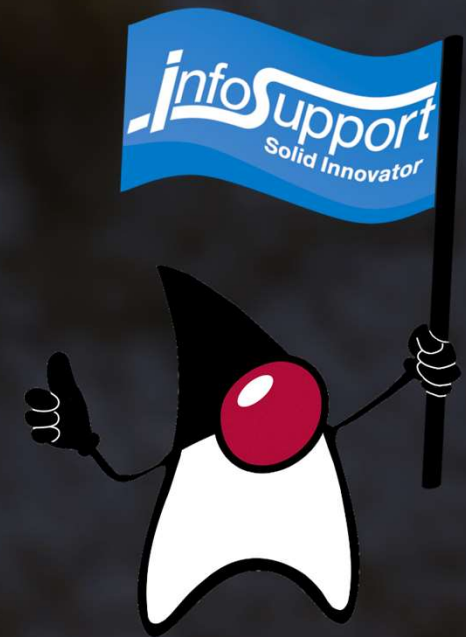


# Modern Testing Patterns

Tom Cools

@TCoolsIT

[www.tomcools.be](http://www.tomcools.be)







A scenic mountain landscape with a wooden fence in the foreground and a dirt path leading towards a hiker in the distance. The sky is blue with some clouds, and the mountains are hazy in the background.

# Tests as a safeguard

Give us **more** certainty that our  
code will work in production.

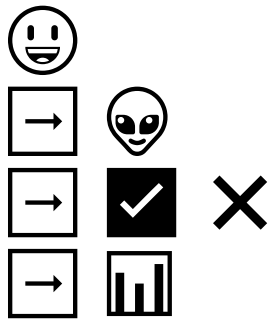




# Mutation Testing



# How it works



1. Source code
2. Mutants
3. Killed/survived
4. Report

```
public boolean bikesAvailable() {  
    return this.getAvailableSlots() >= 1;  
}
```



```
public boolean bikesAvailable() {  
    // Mutant 1: this.getAvailableSlots() > 1  
    // Mutant 2: this.getAvailableSlots() < 1  
    // Mutant 3: true  
    return this.getAvailableSlots() >= 1;  
}
```





[pitest.org](https://pitest.org)



[stryker-mutator.io](https://stryker-mutator.io)





# Improving Database Tests



# H2 In memory with Spring



# H2 In memory with Spring

+ Quick Tests

+ Auto DDL setup

- Auto DDL setup ☹️ -> Assumes Database in production does the same

- Not the real database

# H2 In memory with Spring + Liquibase



# H2 In memory with Spring + Liquibase

- + Quick Tests
- + Database setup is same as production (constraints, FKs, ...)
- Slower startup, because database needs to be constructed
- Not the real database



# TestContainers <3 Spring (and Liquibase)

# TestContainers <3 Spring

- + Very accurate performance
- + Database setup is same as production (constraints, FKs, ...)
- Slower startup, due to liquibase AND testcontainers starting
- Requires external software to be installed (Docker)

# Improving API Tests



# Using TestContainers for APIs



# TestContainers for APIs

- + Very accurate testing
- + Easy to maintain on changes of API
- Availability of API in a Docker container?
- Transient dependencies (API container requires other containers?)

# Wiremock



# Wiremock


- + Easy to setup/configure
- + Recording functionality for updates of API
- + Options to test for network issues (delays/faults)
- Synchronization between API and Wiremock setup?
- Often doesn't include all request details

# Test By Contract





# Contract Testing

- + Encourages up-front design and collaboration
  - + Generates both Tests (Producer) and Stubs (Consumer)
  - + If Consumer Driven: See what consumers you will break
- 
- More complex setup
  - Both Consumer and Producer need to agree to work like this





**Be consciously on the lookout to improve your  
tests with regards to COST/BENEFITS**



[tomcools.be/talks/testing-arnhemjug](http://tomcools.be/talks/testing-arnhemjug)

Tom Cools

@TCoolsIT

[www.tomcools.be](http://www.tomcools.be)

