

Reflection Report on the article Programmer Test Principles

The testing process of development is often a dreaded task with many different testing styles; it can be hard to narrow down how to test effectively. In the article “Programmer Test Principles” (June 2019), the author Kent Beck aims to switch the discussion from testing details to a more productive conversation about testing principles. Beck states that when we talk about testing details, we can agree on the fundamental principles, and despite this, our answers can still differ due to different contexts. However, if we discuss development principles, we create an environment that allows us to understand the differences in contexts and reach a conclusion.

I chose this reading because I find writing Tests for my code takes is far too time-consuming. I struggle to create durable tests that remain stable during long-term development; I am looking for a more efficient and robust way to write tests and am interested to see if a new perspective on testing could help me approach this issue.

Beck describes tests as an oracle, and like an oracle predicts the weather, your tests can predict what will happen if the application is deployed. Beck’s article’s main point is to emphasize the principles behind testing rather than the details. In Beck’s view, tests should be fast, deterministic, predictive, sensitive, and cheap to read, write or change if these core principles are met. It will minimize wasted programming time by reducing wait times for tests; tests will run reliably, respond to the software’s behavior changes, fail when there are structural changes to the code base, and be cost-effective to write and change. Beck’s development outline seems to be closely tied to Behavior-driven development while still taking ideas from test-driven development.

Beck’s core principles in this article are beneficial criteria to incorporate when implementing any type of testing. Whether it is behavior-driven development or test-driven development when implementing testing, it is vital to keep in mind Beck’s principles of development testing.

In the industry, it is vital to cater your agile development process to the context in which you are working. At the beginning of Beck’s article, he disregards the importance of testing styles; this is further enforced by outlining the fundamental principles that all testing should satisfy; this is the main point of Beck’s article. To get universities, industry, and general developers to worry less about code coverage, less about unit testing, to think more about designing tests to suit user functionality, not being too explicit when creating tests, to make them versatile and robust. Doing this effectively would be an exceedingly valuable skill in both industry and my immediate work in building a web application for SENG302, RIP routing protocol for COSC364, and personal projects as a software developer for automation and finance.