

**Sourabh Shetty**

## **A Dissection of the Test-Driven Development Process: Does It Really Matter to Test-First or to Test-Last?**

While I liked the idea of the paper, I found it hard to understand. A lot of the math and the models were nearly incomprehensible for me personally. It could be just a factor of my own inexperience with research papers, but I was able to understand the previous papers for the most part just fine but I felt I struggled with this one.

As for the matter itself, I was surprised that they discovered that the order in which unit tests were written wasn't important. I felt intuitively like it should have, despite not knowing exactly how, so it was interesting to read that it wasn't.

I'd have liked to read a lot more about whether writing the test first and writing the code to make that test pass could have detrimental effects. For instance, and I suggest this as a future research topic: perhaps a programmer could be so heavily influenced by the idea of having the test cases pass and so spends more time on making sure the test cases pass rather than focusing on the code or optimizing it, leading to overall worse code, and perhaps even code that fails test cases that weren't accounted for.

## **Analyzing the effects of test driven development in GitHub**

I see the authors have listed their focusing on just Java-based projects as a threat to validity, but I personally don't have a problem with it. There are so many diverse Java projects on GitHub that I believe that just focusing on Java should be good enough.

I did have a problem with one of the relaxations the authors had made though, wherein they even considered projects where the test files were committed even a week after the source code as test-driven. While they very were might be in some cases, it still breaks the very idea of writing tests first, since the authors don't verify that.

While I am satisfied with their focus on Java, as a future research idea, I'd like to see comparisons against projects on GitHub from different languages as well.

## **What Do We (Really) Know about Test-Driven Development?**

I don't believe this was a very informative paper. I don't feel like I learned anything new by reading it. This seemed less substantive than even the abstracts of the some of the previous papers.

One thing I did like was that it referred to Test-Driven Development as 'controversial'. While the previous two papers also focused on TDD, they didn't give me any impression as to how TDD was actually perceived, and so finding out that it was seen as controversial made a lot of sense and shed some new context on the entire topic for me.

As a future research idea, maybe some more discussion and research into why TDD isn't adopted more widely.