Software Engineering Agile Method Proposal

October 26, 2010

Max Stillwell

TABLE OF CONTENTS

1 INTRODUCTION……pg 3

2 HISTORY……..pg 3

3 CONCEPT……pg 3

3.1 ACTIVITES……pg 3

3.2 VALUES……pg 4

4 CONCLUSION……pg 5

5 WORKS CITED……..pg 6

1 INTRODUCTION

Extreme Programming is a software development methodology developed by Kent Beck that helps deal with changing customer requirements and helps improve software quality. It advocates frequent releases of the program in short development cycles. I am proposing that Group 3 use Extreme Programming as its agile software process this year is Software Engineering.

2 HISTORY

The development of Extreme Programming began with Kent Beck when he became project lead of the Chrysler Comprehensive Compensation System project and began to refine the existing development method that was used in the project. Along with his frequent collaborator, Ward Cunningham, he fixed many of the problems the development team was having with their development process. The methodology Kent came up with mainly was just taking all the best practices of software development at the time and took them to “extreme” levels. For example, test-first development had been used by NASA for Project Mercury in the 1960s, nearly forty years before it began a practice of Extreme Programming.

3 CONCEPT

Extreme Programming is comprised of four core activities that are preformed in repetition during the development process. They are Coding, Testing, Listening, and Designing. Along with these activities Extreme Programming has five values that it considers to be important to the development process. They are Communication, Simplicity, Feedback, Courage, and Respect.

3.1 Activities

The first of these activities is Coding of which advocates of Extreme Programming argue is the only truly important product of the software development process because without code there is no working product. Extreme Programming advocates that when faced with several alternatives to a programming problem you should just program all of the solutions and use automated tests to determine which solution is the best. Extreme Programming also advocates the use of code to help explain a complicated problem to other programmers.

The second activity of Extreme Programming is Testing. Without testing a function you cannot be sure that it even works properly or not which can and most likely will lead to bugs and design errors in large pieces of software. The thought in Extreme Programming is that if some testing can remove some of the software’s flaws then a lot of testing can remove a lot of the flaws. There are two kinds of testing preformed in this activity, Unit tests and Acceptance tests. Unit tests are to determine whether or not a feature works as intended. Similar to Test Driven Development which is an offshoot of this idea programmers write as many automated tests as the can. The goal of these tests is to find ways to break the program so if all the tests pass then that feature is complete. Every feature is tested this way before

moving on to the next one. Acceptance tests verify that the program works as the customer's requirements say it should. These occur during the release planning phase of the program.

The third activity is Listening where programmers listen to what the customer wants the program to do. The programmers must understand what the customer wants so they can give the customer feedback on how the program may work or why it cannot be done.

The final activity in Extreme Programming is Designing. Extreme Programming says that while one can come a long way without a design eventually they will get stuck because the program will become too complex for the programmer to handle. By creating a design to organize the program one can avoid a lot of dependencies in a program which means changing one section of the program will not break all the other sections.

3.2 Values

The first of the five values is Communication. In Extreme Programming the goal is for all of the programmers to have the same view of the system as the users will. Because of this Extreme Programming values simple designs, common metaphors, collaboration of the users and programmers, frequent verbal communication, and feedback.

The second of the values is Simplicity. Extreme Programming advocates starting with the simplest solution because extra functionality can be added later. The goal is to follow the "You ain't gonna need it" approach where you only work on the features you need today instead of the ones you may need tomorrow. Simplicity in design and coding can also lead to better communication amongst the programmers on the team.

The third value of Extreme Programming is Feedback which includes feedback from the program through things like unit tests, feedback from the customer through things like acceptance tests, and feedback from the team. Feedback is important as it lets the developers know what works and what doesn't.

The fourth of the values is Courage. Since in Extreme Programming you are coding for today and not tomorrow programmers having courage allows them to feel comfortable rewriting their code when it needs to be. Courage is also need when throwing away code, the programmer needs to have the courage to throw away something that is obsolete or doesn't work no matter how much work was put into that code. Lastly, programmers need courage to be persistent as they may spend days on that same problem to finally solve it on the final day only if they are persistent.

The final value of Extreme Programming is Respect for others and for oneself. Programmers should never commit changes that will break the program or cause more work for their fellow programmers. Respect is very dependent on the four earlier values and it will lead to a higher level of motivation and loyalty towards the team and project.

4 CONCLUSION

One of the reasons I suggest Group 3 use Extreme Programming this year is that the values and activities seem very important when working in a large group and a large piece of software. Things like Communication and Feedback are very important when developing a large piece of software, especially GUS since it is very customer based. Simplicity is also a very important value for GUS as the simpler and easier it is to use, the more groups on campus will be willing to switch to and use it. One of the main points of Extreme Programming is having frequent releases to the customer to use and test which can be easily done with GUS as it is mostly a collection of features for managing groups. We can program one feature at a time like in Feature Driven Development, make sure that feature passes the testing activity, and then release that to the customers to test and give feedback on what works and what doesn't so we can then go back and make the program better and in the end the better and more user friendly we make GUS the more groups will be likely to use it.

Works Cited

"Extreme Programming." *Wikipedia*. 28 Sept. 2010. Web. 24 Oct. 2010. <http://en.wikipedia.org/wiki/Extreme_programming#Practices>.

"Extreme Programming." *Extreme Programming: A Gentle Introduction.* 28 Sept. 2009. Web. 24 Oct. 2010. <http://www.extremeprogramming.org/>.