

Kent Beck and his contributions to Software Engineering

Cian Boland

November 2020



Figure 1: Kent Beck

1 Introduction

As I have worked through this module and worked through the assignments, I have begun to understand the importance of testing all possible lines and outcomes of your code before deploying it. It gives developers insight into issues, clients confidence in the product and allows a higher and more streamlined workflow. An example of a failure to test leading to serious consequences would

be the Leaving Cert results program, which had a small but serious error that would have been caught by extensive testing from a Test-Driven Development process.

Kent Beck is an American software engineer, born in 1961. Beck studied at the University of Oregon and received a M.S in computer and information science. He began working for Chrysler on the Comprehensive Compensation System. During this development, his team adopted a new process of working which became known as "Extreme Programming".

2 Extreme Programming

Extreme Programming is a type of Agile development. Its main parts are frequent releases in short cycles to allow for changing customer requirements. This involves pair programming, unit testing, not programming features until need and simple code. This is the basis for a significant amount of programming style and types today. The main Values for Extreme Programming were

- Communication - Importance of face to face discussion and regular meetings
- Simplicity - Addressing current goals and not to try and predict future goals avoiding waste
- Feedback - Feedback from clients and managers allows devs to identify areas for improvement and revise implementations
- Courage - Kent Beck described this as "Effective action in the face of fear", courage to accept feedback and act on it
- Respect - Respect team members so that communication and feedback are effective and you can work together to identify solutions.

Extreme programming was pioneered by Kent Beck and he remains its main proponent. His idea of taking many best practices of software development and taking them to the 'extreme' such as testing every line of code and not allowing code to reach production until it had passed every test was used in projects such as NASA's Project Mercury. This is a change from the best practice that existed before of writing formal test documents such as acceptance testing that would test large sections of code rather than each small section.

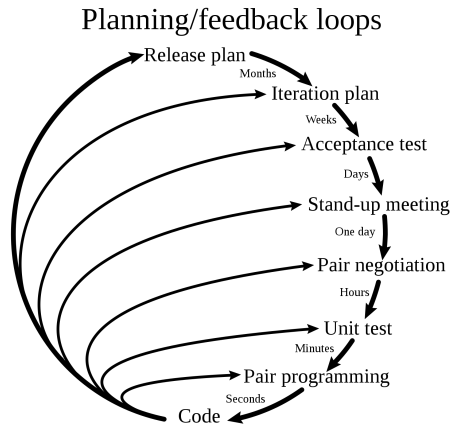


Figure 2: Extreme Programming cycle

3 Unit Testing and Test-Driven Development

The cornerstone of extreme programming, unit testing has become increasingly important in software engineering. Unit Testing is the name for a process that involves testing individual components of an application. Kent Beck created a testing framework for Smalltalk called SUnit. This created a series of frameworks called the xUnit frameworks. JUnit testing for Java or PYUnit for Python. A test-driven development process relies on all requirements to be converted into test cases using something similar to xUnit frameworks and for all development to be tested against all cases during the process rather than on completion. The development cycle of TDD is as follows

- Add test
- Run all tests and see if the test fails
- Write code
- Run tests with new code
- Refactor code to succeed on tests

4 Conclusion

In conclusion Kent Beck's contributions to the field of software engineering are significant and far reaching. The development of XUnit testing allowed for the creation of Test-Driven development. These allow for an efficient and streamlined development process minimizing bugs left once a product is deployed. The

creation of Extreme Programming helped develop Agile and is used in at least some capacity by IBM, Ford and ThoughtWorks. Its tenets have influenced other agile type development processes, who implement them in less rigid or strict forms. Kent Beck is still a proponent of these ideas today and over the last few years has tried to help implement his ideas such as Extreme programming and xUnit into the areas of social interaction and health both of which are important parts of the software engineering puzzle.