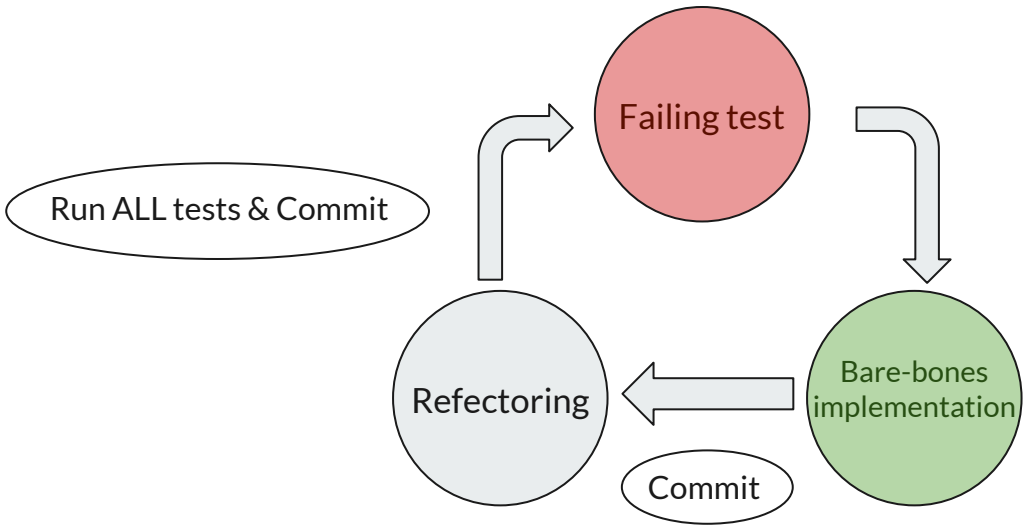
Test-Driven Development (TDD)

Personal Takeaways by Maria Schönholzer

3 Steps


Failing test
Bare-bones implementation
Commit
Refectoring
Run ALL tests & Commit

Test

DOs & DON'Ts

- Naming
  - MyAmazingClass**Should**
    - beNice_WhenAskedNicely()

```
class TicTacToeShould {

    @Test
    public void setCurrentPlayerToXInitially () {
```

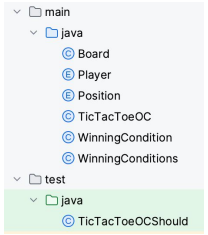
- Structure: Arrange/Act/Assert

```
@Test
public void setCurrentPlayerToXAfterOPlays() {
    TicTacToe board = new TicTacToe();
    board.play(Position.MIDDLE_LEFT);

    board.play(Position.MIDDLE_LEFT);

    String player = board.getCurrentPlayer();
    assertEquals( expected: "X", player);
}
```

- Test Behaviour not Implementation
  - One test class -> Multiple production classes



Test

DOs & DON'Ts

- One test at a time -> test single behaviour
  - multiple 'assert' statements possible
- Use testing framework to avoid repetition
  - `@ParameterizedTest`
  - `@BeforeEach`
- Test from point of view of API user

Implementation

DOs & DON'Ts

- OK to start with fake implementation
- 'Just enough' to pass a test
- Use IDE
  - Context Actions in IntelliJ
- Green test -> Commit!

Refactoring

DOs & DON'Ts

- Never with RED tests!
- Use IDE
  - Refactoring or Context Actions in IntelliJ
- Do not start refactoring too early
  - Rule of 3 repetitions
  - 'Right' abstraction
    - IF-condition vs IF-block as a whole
- Guidelines
  - Transformation Priority Premise (TPP)
  - Object Calisthenics
- Run ALL tests and commit if green

“Learn the rules like a pro, so you can break them like an artist.”

Pablo Picasso

Further Reading

- TDD
  - by Kent Beck https://tidyfirst.substack.com/p/canon-tdd
  - by Martin Fowler https://martinfowler.com/bliki/TestDrivenDevelopment.html
- TPP
  - by Robert C. Martin (Uncle Bob) https://blog.cleancoder.com/uncle-bob/2013/05/27/TheTransformationPriorityPremise.html
- Object Calisthenics
  - by Jeff Bay https://www.bennadel.com/resources/uploads/2012/ObjectCalisthenics.pdf