

How Unit Testing Saved My Career

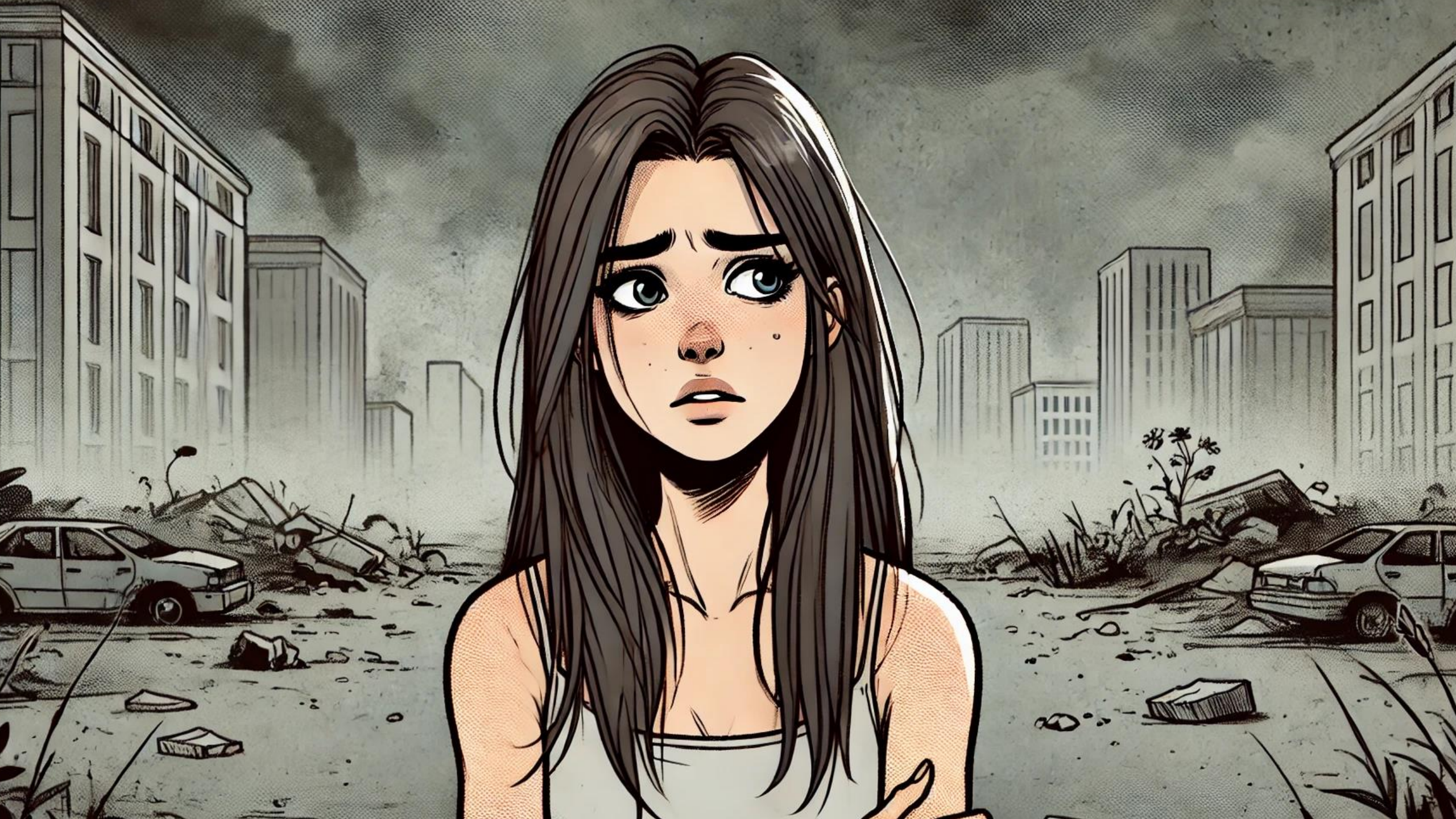
Annelore Egger



About Me

- Career Saved By Unit Tests
- BSc in Computer Science
- 8 years of programming experience
- Software Engineer, OpenValue Switzerland since May 2024







JUnit

```
@Test
void deposit_validAmount() {
    // Given
    account = new BankAccount(new BigDecimal("100.00"));

    // When
    account.deposit(new BigDecimal("50.00"));

    // Then
    assertEquals(new BigDecimal("150.00"), account.getBalance());
}
```

JUnit

```
@Test
void withdrawMoreThanBalance_expectException() {
    // Given
    account = new BankAccount(new BigDecimal("100.00"));

    // When, then
    assertThrows(IllegalArgumentException.class,
        () -> account.withdraw(new BigDecimal("200.0")));
}
```

Examples

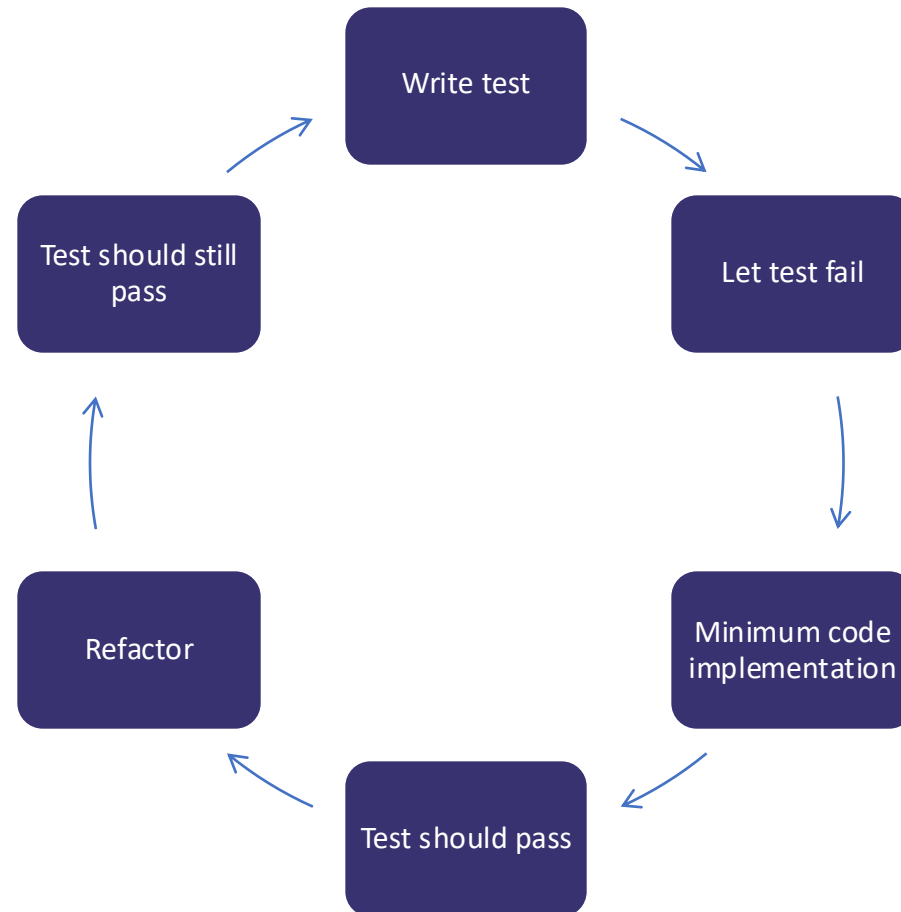
Testable Code...

- ... is understandable code
- ... is cleaner code
- ... gives fast feedback
→ fail fast
- ... gives confidence
- ... is easier to maintain



OPENVALUE
Better software, faster

TDD (Test-Driven Development)

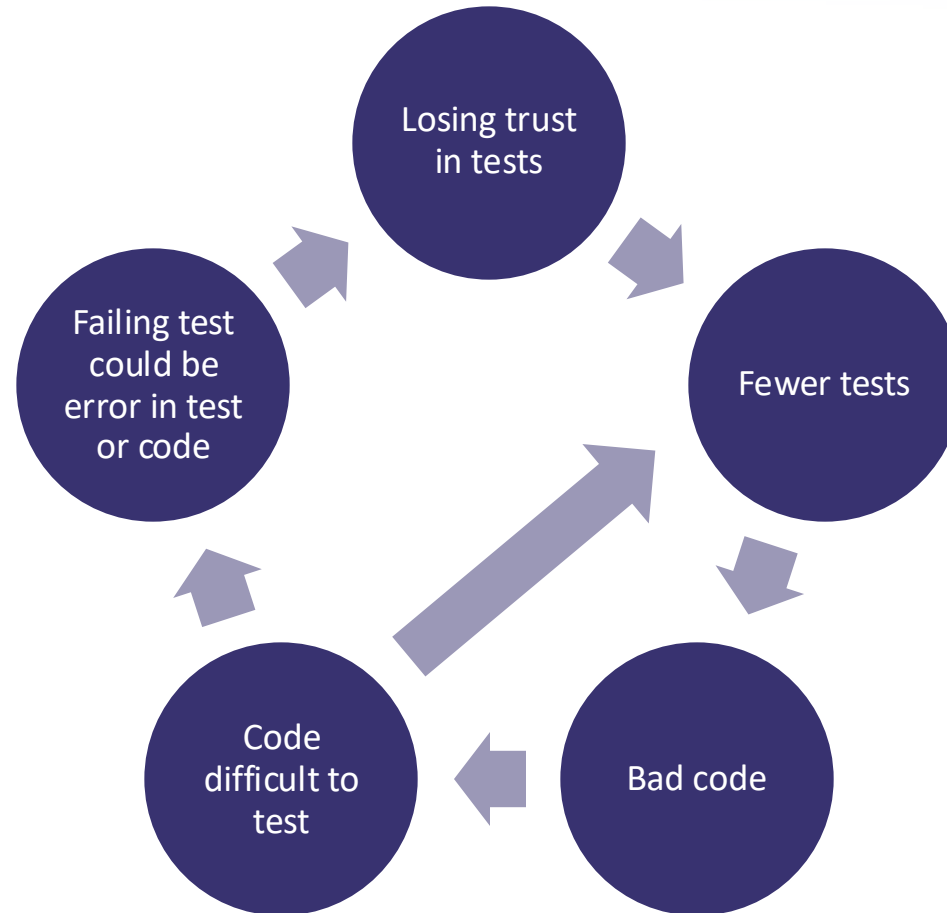


Mockito

- Simplify test setups
- Creating unit tests that truly **only** test the unit, as the other aspects can be mocked away

Examples

What happens if there are almost no tests



Parametrized Tests

```
@ParameterizedTest
@CsvSource( {
    "2, 4",
    "3, 9",
    "4, 16"
})
void testWithCsvSource(int input, int expected) {
    assertTrue(input * input == expected);
}
```

Parametrized tests

- Seeing at one glance which cases were tested, and if some cases are missing
- Easier to cover all the cases
 - Boundary values
 - Negative and positive tests
- Developers rather add additional test cases

Parametrized Tests

- Multiple possible sources make parametrized tests possible in all kinds of situations (e.g. `@ValueSource`, `@EnumSource`, `@MethodSource`)
- See [Guide to JUnit 5 Parameterized Tests](#)

AM I TESTING MY CODE

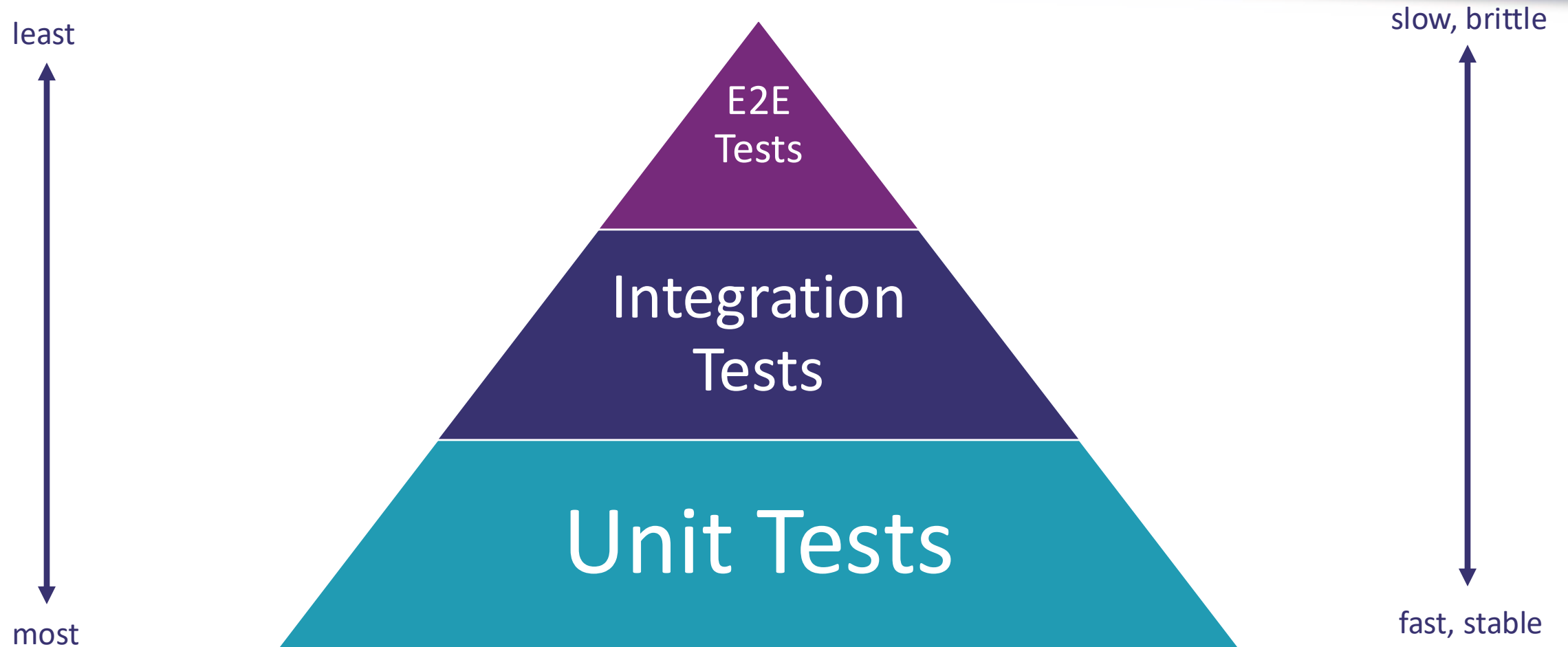


OR IS IT TESTING ME?

Challenging complex setup

- **NOT:** We shouldn't do it like this!
BUT: Why are we doing it like this?
- Is the code too complex?
- Could mocks be introduced to reduce complexity?
- Does the test setup only contain the necessary things?
- Is the test on the right level on the testing pyramid?

Testing pyramid





Thank you for your attention



✉ annelore.egger@openvalue.ch

🐙 [anneloreegger](https://github.com/anneloreegger)

in [/in/anneloredev](https://www.linkedin.com/company/anneloredev)

✂ [@AnneloreDev](https://twitter.com/AnneloreDev)

▶ [@anneloredev](https://www.youtube.com/channel/UC...)

Further Resources

- [SRP](#)(Single Responsibility Principle)
- [The Practical Test Pyramid \(Martin Fowler\)](#)

Sources

- [ISTQB Glossary](#)