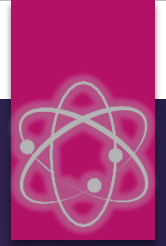
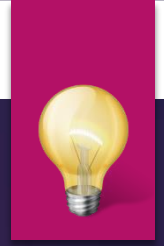
Technical Practices

Never underestimate
the Simplicity



Three laws of TDD

- Failing test (for the right reason) **RED**
- Code to pass the test (just sufficient, fake or obvious) **GREEN**
- Refactor (generalize) just as much as you really test **TRIANGULATION**

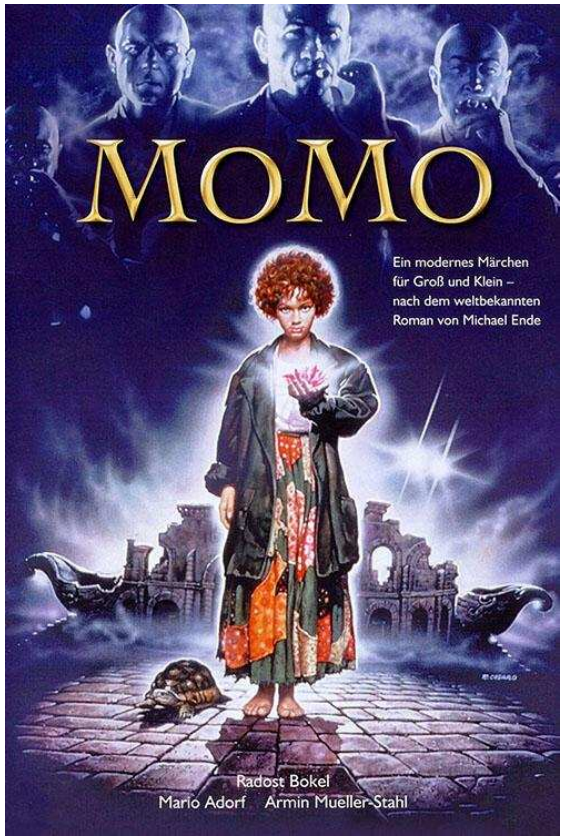


Triangulation is hard

- Think that you only know the **public API** of your Code
- Using multible indirect test to assert **it behaves how you thought** it should
  Do **not test the inner** implementation
- The design should become obvious, when it **repeats** or getting **redundant**
- Be careful in adding new behavior before you tested the **existing** behavior
- Think about **boundaries**

TDD in Mob or Pair: Keep slow to arrive fast

- Nobody is left behind (community cohesion)
- Take initiative when partner is stuck, what lowers frustration
- Learning habits from your mates
- Help to find a «ubiquitous language»
- Following the same «standards/practices»
- Duplication as an intermediate step (rule of three)

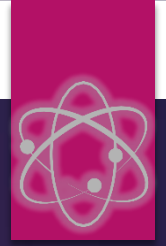


https://www.moviepilot.de/movies/momo-2

Great Habits: F.I.R.S.T

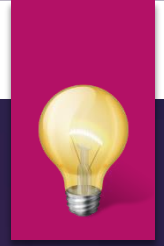
- **Fast**: so you dont mind to run the test often
- **Isolated**: the order of test doesn't matter
- **Repeatable**: same Result
- **Self validating**: red or green – no interpretation
- **Timely**: written before the code

https://schnaeppchenfuchs.com/freizeit/carrera-digital-132-start-set-fuer-199e-carrera-bahn-mit-f1-boliden



Great habits

- Start from the assertion
- Test structure **A**rrange **A**ct **A**ssert
- Names Testclass**Should**
  - doSomethingThatIsExpected → lead to a readable «high level documentation»
- Fail for the **right reason**
- Meaningful **feedback** of tests
- Write the **simplest** code to pass the test
- **Rule of three** to tackle duplication

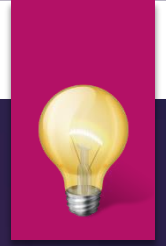


Thougths about Tests in Junit 5

- Why should i use this? **@DisplayName**(“it does Something I expected")

- @TestMethodOrder(OrderAnnotation.class)
  - @Test
    @Order(1)
    void yourFirstTest(){…
  - @Test
    @Order(2)
    void yourSecondTest(){…

- @TestMethodOrder(MethodName.class)
- @TestMethodOrder(Random.class)
- @TestMethodOrder(CustomImplementation.class)
  → CustomImplementation implements MethodOrderer



Unit Test Naming Conventions

- MethodName_StateUnderTest_ExpectedBehavior / MethodName_ExpectedBehavior_StateUnderTest `isAdult_AgeLessThan18_False`
  - → Method name changes? Length?
- test[Feature being tested] / just the Feature `testIsNotAnAdultIfAgeLessThan18`
- Should_ExpectedBehavior_When_StateUnderTest `Should_ReturnFalse_When_AgeLessThan18`
- When_StateUnderTest_Expect_ExpectedBehavior `When_AgeLessThan18_Expect_isAdultAsFalse`

TPP: Transformation Priority Premise (shortened)

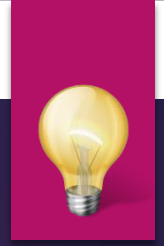
- Constant 1
- Scalar argument 1+2
- Statement arguments 5+3-4
- Conditional If(condition) return argument
- Array [«dog», «cat»]

https://transformers.fandom.com/de/wiki/Bumblebee_(Movie)

TPP: Transformation Priority Premise (shortened)

- Container {Dog, Cat}
- Loop/recursion for(){} while(){}
- Function calculate()
- Mutation Variable
  - String truth = «is yours»
  - truth = «has changed»
- Switch case switch(){ case A: … break; case B: … break;default: …}

https://transformers.fandom.com/de/wiki/Optimus_Prime_(Movie)



Rules of the Mob

- Respect, don't harm
- Be careful with jokes and «running gags»
- Mind to get peoples solutions or intentions → maybe it leads to the same or even better refactorings (be patient)
- Important Setup: Driver, Navigator, Mob, Timer

https://www.thatmomentin.com/gangs-of-new-york-2002-and-the-fight-at-five-points-moment/the-priest-gangs-of-new-york/

Object Calisthenics

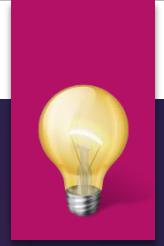
- One level of intention per method
- Don't use ELSE
- Wrap all primitives and strings
- Wrap all collections to Classes
- One dot per line

https://en.wikipedia.org/wiki/Popeye

Object Calisthenics

- No abbreviations ~~WTF YOLO~~
- Keep all entities small: packages, lines per class/method, arguments per mehtods
- A class has in maximum two instance variables
- No public getters/setters/properties

https://justenglish.me/2012/07/11/dont-panic/



Mob: What is the obvious implementation

- Go for what the mob understands, refactor together
- its hard to name the test, what is the intention of your next code step

https://www.taipeitimes.com/News/feat/archives/2003/08/22/2003064830

Questions

https://www.reddit.com/r/nomanshigh/comments/6gh7cc/i_build_a_super_computer_in_no_mans_sky_finally_i/

Thank you for your attention

Res Gilgen
resgilgen@gmail.ch

Resources

Agile Technical Practices Distilled

(Pedro Moreira Snatos, Marco Consolaro, Alessandro Di Gioia)

Alcor Academy Lessons

(Marco Consolaro, Alessandro Di Gioia)

Junit Execution Order

https://mkyong.com/junit5/junit-5-test-execution-order/
https://junit.org/junit5/docs/current/user-guide/#writing-tests-test-execution-order-methods

Unit Test Naming Conventions

https://dzone.com/articles/7-popular-unit-test-naming

https://www.pngrepo.com/svg/127141/atomic-theory

https://www.iconfinder.com/icons/87436/idea_icon