



Praktische Einführung in gute Tests

Für Einsteiger – #folienarm

Thomas Much

 @thmuch

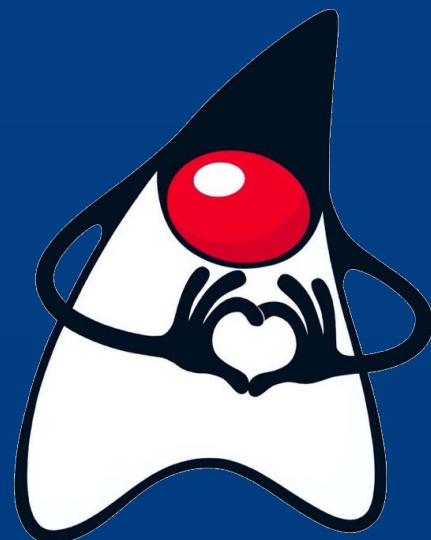
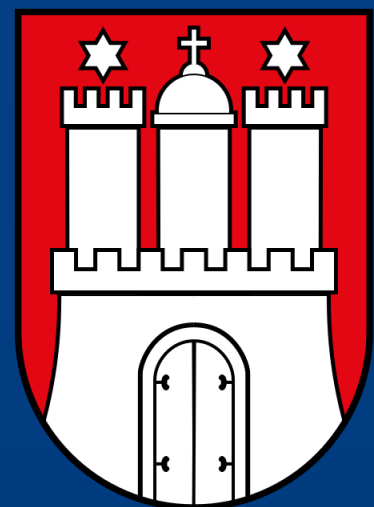
09.11.2021

Softwareentwickler

Technical Agile Coach



 @thmuch



Ein paar Folien zur Einleitung

Live-Coding

Kurzer Abschluss

Q & A

Warum dieser Vortrag?

Neu- Quer- Wieder-

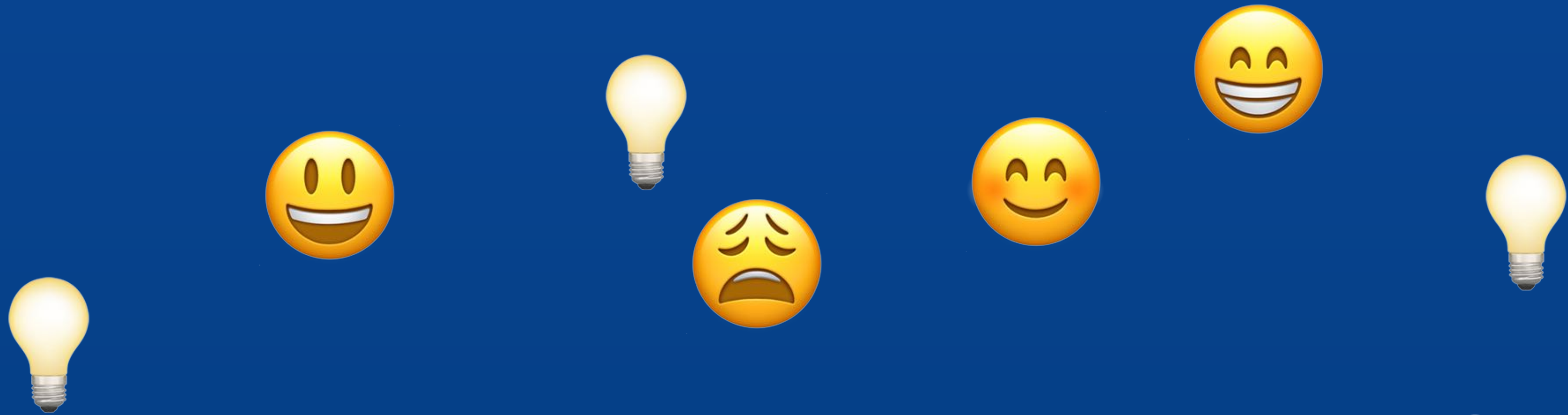
Einsteiger

Copy & Paste

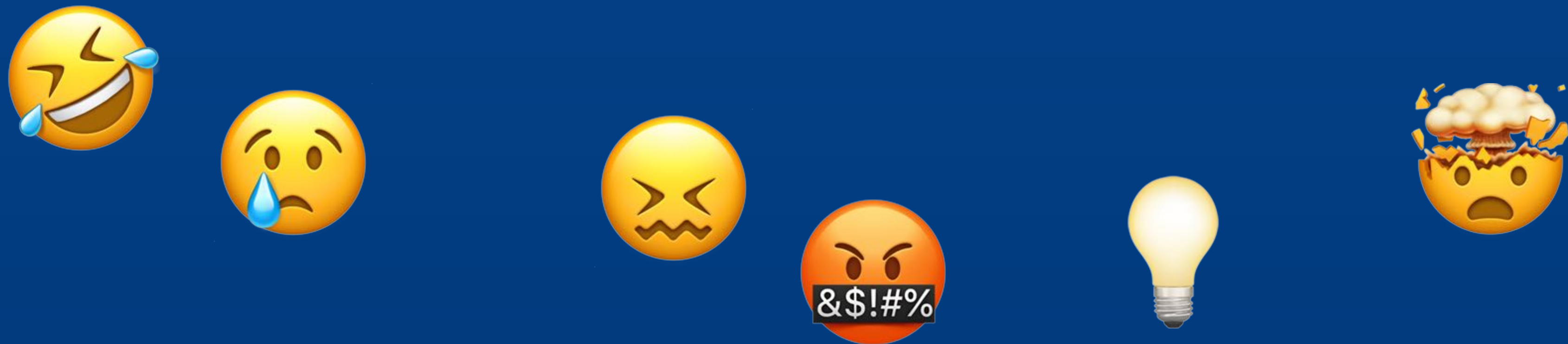
Cargo
Cult

Tradition

Grundlagen. Good Practices.



Was sind „gute Tests“?



Gründe gegen gute Tests

eh nicht
wartbar



eh ständig
kaputt

eh nicht
lesbar

nicht jetzt,
keine Zeit



Nachtesten ist
doof & teuer

Hauptsache
Code-Abdeckung



So viel Qualität
kostet zu viel

Wie testen?

TDD?

Test Driven **Design** & Development



Test First?

Auf jeden Fall nicht „alle Tests vorher“

„Testbegleitend“

Schritt für Schritt

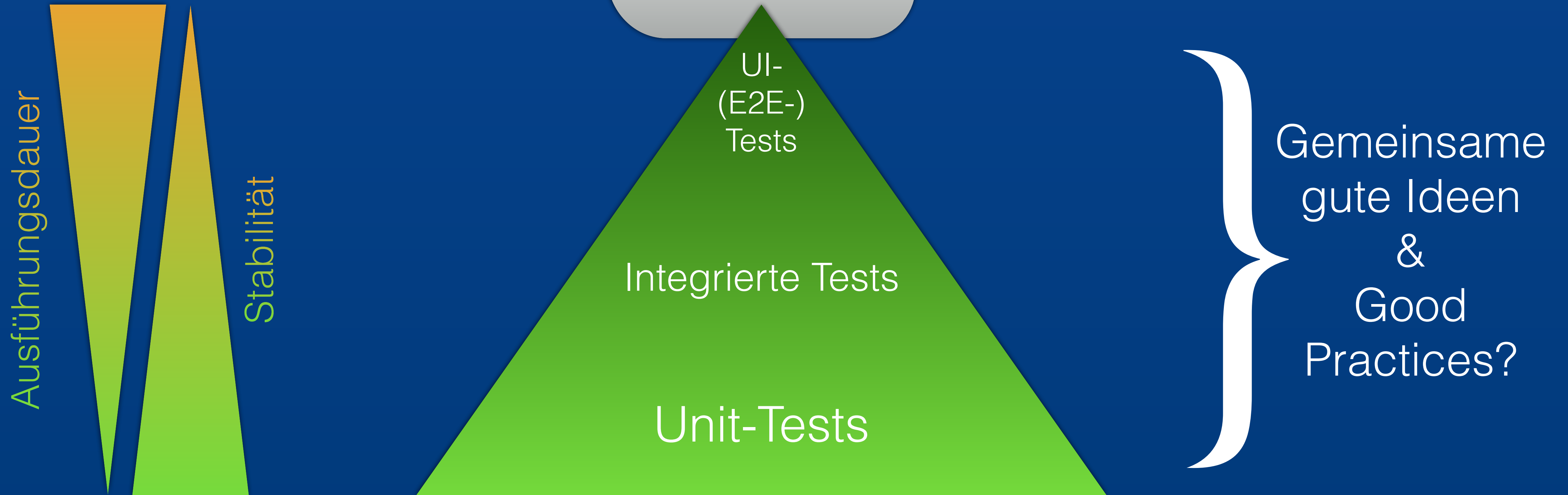
Kleine Experimente!

Wer testet?



Alle?! Lernen von den anderen!

Testpyramide



Anforderung:

Mit dem **Abo-Service** kann ein **Kunde** für ein **Zeitschriften-Produkt** ein Abonnement abschließen.

Der Service liefert die **Abo-Nummer**.
Die Abo-Nummer ist eine **8-stellige Zahl**, die mit der Ziffer „1“ beginnt.

Live-Demo

<https://github.com/thmuch/einfach-gute-tests>

Fazit

Verständliche, wartebare Tests

Jede Testmethode testet **eine Sache**

Given/When/Then bzw. Arrange/Act/Assert hilft dabei

Explizites Setup bevorzugen

Überraschungen durch implizite Magie vermeiden

Namen etc.: Nicht zu lange drüber nachdenken!

Besser **nach und nach verbessern**

„Make it work, make it beautiful, make it fast“

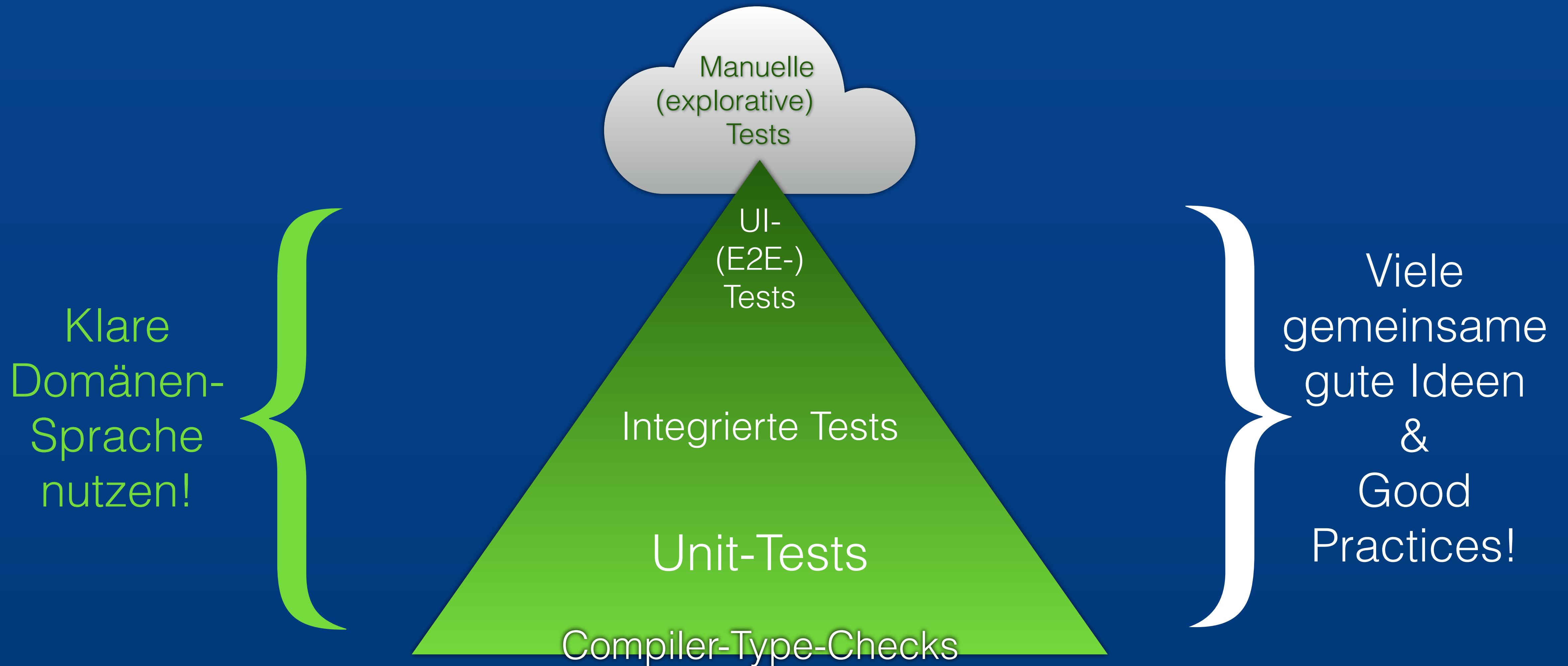


Programming Wisdom
@CodeWisdom

"Make it work, then make it beautiful, then if you really, really have to, make it fast. 90 percent of the time, if you make it beautiful, it will already be fast. So really, just make it beautiful!" – Joe Armstrong

6:15 nachm. · 7. Apr. 2021 · Buffer

Testpyramide (Reprise)



Nicht „blöd“
Gerne **clean!**

Fürs **Lesen**
optimiert

Ohne
„Magie“

Ohne
Überraschungen

Write. Boring. Code.

Auch bei Tests.

Gute Tests



TDD

Testpyramide

Test First

Stabilität

Unit-Tests

Fragen?

JUnit

Benennung

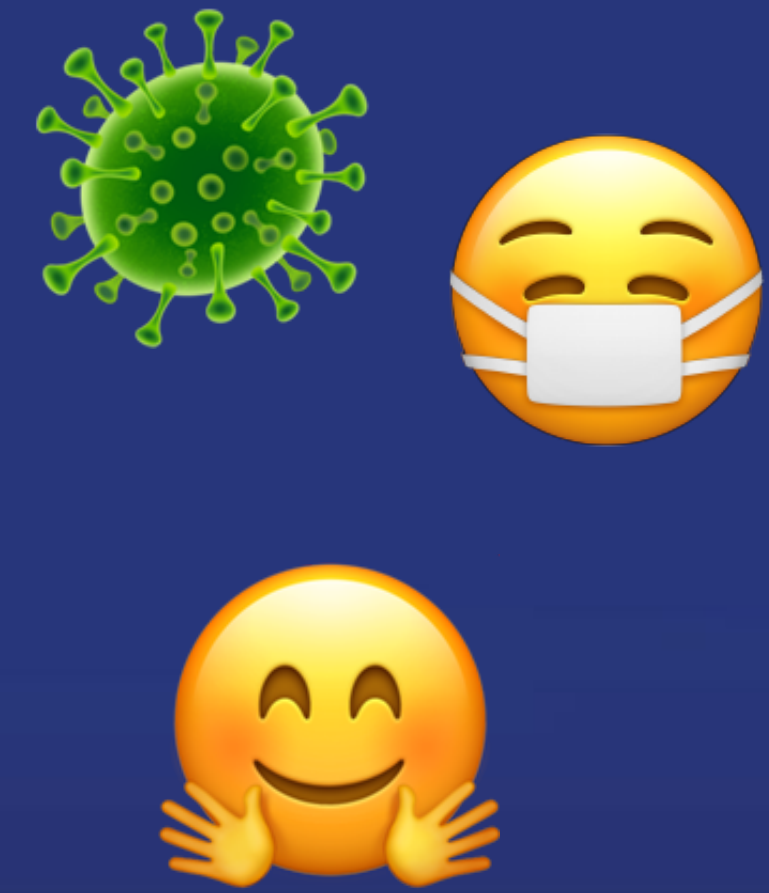
Testdaten

Given-When-Then

Testabdeckung

Lesbarkeit

Wartbarkeit



Vielen Dank!

Thomas Much
 @thmuch

Links & Literatur

- <http://agileinaflash.blogspot.com/2009/02/first.html>
- <https://docs.microsoft.com/en-us/dotnet/core/testing/unit-testing-best-practices>
- Kent Beck, "Test-Driven Development by Example", 2002
- Steve Freeman, Nat Pryce, "Growing Object-Oriented Software, Guided by Tests", 2009
- Michael C. Feathers, "Working Effectively with Legacy Code", 2013
- Emily Bache, "Mocks, Fakes and Stubs", 2013