

TDD

What They Forgot to Tell You

Grzegorz Piwowarek

@pivovarit

{ 4comprehension.com }

Senior Software Engineer @ Hazelcast

Independent Trainer

@pivovarit



Chris Birchall
@cbirchall



How to TDD

1. Write the code
2. Write the tests
3. `git rebase -i` to reorder your commits

4:18 PM · Oct 1, 2020 · Twitter for iPhone

959 Retweets **94** Quote Tweets **5.1K** Likes

<https://twitter.com/cbirchall/status/1311671776007593985>

TDD Cycle

1. Write a (failing) test
2. Write the code
3. Run tests
4. Refactor (or fix the code and go back to 3)

TDD Benefits

1. Better design
2. Shorter development time ([source](#))
3. Fewer bugs ([source](#))
4. It's professional (solutions come after a problem is formulated)

How does TDD improve the design?

How does TDD decrease the development time?

How does TDD decrease the number of bugs?

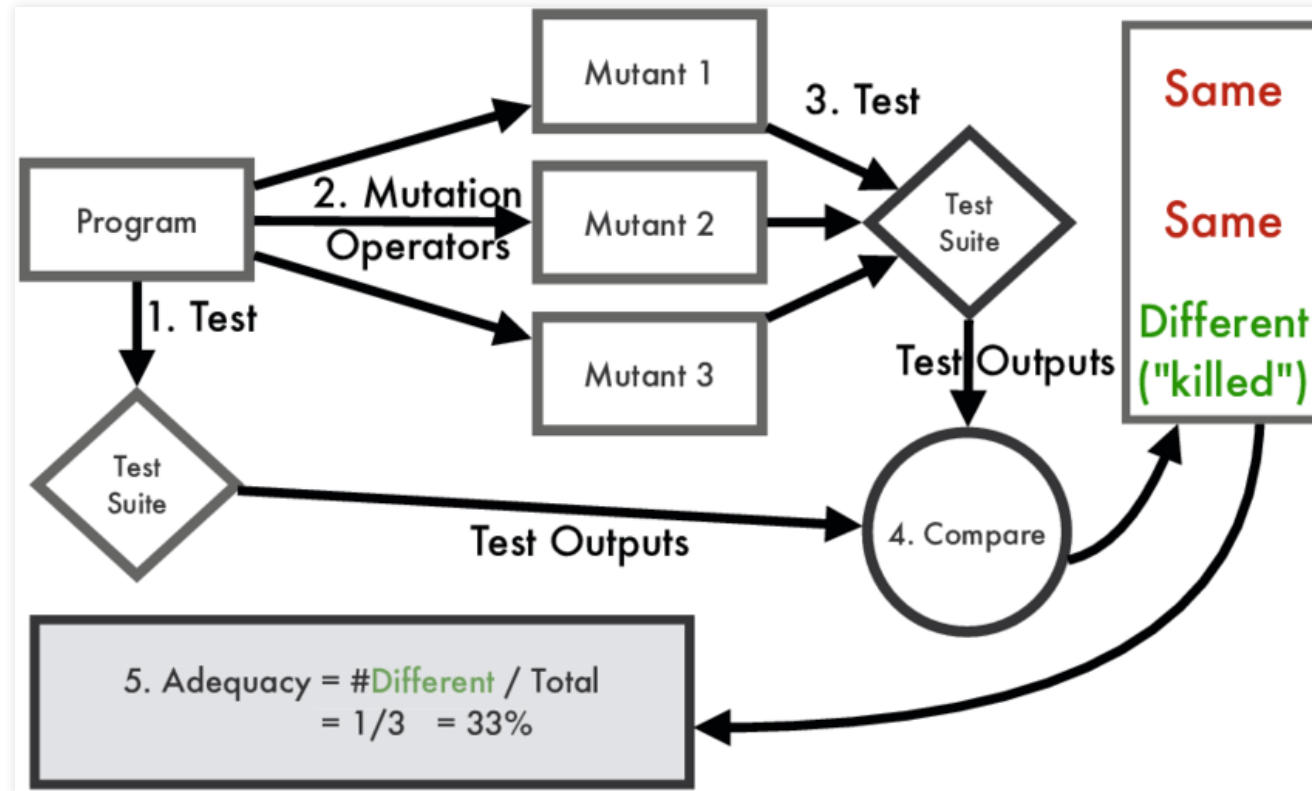
Why it's professional?

insert a demo here

Quis custodiet ipsos custodes?
"Who will guard the guards themselves?"

Mutation Testing

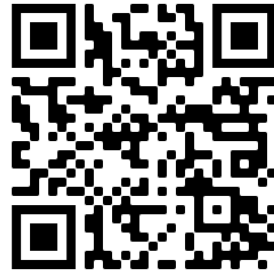
Mutation Testing



- Test-Driven Development by Example, Kent Beck
- Improving your Test Driven Development in 45 minutes, Jakub Nabrdalik

Thank You!

<https://pivovarit.github.io/talks/tdd>



Twitter: @pivovarit

