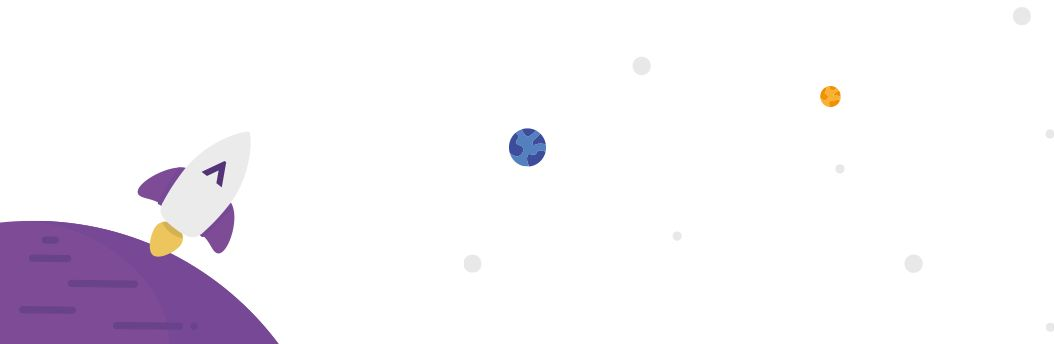
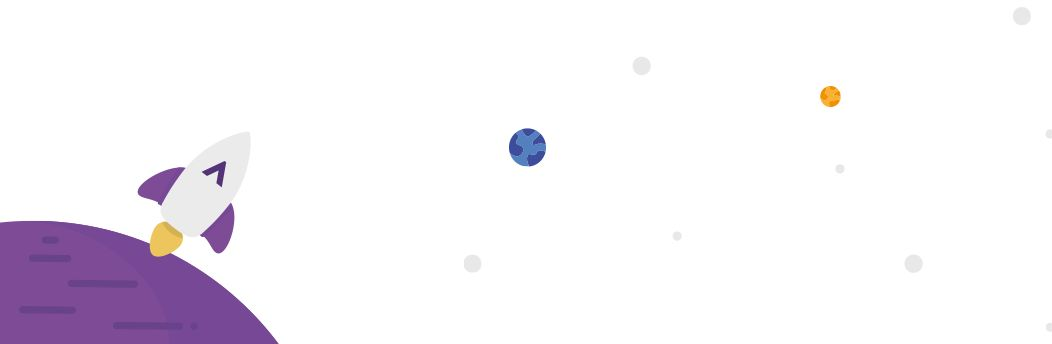


AVISI



AVISI

Test-Driven Development (TDD)



Even voorstellen

Dirk Groot

43 jaar oud

getrouwd

3 kinderen (13, 15, 18)

woont in het prachtige Arnhem

Functie

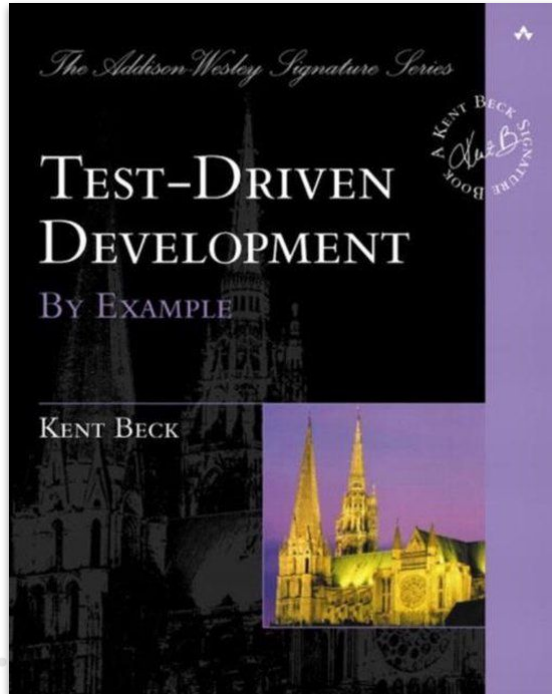
Software Developer/Architect bij [Avisi](#)

Historie

Software development, design en architectuur in radio/tv, overheid en fintech



November 2002

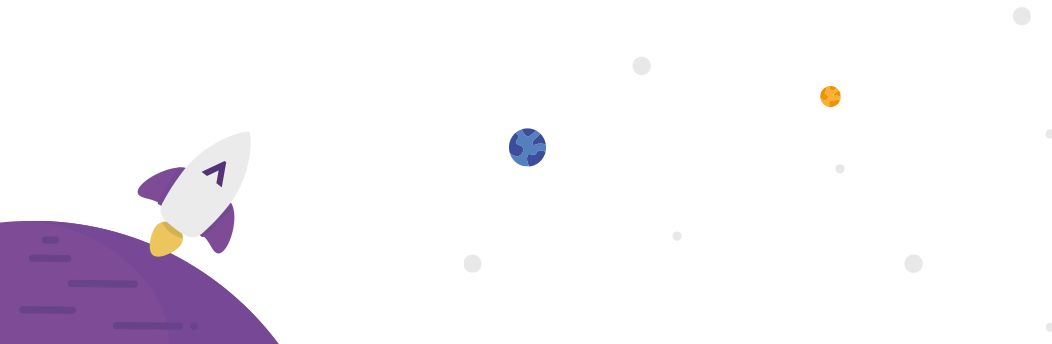


De basis [1 / 2]

2 Regels

1. Write a failing automated test before you write any code.
2. Remove duplication.

Bron: Test-Driven Development By Example

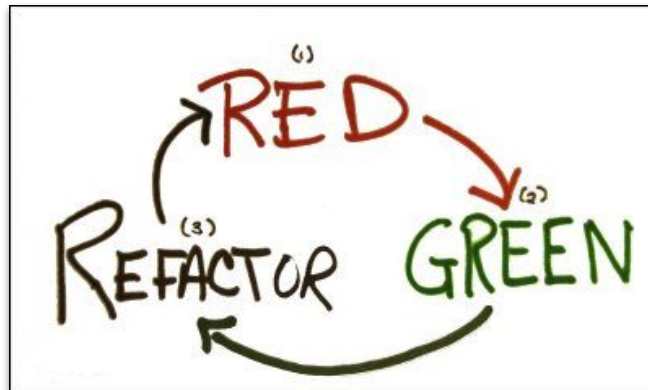


De basis [2/2]

De red/green/refactor cyclus

1. **Red** - Write a little test that doesn't work, and perhaps doesn't even compile at first.
2. **Green** - Make the test work quickly, committing whatever sins necessary in the process.
3. **Refactor** - Eliminate all of the duplication created in merely getting the test to work.

Bron: Test-Driven Development By Example



Demo: FizzBuzz

```
public class FizzBuzz {  
    public static String say(int i) {  
        // TODO  
    }  
}
```

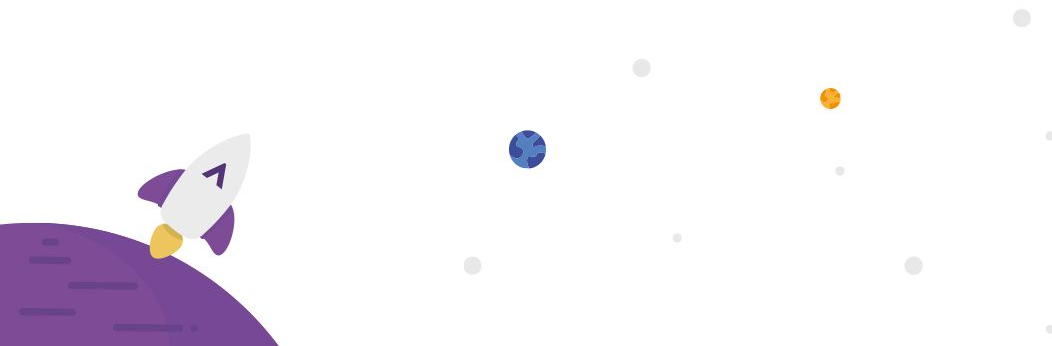
Postcondities:

- Return **"Fizz"** als **i** deelbaar is door **3**
- Return **"Buzz"** als **i** deelbaar is door **5**
- Return **"FizzBuzz"** als **i** deelbaar is door **3** én **5**
- Return **i** als string in alle andere gevallen



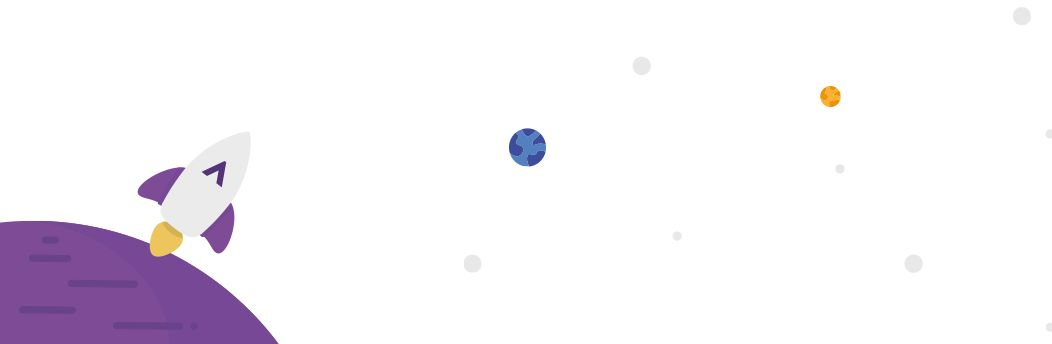
Ontwikkelingen

- Elke programmeertaal heeft een unit test framework
- IDE's ondersteunen unit testing
- Build tools ondersteunen unit testing
- Unit tests in CI/CD pipelines
- BDD/ATDD



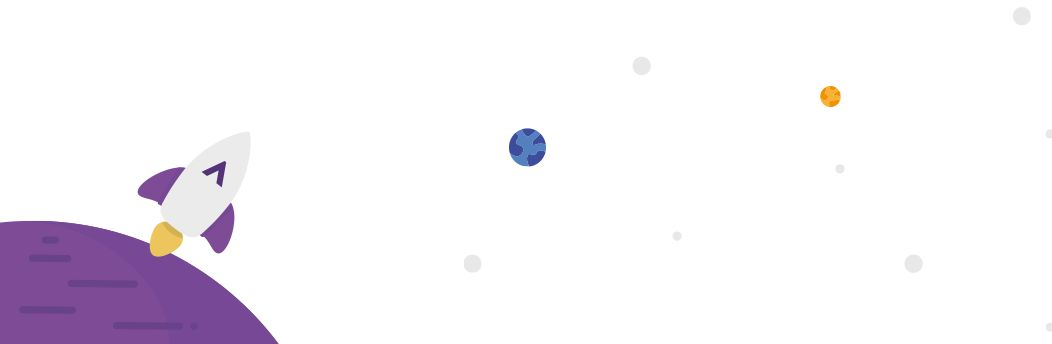
Het “verkoopverhaal” van TDD

- Geautomatiseerde regressietest suite
- Vertrouwen
- Refactoring zonder angst
- Kwaliteit
- Documentatie
- Korte feedback loop
- Alle tests geslaagd? → Klaar!



Waarom test-first?

- De kans om de test zelf te testen
- Interface design
- Je weet waarom je een test schrijft
- Achteraf tests schrijven is saai



April 2014

TDD is dead. Long live testing.

By David Heinemeier Hansson on April 23, 2014

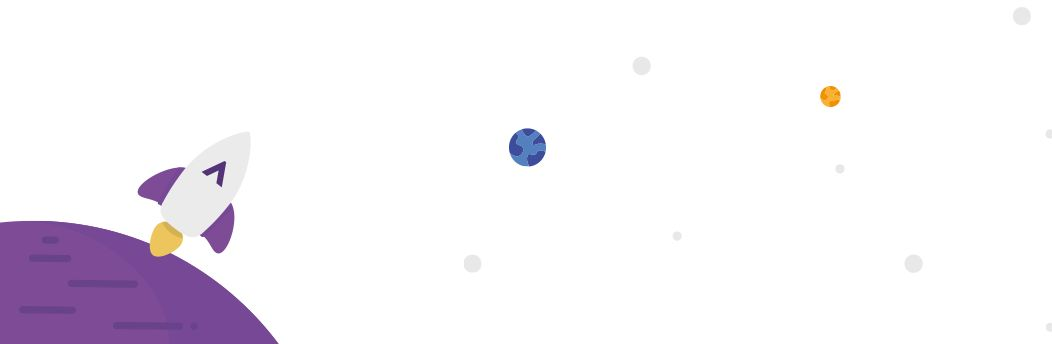
Test-first fundamentalism is like abstinence-only sex ed: An unrealistic, ineffective morality campaign for self-loathing and shaming.

<https://dhh.dk/2014/tdd-is-dead-long-live-testing.html>



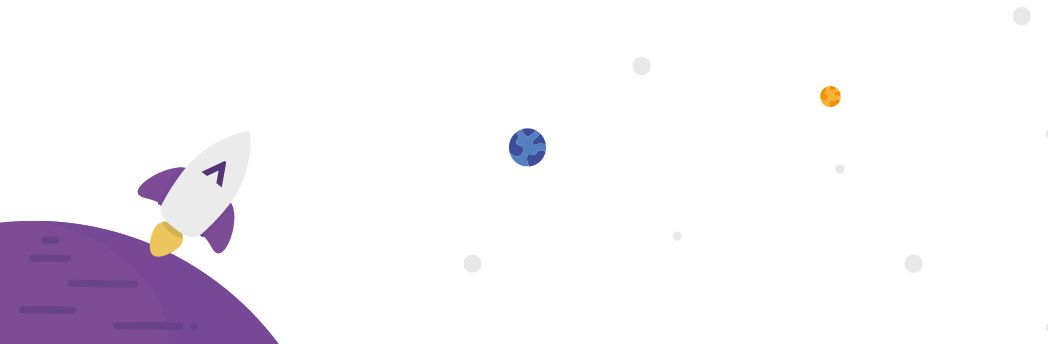
De realiteit

- Trage test suites
- Flaky tests
- Lage testcoverage
- Refactoring → falende tests
- “Wat wordt hier getest???” 🤖
- Veel boilerplate code
- Mocking frameworks



Observaties

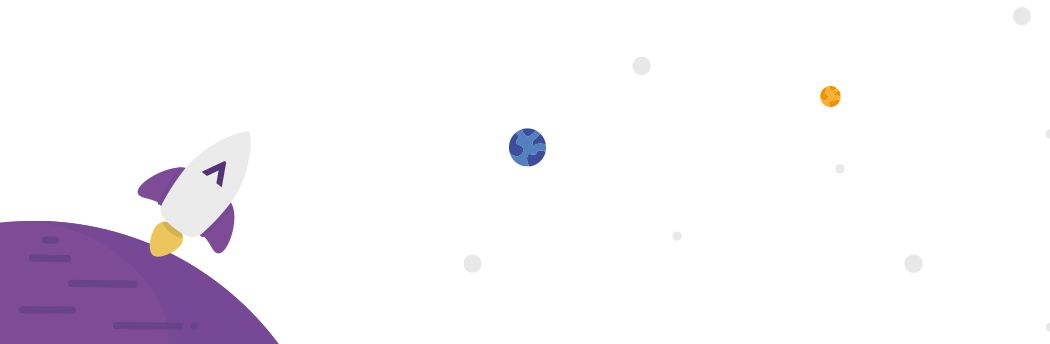
- Elke class in de productiecode heeft een corresponderende unit test class
- Classes worden in isolatie getest, collaborators worden gemocked
- Gebruik van mocking frameworks in tests
- Unit tests maken gebruik van de dependency injection container
- Unit tests maken gebruik van een (in-memory) database



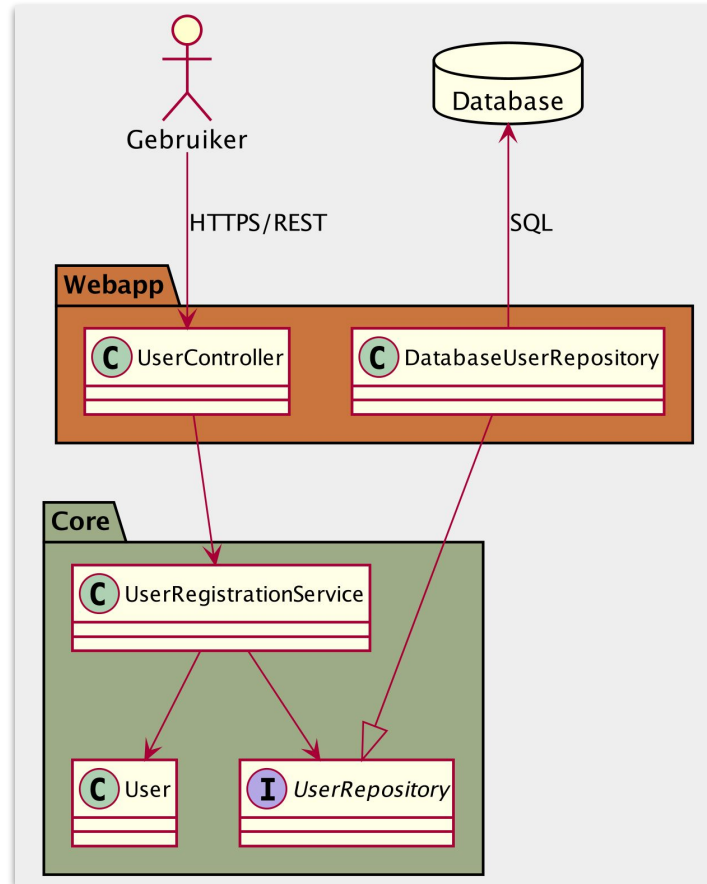
Case study: Registreren en inloggen

Features

- Nieuwe gebruiker registreren
 - Gebruiker kiest gebruikersnaam en wachtwoord
- Inloggen

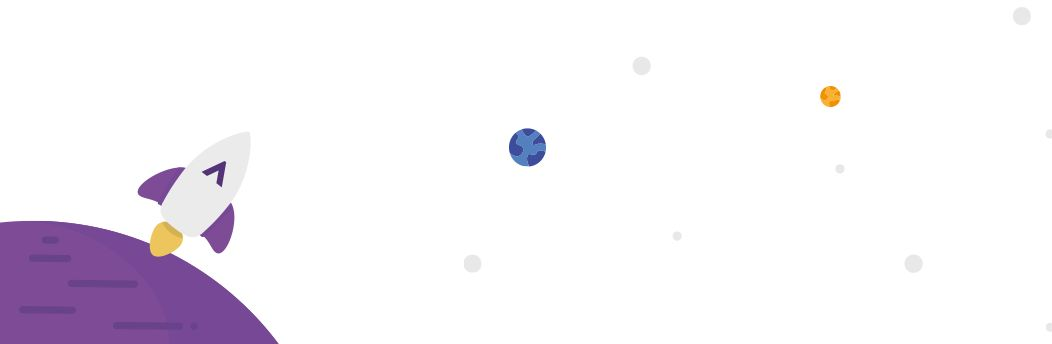


Architectuur



Tips

- Test tegen stabiele interfaces
- Mock collaborators die het testen moeilijk of traag maken
- Geen mocking code in unit tests
- Humble object voor lastig te testen classes



Test Desiderata (Kent Beck) [1 / 2]

- **Isolated** — tests should return the same results regardless of the order in which they are run.
- **Composable** — if tests are isolated, then I can run 1 or 10 or 100 or 1,000,000 and get the same results.
- **Fast** — tests should run quickly.
- **Inspiring** — passing the tests should inspire confidence
- **Automated** — tests should run without human intervention.
- **Specific** — if a test fails, the cause of the failure should be obvious.
- **Deterministic** — if nothing changes, the test result shouldn't change.
- **Predictive** — if the tests all pass, then the code under test should be suitable for production.

https://medium.com/@kentbeck_7670/test-desiderata-94150638a4b3

Test Desiderata (Kent Beck) [2 / 2]

- **Writable** — tests should be cheap to write relative to the cost of the code being tested.
- **Readable** — tests should be comprehensible for reader, invoking the motivation for writing this particular test.
- **Behavioral** — tests should be sensitive to changes in the behavior of the code under test. If the behavior changes, the test result should change.
- **Structure-insensitive** — tests should not change their result if the structure of the code changes.

https://medium.com/@kentbeck_7670/test-desiderata-94150638a4b3



De moraal van dit verhaal

Test-first

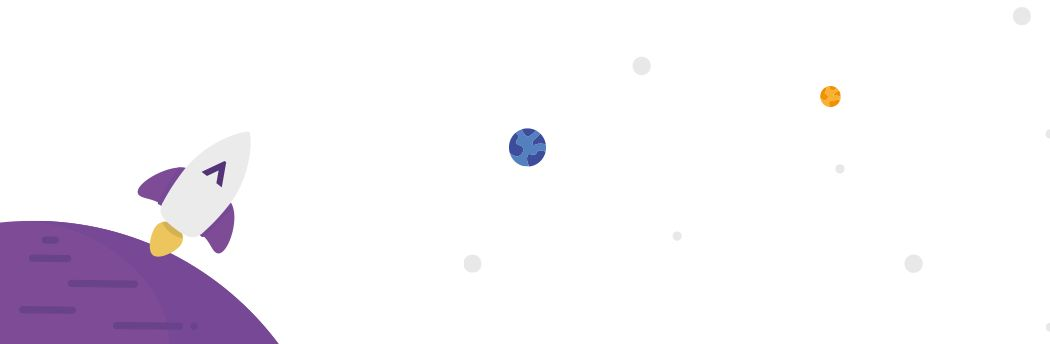
Gewoon doen!

Coupling

Ontkoppel unit test code van de structuur van de productiecode

“Units”

Test units of behaviour, geen units of structure



Meer informatie

Boek

Test-Driven Development By Example (Kent Beck)

Is TDD dead?

- [TDD is dead. Long live testing. \(DHH\)](#)
- [Is TDD Dead?](#)

Test desiderata

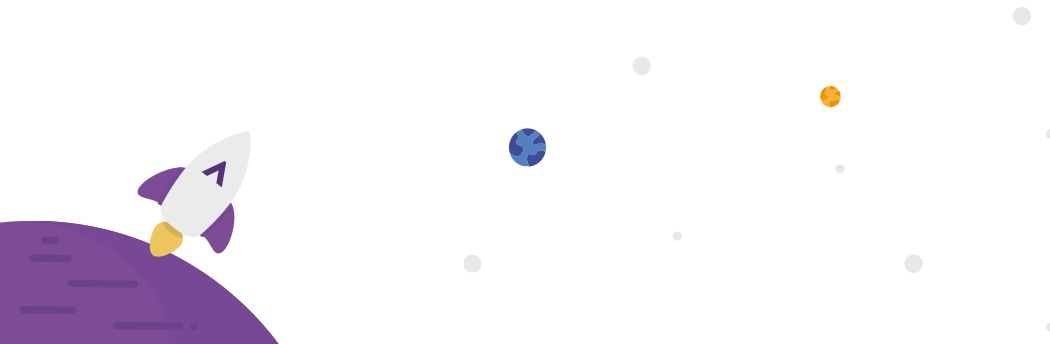
- [Programmer Test Principles. BDD versus TDD. This test tool versus... | by Kent Beck](#)
- [Test Desiderata. Go placidly amid the noise and haste... | by Kent Beck](#)
- [Test Desiderata YouTube Playlist](#)



Nog meer informatie

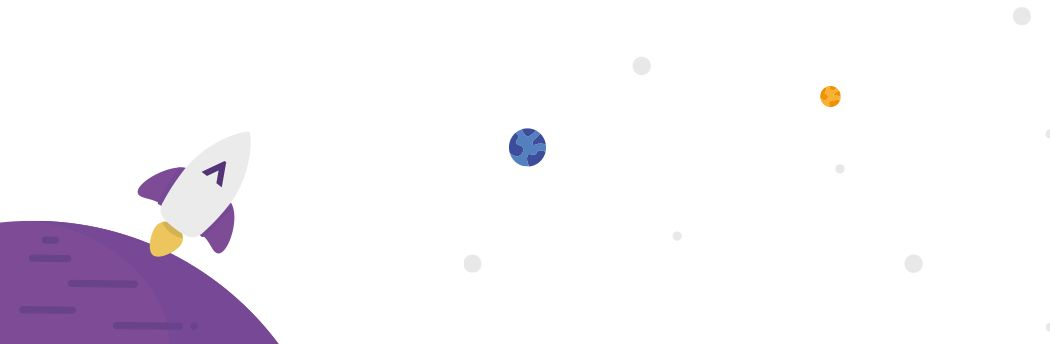
Software design/architectuur

- [Hexagonal architecture \(software\)](#)
- [HumbleObject](#)



Source code

https://github.com/avisi/techday_archives/tree/master/2021-05-27-tdd



Vragen?

Neem gerust contact met mij op: d.groot@avisi.nl

