

Advanced Testing Patterns

Tom Cools
@TCoolsIT
www.tomcools.be



infoSupport
Solid Innovator





Manager immediately
yells at you: “WTF is
going on!?!”

Monitoring is red

You are “on-call”
today

More calls arrive
at helpdesk

Bug caused data
corruption

Broken Systems
Suck



Tests as a safeguard

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

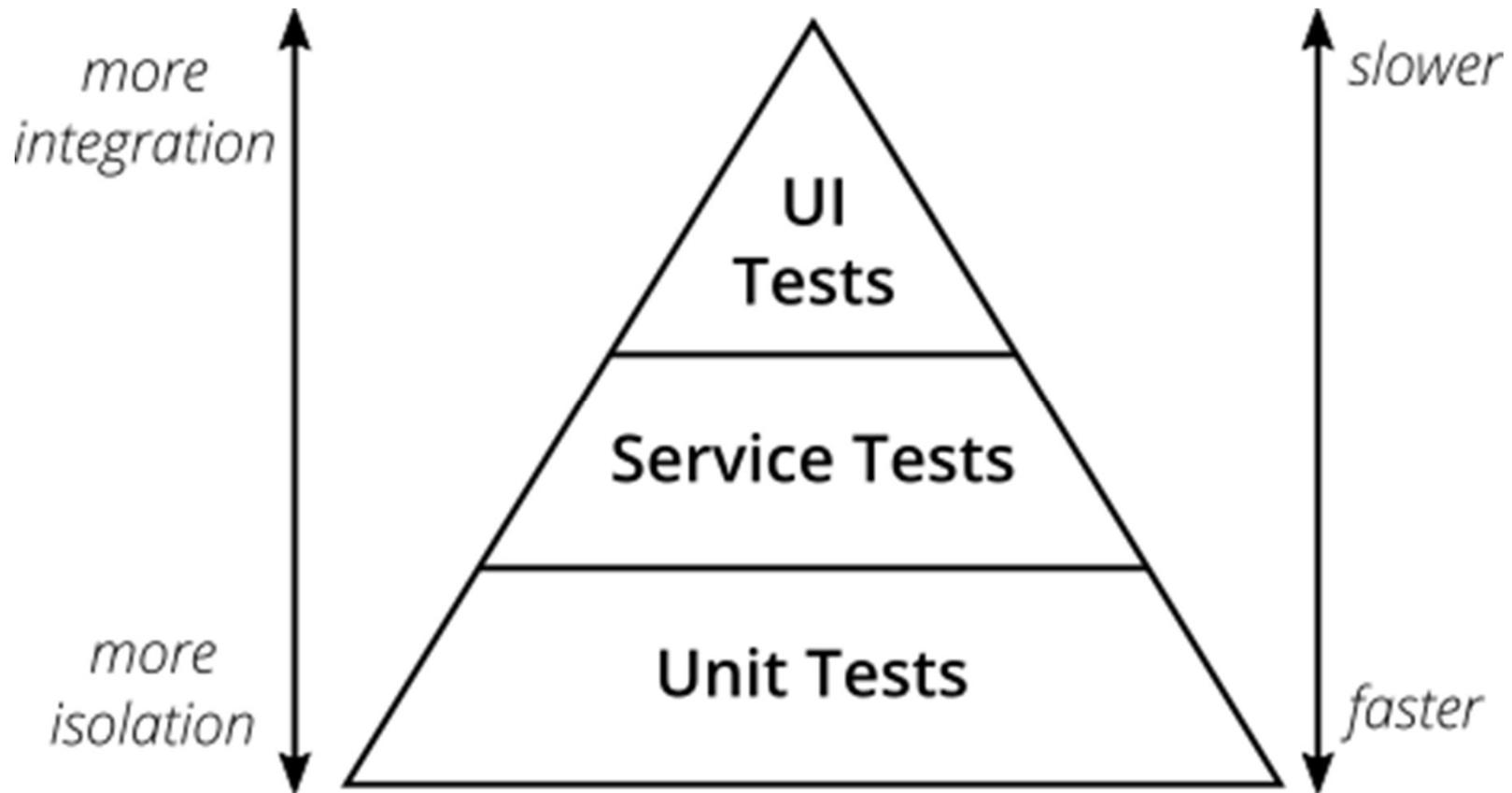


Image taken from: <https://martinfowler.com/articles/practical-test-pyramid.html#UnitTests>



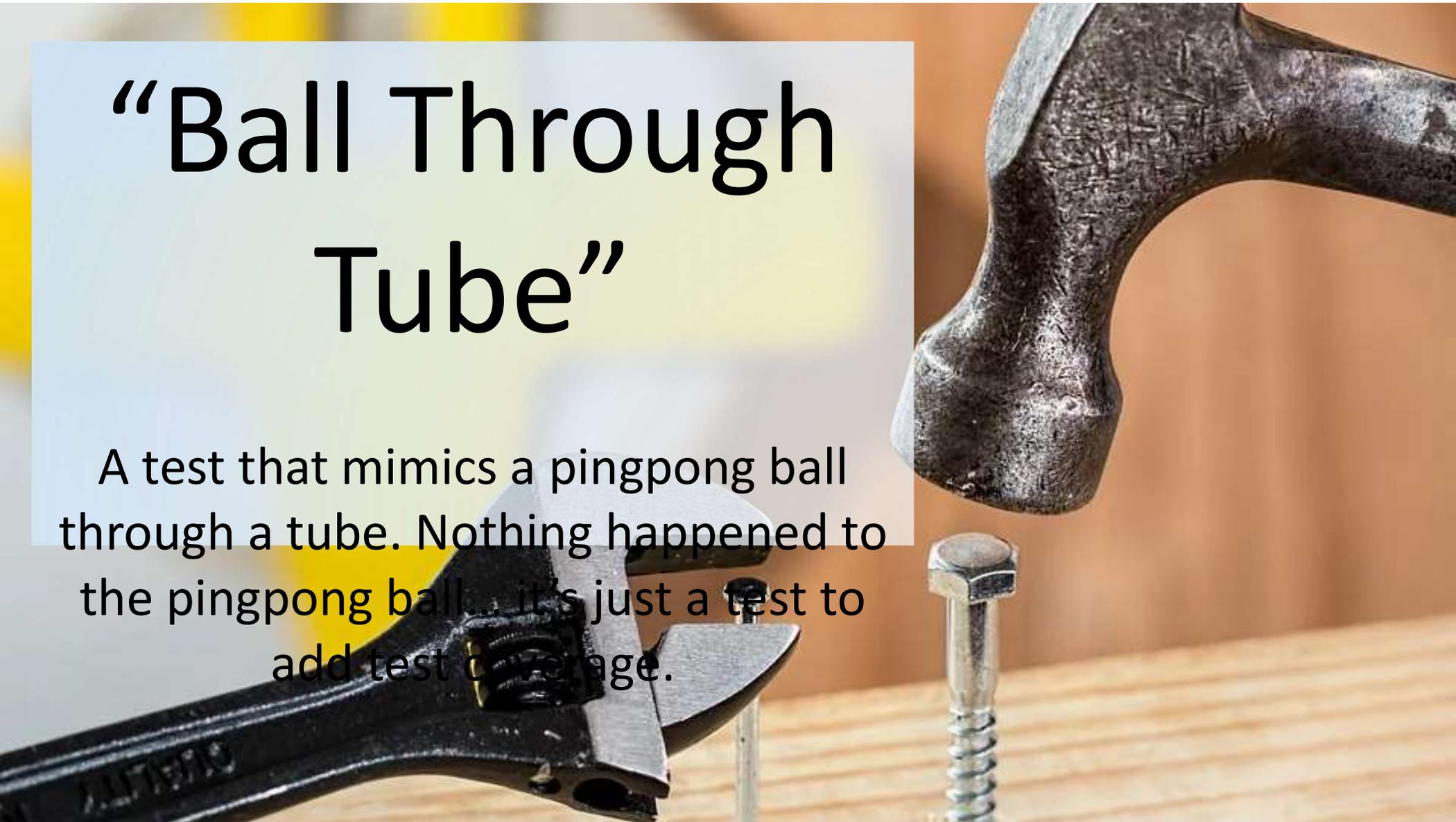
Code Coverage Gatherer

Testing trivial code just
to increase “code coverage”.



“Ball Through Tube”

A test that mimics a pingpong ball through a tube. Nothing happened to the pingpong ball... it's just a test to add test coverage.



Microservices

The microservice architectural style is an approach to developing a single application as **a suite of small services**, each **running in its own process** and communicating with lightweight mechanisms, often an HTTP resource API. These services are **built around business capabilities** and **independently deployable** by fully automated deployment machinery. There is a **bare minimum of centralized management** of these services, which may be written in different programming languages and use different data storage technologies.

- *Martin Fowler & James Lewis*

Tim Peeters (Guest)

+80

DP



ME



Vochten Hans

VM

Van Audenaeren Mats

DG

De Paepe Geert



Vochten Bart



Tim Peeters (Guest)

Improving Database Tests



H2 In memory with Spring



H2 In memory with Spring

- + Quick Tests
- + Auto DLS setup
- Auto DLS 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



Manager immediately
yells at you: “WTF is
going on!?!”

Monitoring is red

You are “on-call”
today

More calls arrive
at helpdesk

Bug caused data
corruption

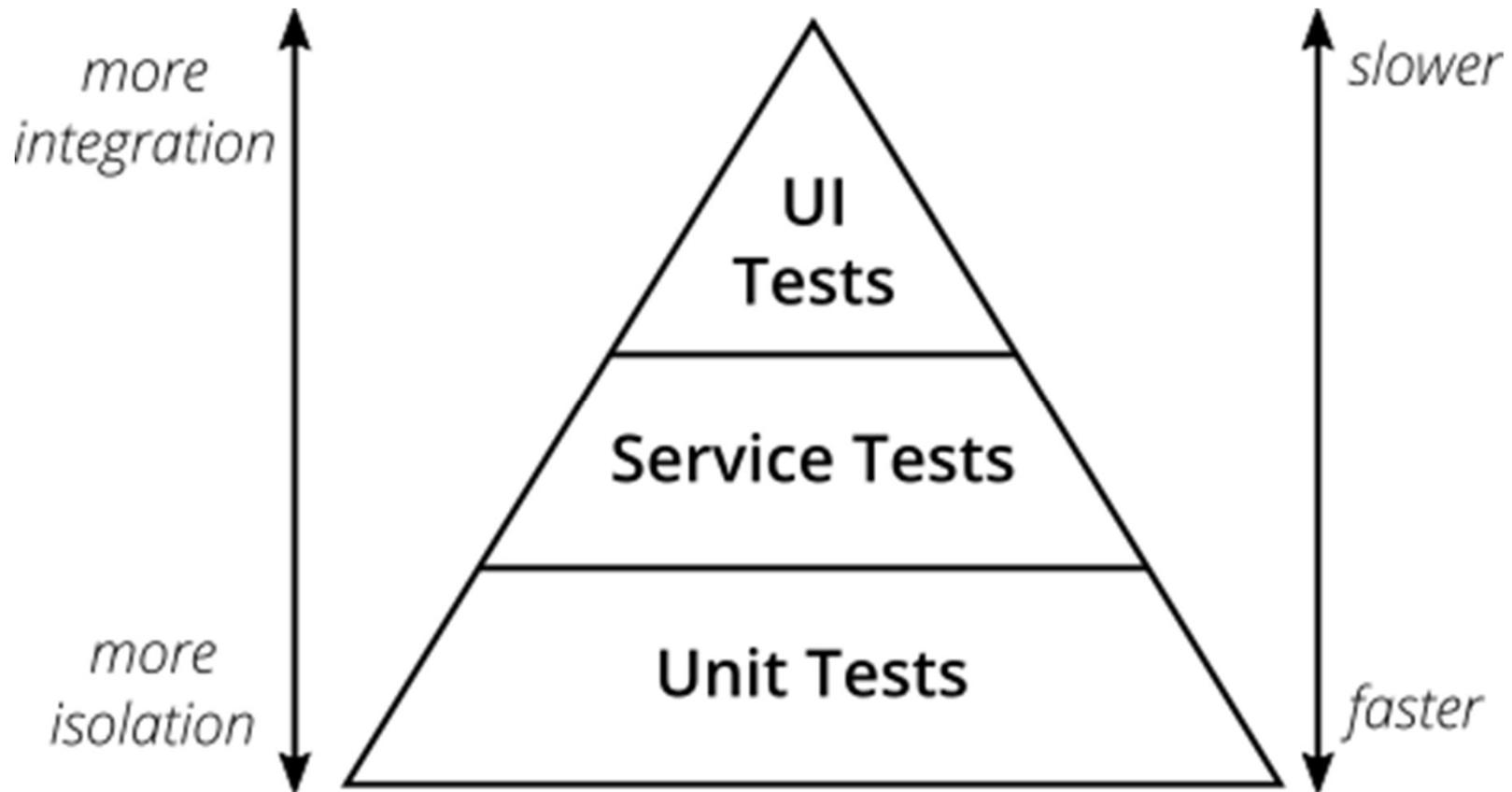
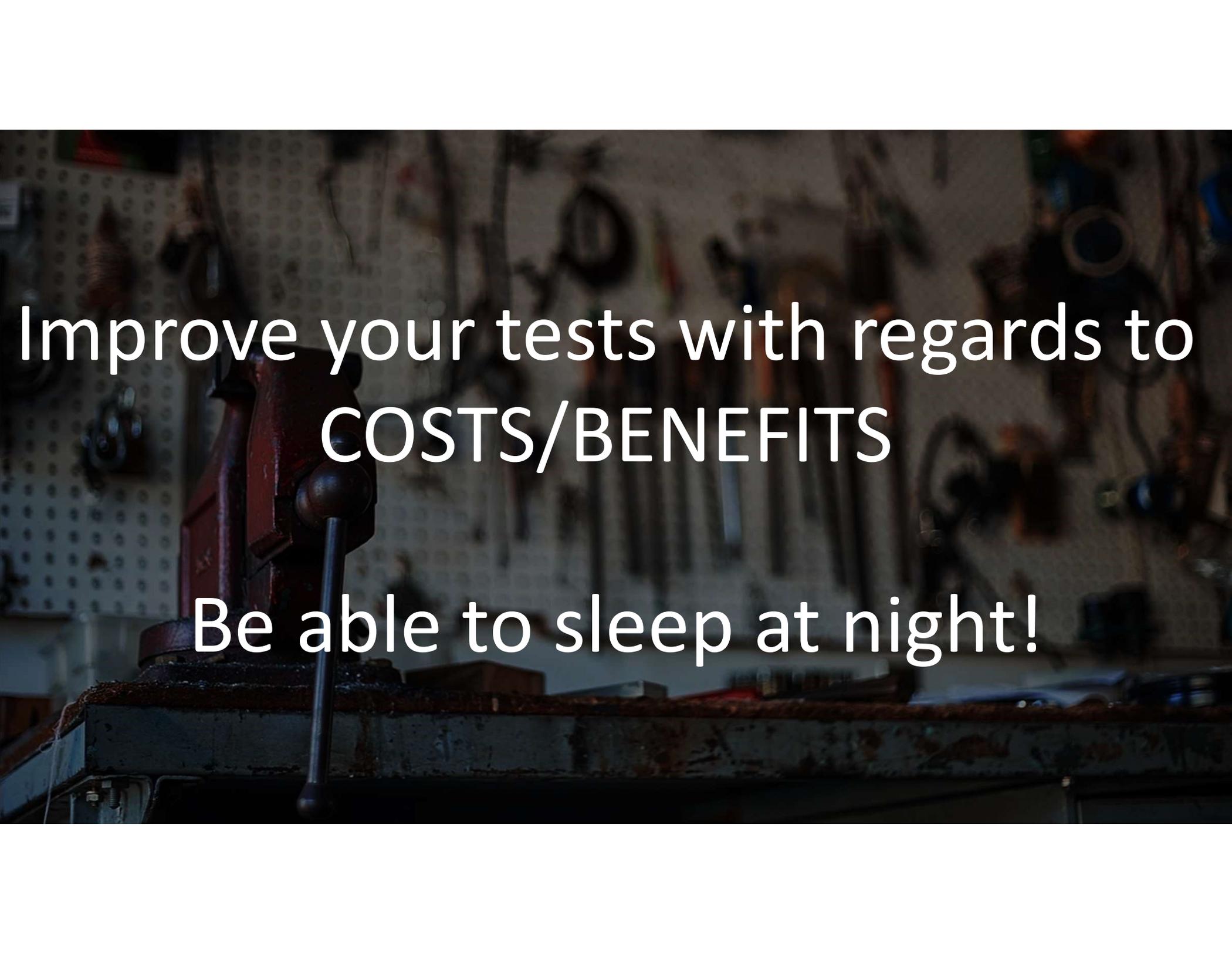
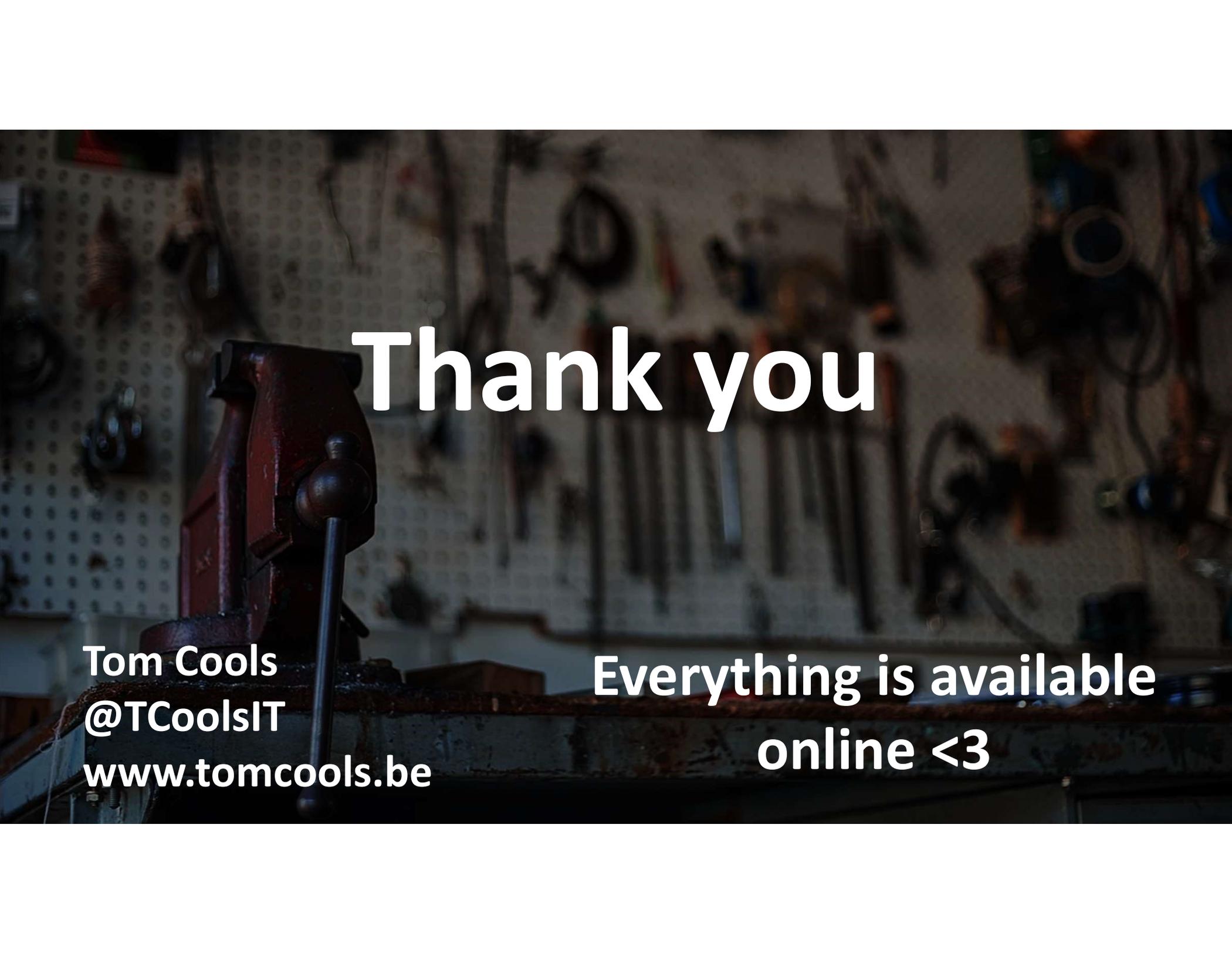


Image taken from: <https://martinfowler.com/articles/practical-test-pyramid.html#UnitTests>



Improve your tests with regards to
COSTS/BENEFITS

Be able to sleep at night!



Thank you

Tom Cools
@TCoolsIT
www.tomcools.be

Everything is available
online <3