

WITH TDD?

Who am I?

Pedro Osternack Correa

- Solution Architect at Slalom _build
- Working on software development for 16 years
 - Experiences on healthcare, oil and gas plants and pipelines, logistics, gas pipelines, accounting, electronic document management, IT infrastructure and automation, cloud migrations.
- Living in Canada for 7 years.
 - The last 2 years in Ottawa.
- ~~Coffee-addicted~~ Home barista
- Juggler in training
- Drummer wannabe
- Find me on socials: @pedrosc

Agenda

- What is TDD?
- TDD cycle
- Anatomy of a test
- Live demo
- Q&A

What is TDD?

What isn't TDD?

- Testing.
 - No, really.
- High level architecture and design
- Mocks
- Over 9000% of code coverage



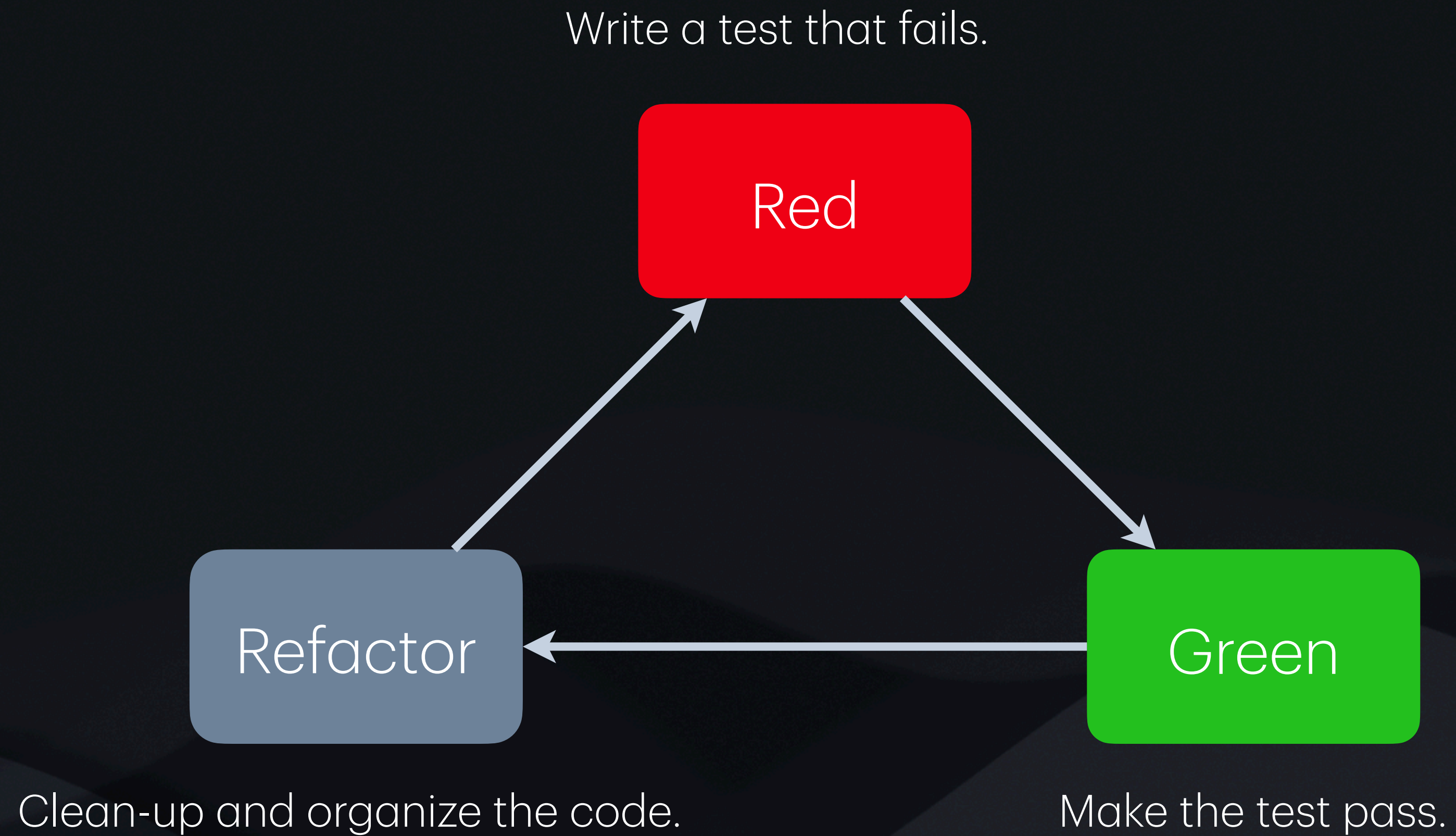
What is TDD

What is TDD About?

- Understanding requirements
- Code Design
- Behaviour
- Fast feedback
- "Separation of concerns"
- Delivering value through maintainable and expressive code
- Experimentation



The TDD cycle



Red

Pick a problem to solve

- Pick a problem, or part of a problem, you will solve.
 - Break your requirements down into workable pieces.
- Define the contract of your solutions
 - What is needed?
 - How will it be called?
 - What will it return?
- Create an usage example of your solution. (aka: a Test)



Green

Make it work

- Implement a solution for the problem you're working on.
 - What is the bare minimum that I need to satisfy my requirements?
- Quick and dirty, but backwards compatible.
 - You don't want to break any other existing tests.
- If you're spending too much time trying to make your test pass:
 - You may be trying to solve a problem that is too big
 - Take a step back, return to the red phase and try to break the problem down further.
 - You may be trying to solve too much at once.
 - Focus on the red test



Refactor

Make it right

- "Refactor to remove duplication."
- Here you'll think about the design of your code.
 - How can I make the solution more cohesive and maintainable?
 - Are there any design patterns that I can implement?
- Refactoring means to change form without changing behaviour.
 - You shall not ~~pass~~ break any test on this phase
 - Neither implement new functionality.



When to use TDD?

When to use T to D your D?

- Whenever possible.
- Whenever it's convenient.
- Legacy systems may be tricky, but there are tools that can help you.

TDD is a tool

Anatomy of a test

Test structure

- AAA - Arrange/Act/Assert
 - Arrange the desired initial state of the system
 - Take a Action on that state
 - Assert that the result if the one you expect
- Given/When/Then
 - Given that the system in on a certain state
 - When I take an action
 - Then I should see X result.

Live Demo

String Calculator

Create a simple String calculator that can receive a string with positive values lower than 1000 and sum all the values in that string. The values may be separated by commas, new lines or both. The user can define a custom separator by passing it preceded by "\\" in the first line of the string. If the custom separator has more than one character it should be contained by square brackets. A list of custom separators can also be defined, in that case each separator should be in square brackets. When custom separators are defined, only those should be used in the values string.

Q&A

Thank you!

<https://linktr.ee/pedrostdc>

