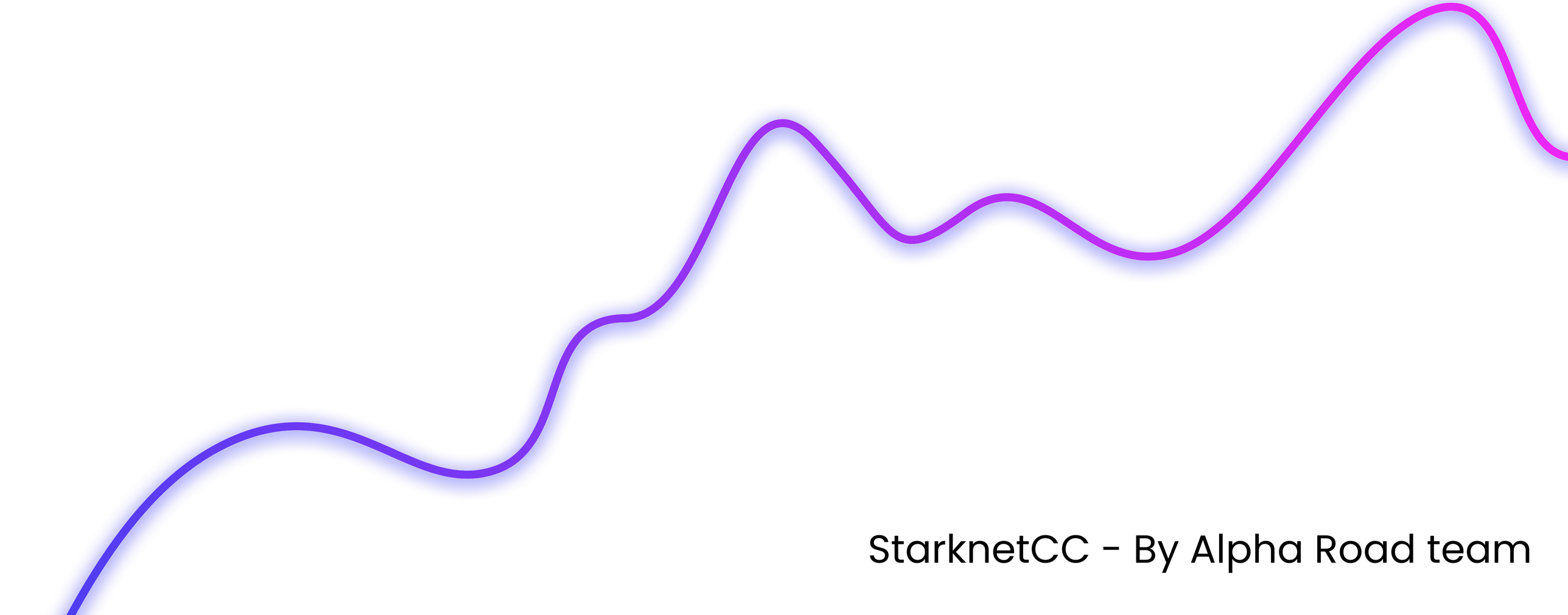
TDD to embrace Cario



StarknetCC - By Alpha Road team


TESTER
C'EST DOUTER
COMMITSTRIP.COM

Who we are

Alpha Road

starknet-ecosystem.com

Florian Bellotti

Senior Software Engineer

Twitter: @FlorianBellotti

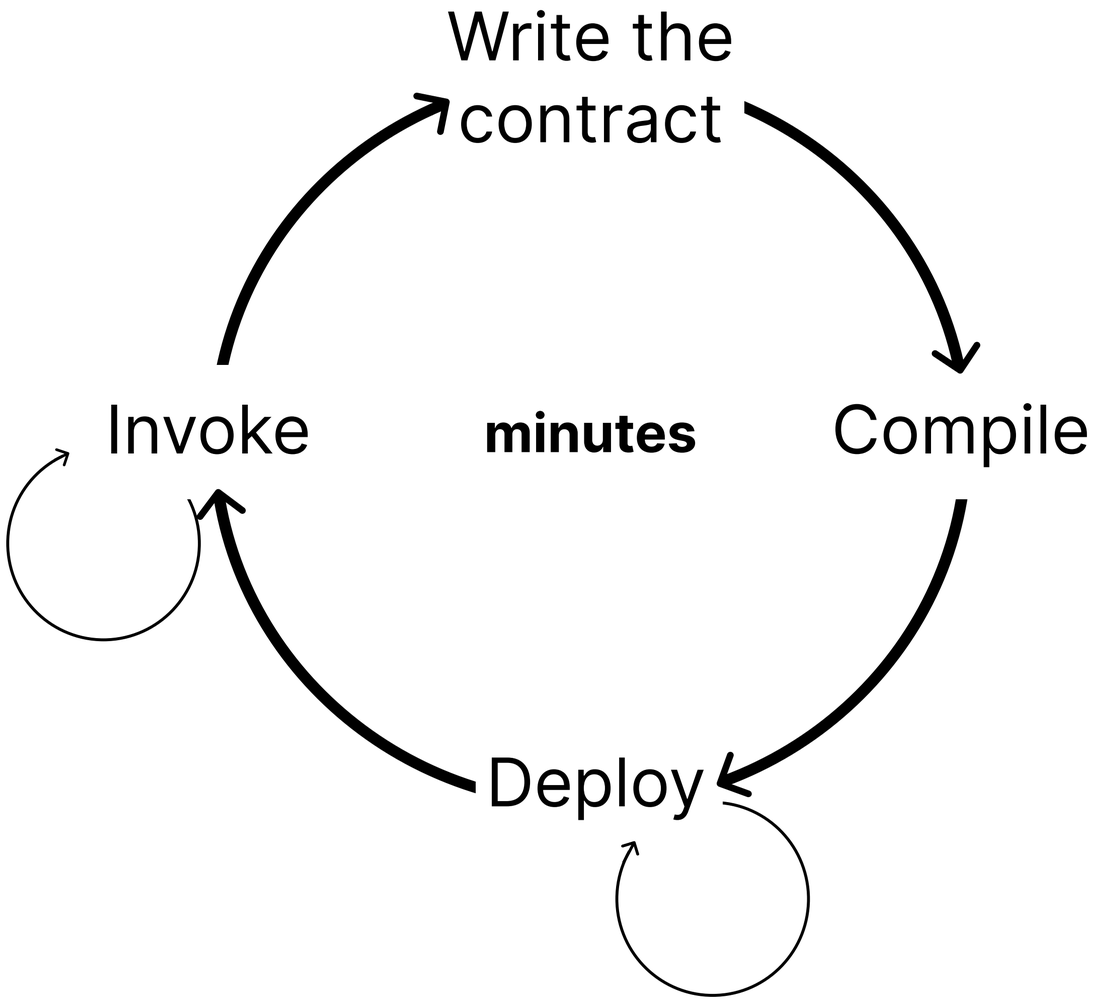
Lenny Aebischer

Co-founder & CTO @ Alpha Road

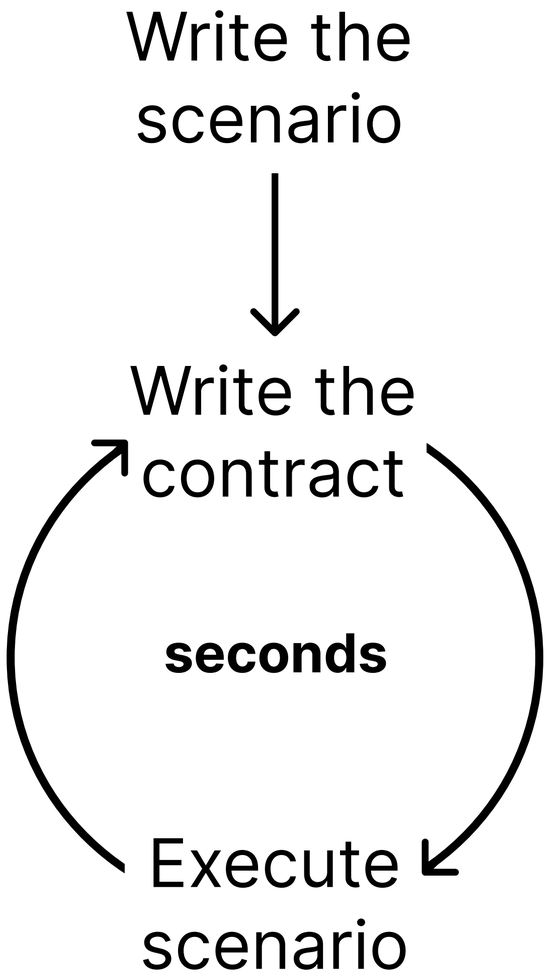
Twitter: 0xlenny_eth

Don’t waste your time

Retry ?



vs



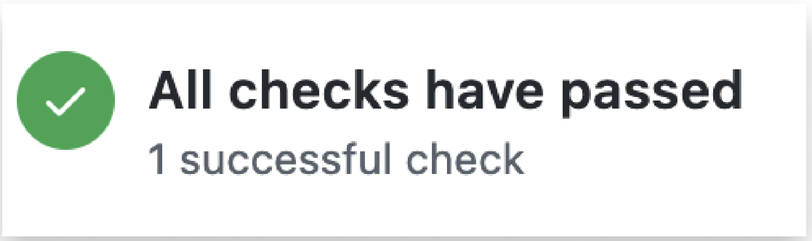
Fight fear

I’m not scared

The code of your contracts is sensitive

Tests allow to:

- Don’t be afraid to refactor, clear or improve your code
- Saving knowledge
- Preserve quality



How to test ?

With **python** ?

- Just use pytest
- Pros: Many libraries, language is simpler, better tooling, better readability...

Or **Cairo** ?

- You can use Protostar
- Cons: Your tests a written in Cairo
- Pros:
  - Your tests a written in Cairo
  - Real unit tests, faster, allow advanced features (mocks, catch events...)

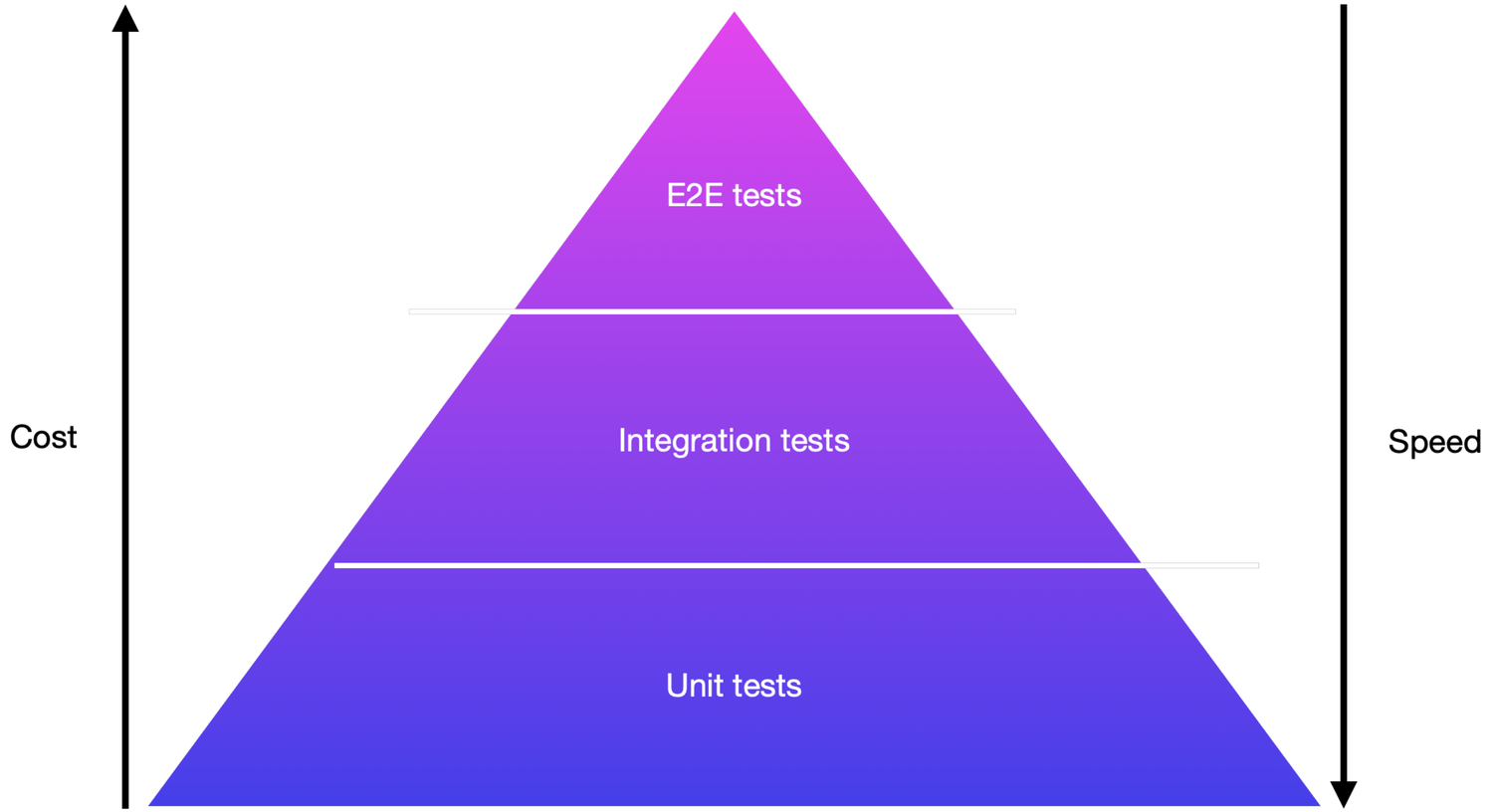
What should you test ?

Test's pyramid

At least write **integration tests**

When using **pytest** you can’t write unit tests, and integration tests can be slow

With **protostar**, you can isolate the code you want to test and have more confidence in you code



Test Driven Development

Not for code coverage

TDD != Test First

Not just a way to write tests, but **more a way to develop**

- It helps to reflect, specify and build
- Code coverage is just a bonus, not the goal

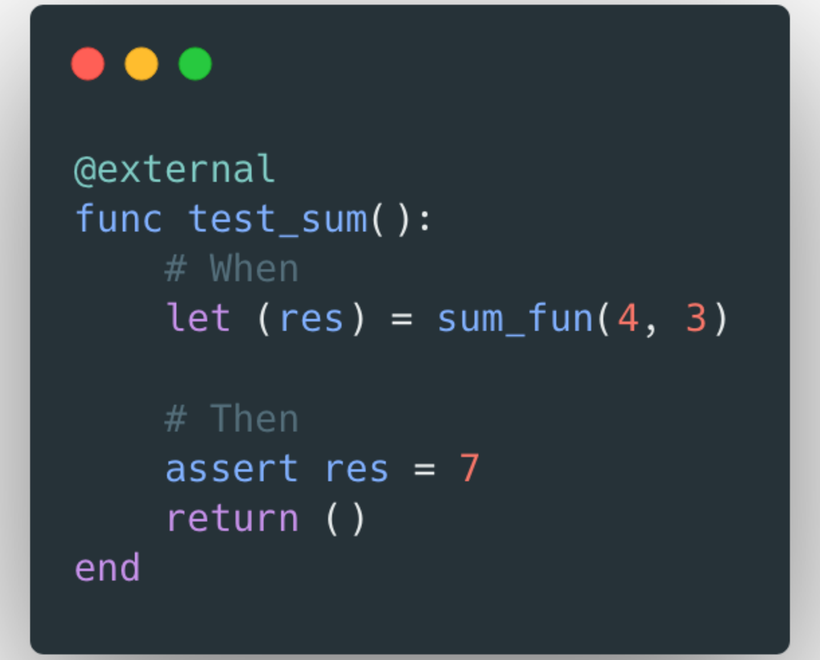
Feedback loop is shorter

Tests are the documentation

Writing tests = decouple code = Better design (flexible, maintainable)

Fight fear

Unit test with Protostar



```
@external
func test_sum():
    # When
    let (res) = sum_fun(4, 3)

    # Then
    assert res = 7
    return ()
end
```

Contract test with Protostar

```
@view
func __setup__():
    %{ context.contract_address = deploy_contract("./AccessController.cairo", [10]).contract_address %}
    return ()
end

@external
func test_free_slot_count_should_return_remaining_slot_number{syscall_ptr : felt*, range_check_ptr}():
    # Given
    tempvar contract_address
    %{ ids.contract_address = context.contract_address %}

    # When
    let (free_slots_count) = IAccessController.freeSlotsCount(contract_address=contract_address)

    # Then
    assert free_slots_count = 10
    return ()
end
```

Contract test with Python

```python
@pytest_asyncio.fixture
async def contract() -> StarknetContract:
    starknet = await Starknet.empty()
    return await starknet.deploy("contracts/AccessController.cairo", constructor_calldata=[10,
owner.public_key])


@pytest.mark.asyncio
async def test_transfert_ownership_should_succeed_when_caller_is_owner(contract: StarknetContract):
    # When
    await contract.transferOwnership(address1.public_key).invoke(caller_address=owner.public_key)

    # Then
    execution = await contract.getOwner().call()
    assert execution.result.owner == address1.public_key
```

Your turn

Starknet-cc-tdd

Clone https://github.com/419Labs/starknet-cc-tdd

Setup protostar or pytest (see README.md)

1. Test and complete boolean.cairo (if protostar) or boolean_contract.cairo (if pytest)
2. Test and complete main.cairo
3. Test and complete AccessController.cairo