Rhea Mae Edwards

Michael Hilton

Software Engineering I (CS 361)

3 March 2016

Adoption of Test Driven Development

Software development models are an initial guideline for programmers to follow in order to move through the process of creating a computational program. Test driven development is a type of software development model that should be implemented in the industry, and to be more specific, within our company. Our company should encourage the adoption of the model of test driven development. Such a model lays out a direction that many programmers can greatly benefit from. There is a type of discipline presented that will have the long run with creating computational programs smoother and more reliant. Even though styles of other software development models may be in current use in our company now, test driven development can enhance the experience of our current habits and improve issues when adopted within our system. Test driven development has many valuable aspects to offer.

Generally, test driven development is a software development process that is built from the basis of three simple rules according to the article “The Three Laws of TDD” written by Robert Cecil Martin. The first rule being that one is “not allowed to write any production code unless it is to make a failing unit test pass.” Rule number two is that one is “not allowed to write any more of a unit test than is sufficient to fail, and [that] any compilation failures are failures.” Lastly, one is “not allowed to write any more production code than is sufficient to pass the one failing unit test” is the third rule that should also be generally followed.

The test driven development cycle begins with writing a test that will fail, followed by writing code to pass a failed test, then if needed and/or wanted to, the refactoring of the code that the program contains of so far, can be done. When following such a cycle, testing small fragments of code at a time during the production of a program, can guarantee the best amount of test coverage. While working with the production code though, it is ideal to try making the smallest changes in the code as possible. Encouraging this software development process in the workplace will add a beneficial type of discipline for our fellow employees overall, with every unit test written.

"Because of the testing modules that are built into [test driven development] continuous integration development model" there is "all without fear of 'breaking' the application" (Benefits of Test-Driven Development). Not only will encouraging the use of test driven development will readability and understanding of how a body of code was written of a program is clearer to a different pair of eyes, reassurance that such programs will become more reliable is increased through its continuous executable documentation.

The application of the test driven development process can also be positively applied to the main types of software that we develop- e-commerce software to sell our products, mobile applications to provide activity tracking for our product users, and with developing promotion and special even website for our company overall. According to the article "Benefits of Test-Driven Development" by Base36 Smart Solutions, "test driven development gives organizations the ability to painlessly update their software to address new business requirements or other unforeseen variables," which could hold true for our company through the adoption of the software development process of test driven development throughout the company.

With the selling of our products with our e-commerce software, it is key to just first simply test that the basics of its functions of transactions work properly. The programs that sell our products that we create, will have properties and aspects to it that can always simply be tested. Having a great amount of test coverage with numerous unit tests will provide assurance of those functions and the programs as a whole.

Furthermore, with writing mobile applications that have the ability to track physical activity for our consumers, can also benefit with the adoption of such a software development process. Building these mobile applications rely on a constant routine type functionality where unit tests along with refactoring will make the amount and organizational display of its code more visually appealing and understandable to a viewing individual.

Similarly, the back-end instructions of our promotional and special event websites can also take into consideration the adoption of this software development process of test driven development. If the websites takes in information from certain databases, testing calls that process that data in order use and display that information, can be done along with unit tests, even though there are also many other aspects in creating an entire website overall.

Other types of software development processes do exists as guidelines for computational programming. One process is a test last type process where all the code of a program is written and then is tested at the end. This process may seem less tedious as a programmer when writing a program, but following this development process is also more risky in the long run. As a programmer, one has trust that all the untested code that is being written for the entire program, will not cause potentially catastrophic issues in the end. Overall, there is a missing sense of reliability in the faith of a program through this process of saving the testing all of an entire program’s code for last, after being written.

Representing the opposite of the previous mentioned process, is developing code for a program where all the tests are written upfront, and then all of the production code is created afterwards. With this process, the production code of a program is given initial requirements in order to be created, which can lead to unwanted issues to occur after writing a program. A programmer may have realized a more beneficial design that will not pass a pervious created test when written code for the program. There is also the aspect that previously written tests will not work well together due to an initial design. Most of the time, the best of creations of a program occur with continuous improvements throughout its development phase, not always based off a design produced in the beginning.

Also, another type of software development process that is known to be commonly used among the industry, is when code and logic is continuously being added throughout the production of a program without any testing involved and/or an initial purpose to begin with (Soft). Thoughtlessly adding random pieces of code and logic into a program, can lead to errors and problems when running a program. With the abstinences of reasoning and testing when writing code, will lead to confusion and inconsistencies to the overall structure of a program's code. In other sense, greatly decreasing the dependability of a created program.

With adopting the practice of the software development model of test driven development, created programs will greatly become more reliant with the use of quickly running automated tests. There is also no overarching design overall initially which can be caused with the use of other software development models such as one that calls for all tests being written upfront when developing a computational program. There is also that crucial aspect that comes with the little bit of extra effort required to maintain unit tests to ensure the valuable quality of

worthy test covered of a program. Working with a tad more effort will provide a high quality product that will be greatly appreciated as a whole.

Test driven development is an act of design rather than just validating a program. It is an act of documentation rather than obtaining a note of verification. It is not all about testing a program, it is a way to write code in an efficient and meaningful way.

All in all, encouraging the use of test driven development when developing programs for our types of software, should be adapted throughout our entire company. Even though others may argue there are other types of software development processes that are seemingly better to use, using the test driven development model will have many able to overcome certain downsides that other processes are likely to generate when programming. The test driven development process will provide a sense of reassurance with the final product of our software products along with a highly organized back-end viewing. From the e-commerce software to sell our products, to the mobile applications to provide activity tracking for our product users, and even to developing promotional and special event websites, using test driven development as we all take a part in developing, will be a beneficial practice to our creations.

Bibliography

"Benefits of Test-Driven Development." *Base36 Smart Solutions*. Base36, Inc., Web. 19 Jan. 2016. <http://www.base36.com/2012/07/benefits-of-test-driven-development/>.

Soft, Riant. "List of Software Development Model and Methods." *In SlideShare*. LinkedIn Corporation, 27 Aug. 2013. Web. 19 Jan. 2016. <http://www.slideshare.net/RiantSoft123/different-types-of-software-development-model>.

"The Three Laws of TDD." *But UncleBob .com*. Web. 19 Jan. 2016. <http://butunclebob.com/ArticleS.UncleBob.TheThreeRulesOfTdd>.

"What Are the Software Development Models?" *ISTQB Exam Certification*. Mindstream Theme on Genesis Framework, WordPress, 2016. Web. 19 Jan. 2016. <http://istqbexamcertification.com/what-are-the-software-development-models/>.