but a number of commits.

PAGE 1

PAGE 2



Comment 1

????

PAGE 3

PAGE 4

PAGE 5

PAGE 6

PAGE 7

PAGE 8

PAGE 9
