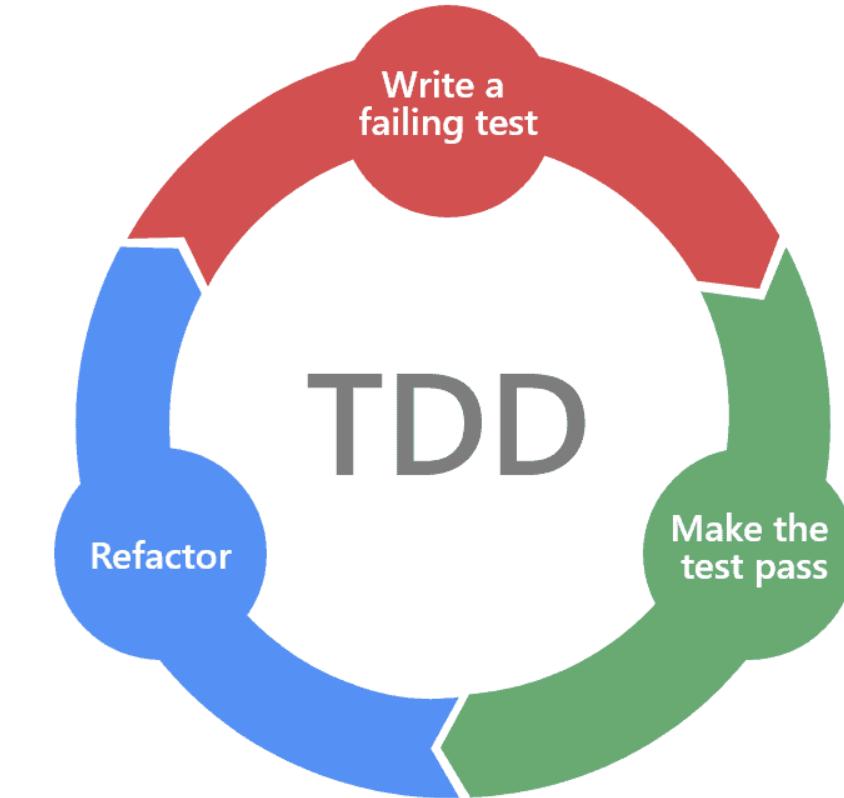
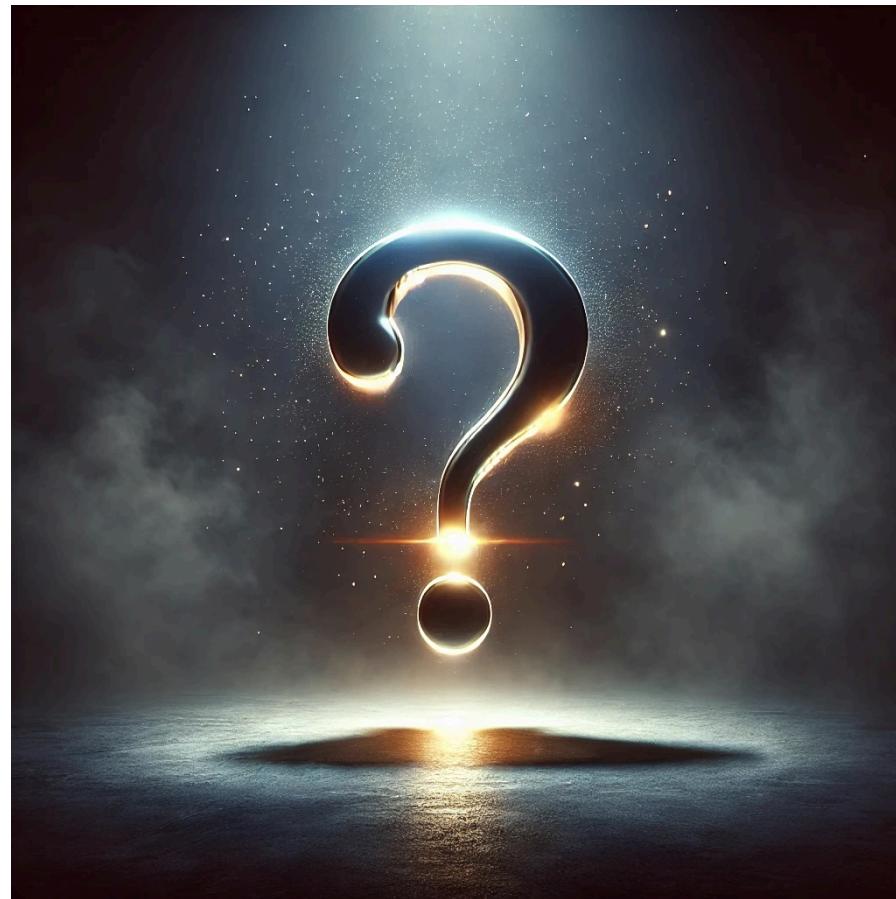


# Test-driven Development (TDD): Warum ist das eine sehr gute Idee ?





**Warum bis du hier ?**

Hast du TDD schon mal  
selbst ausprobiert?

<https://www.menti.com/al9bcx57niwx>





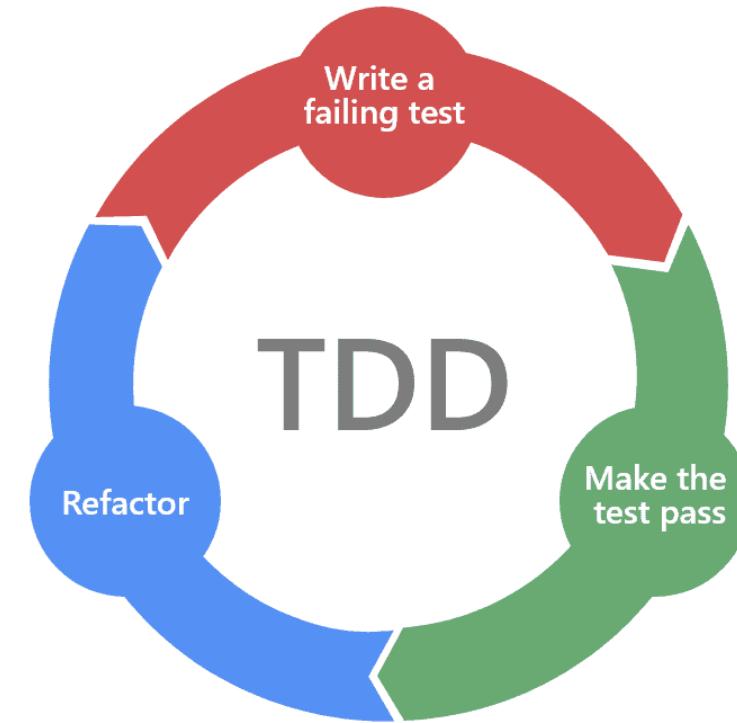
## Wer ich bin:

- Bodo Teichmann
- Softwareentwickler seit mehr als 30 Jahren
- "early adopter" von TDD
- 6 Jahre C Erfahrung, 10 Jahre Java
- CI-Automatisierung, ein wenig Angular , diverse Script Sprachen,
- zuletzt 3 Jahre Spring Boot

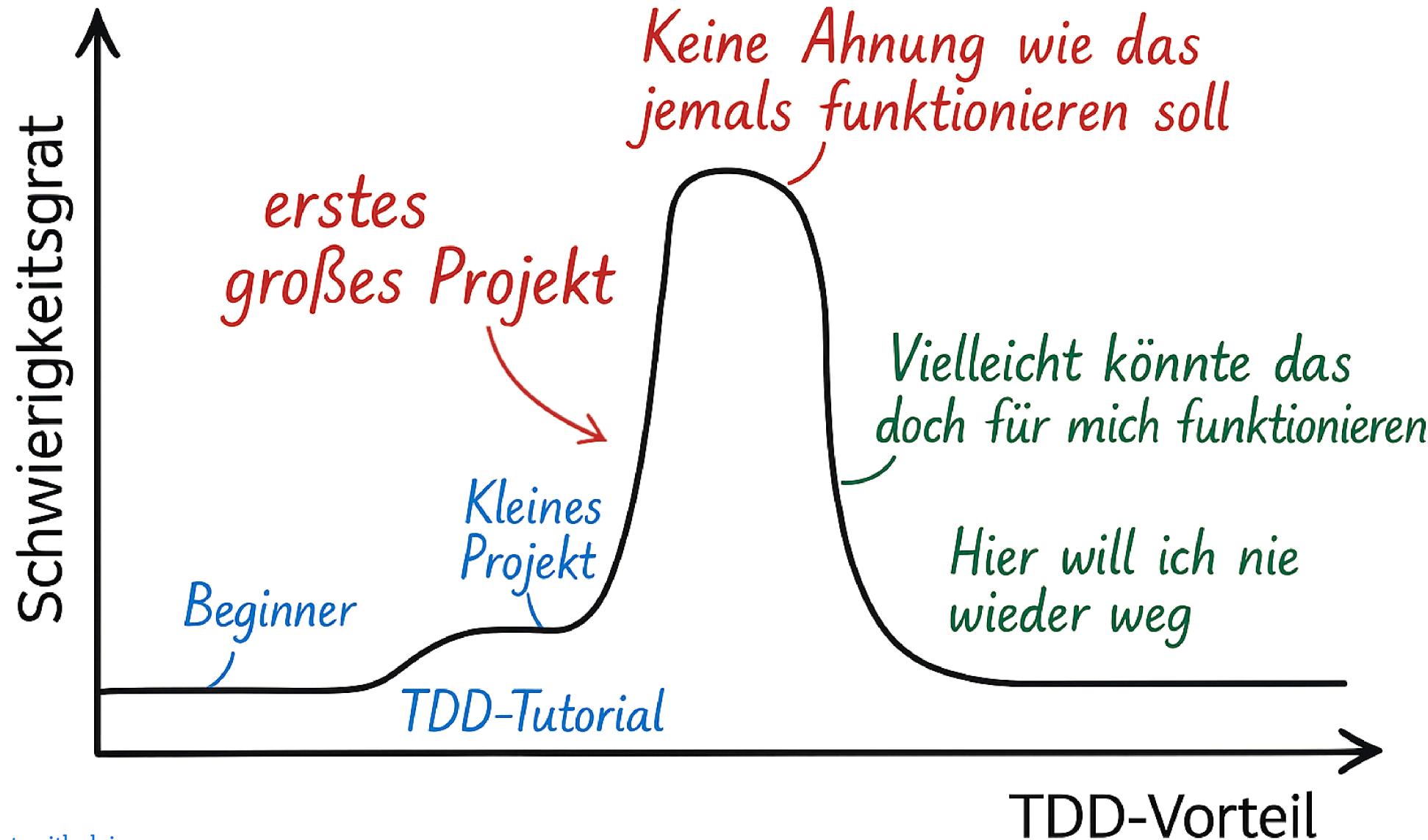
# Was ist TDD ?

TDD = Test-Driven Development

- **Test** First!
- **Driven**: erst Test dann Produktions-Code
- **Tests**: automatisiert, nicht manuelle Tests

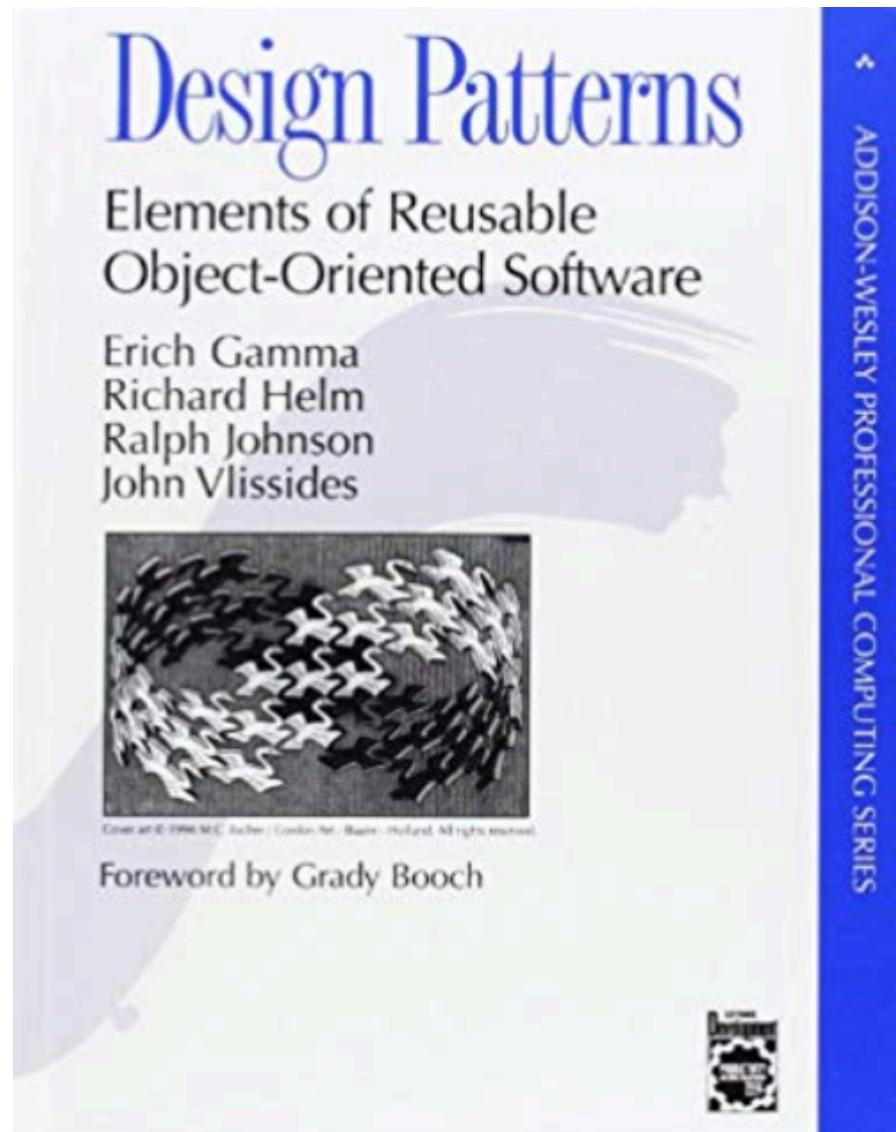








## Vertrauen aufbauen



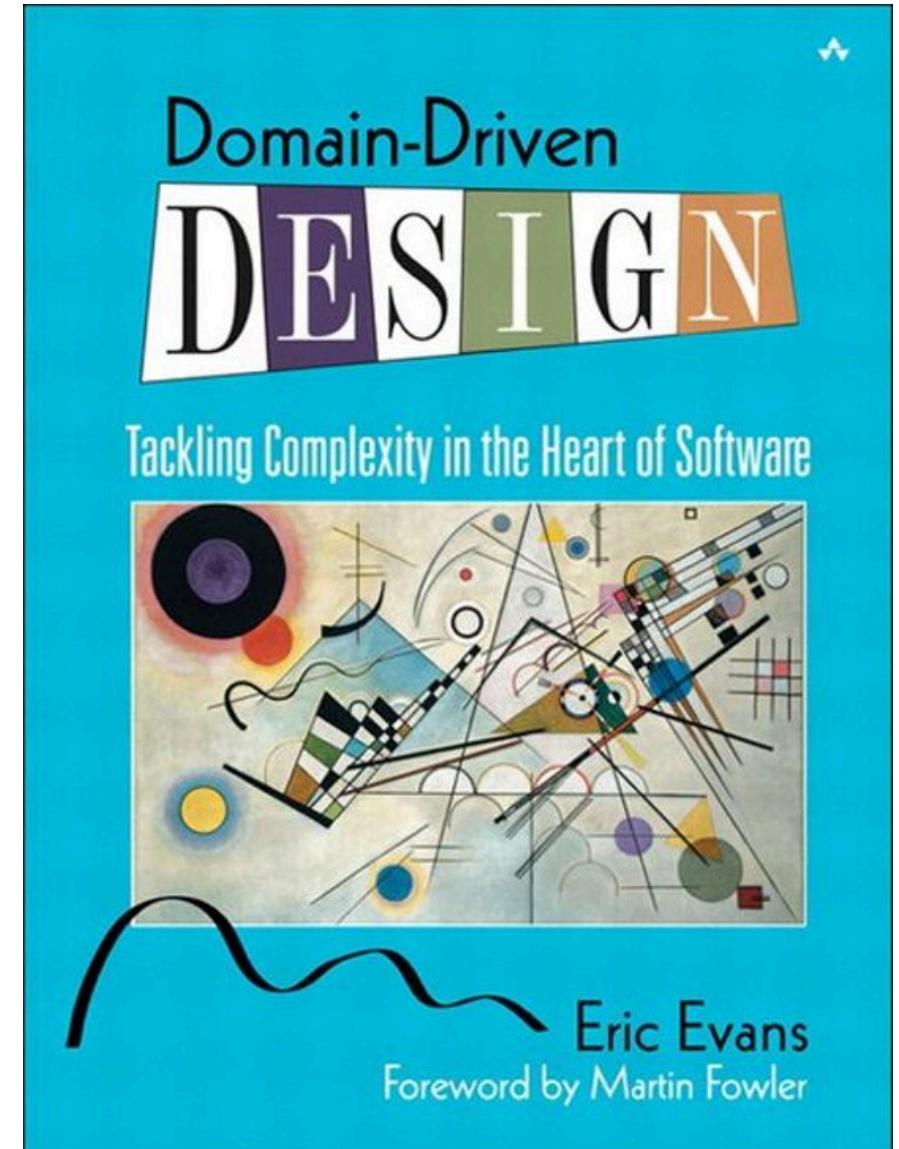
## Warum solltet Ihr jemandem vertrauen...

... der euch empfiehlt, X zu lernen?

nach 32 Jahren AMAZON Verkaufsrang 3  
(Objectorientiertes Softwaredesign )

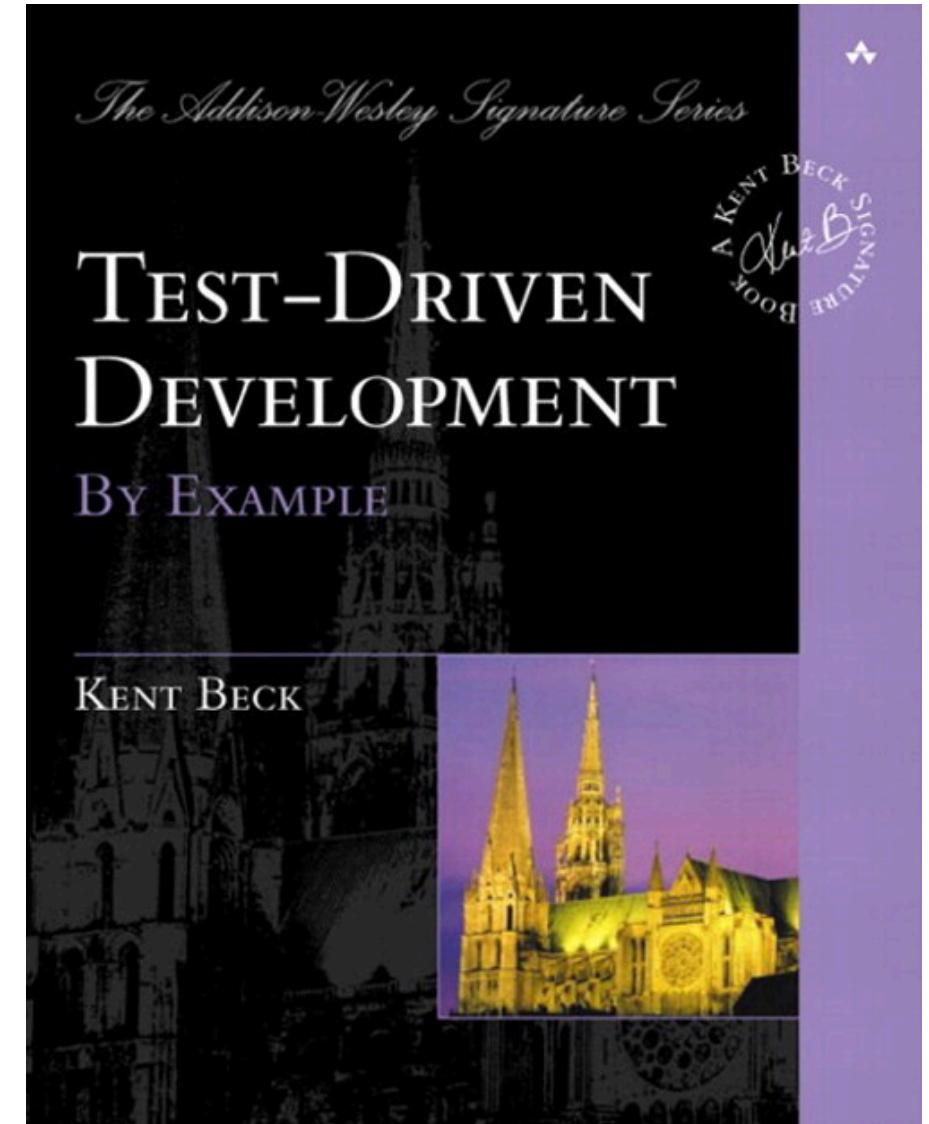
**Warum solltet Ihr ein  
anspruchsvolles Konzept  
lernen...**

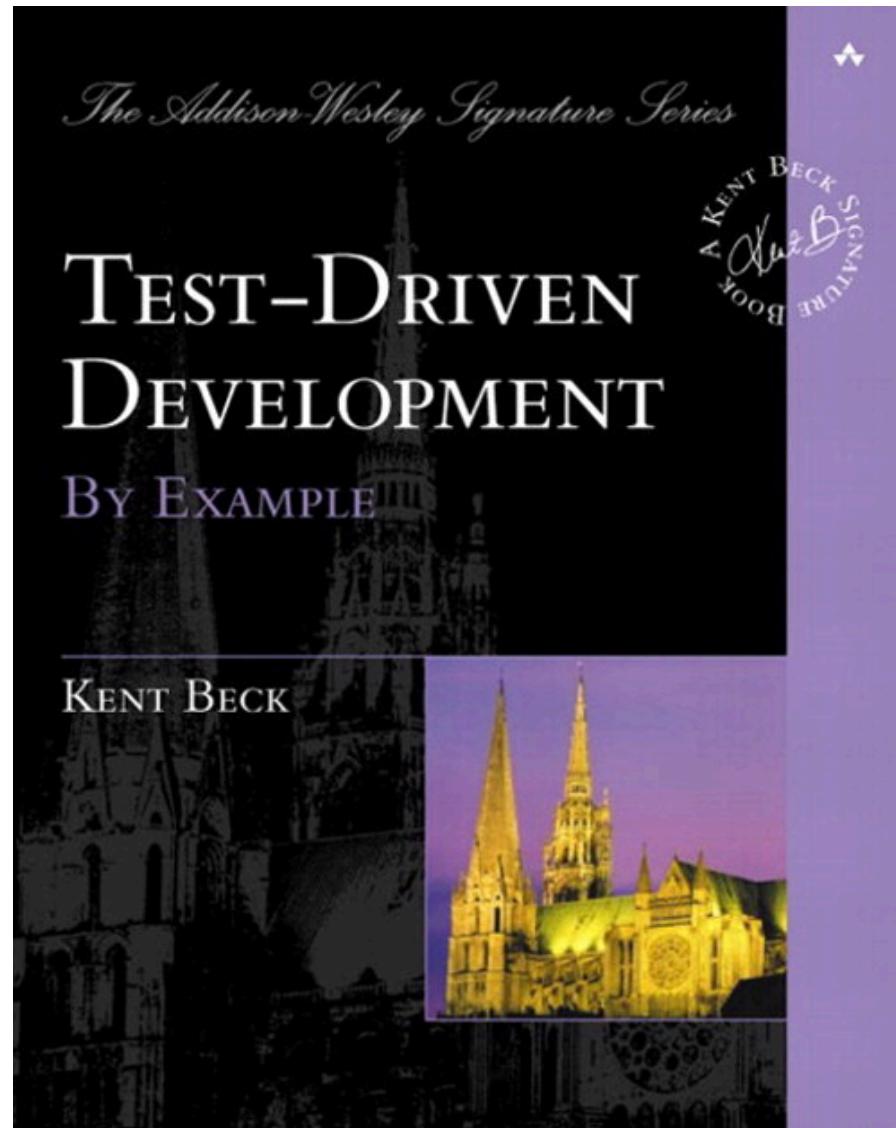
obwohl....



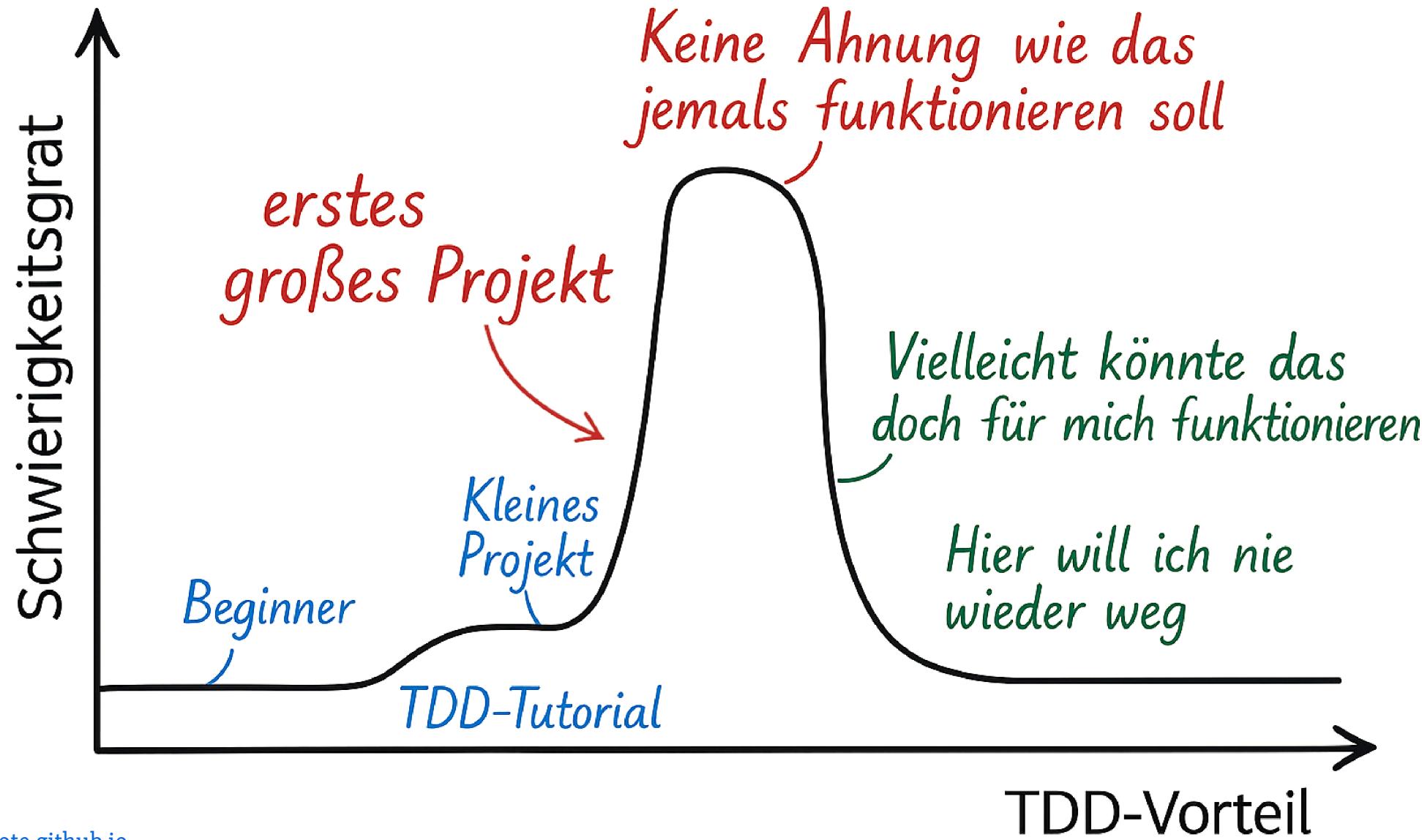
# Test Driven Development

- Grundidee: Test First!
- Also: Wir schreiben automatisch Tests **BEVOR** wir den Produktions-Code schreiben





Buch lesen und fertig ?





## **Reden wir darüber, wie Menschen (und Maschinen) lernen**

... wenn das Lernen ein schnelles  
Feedback gibt

# Reden wir darüber, wie Menschen (und Maschinen) lernen

... wenn das Lernen **KEIN** schnelles Feedback gibt  
z.B.

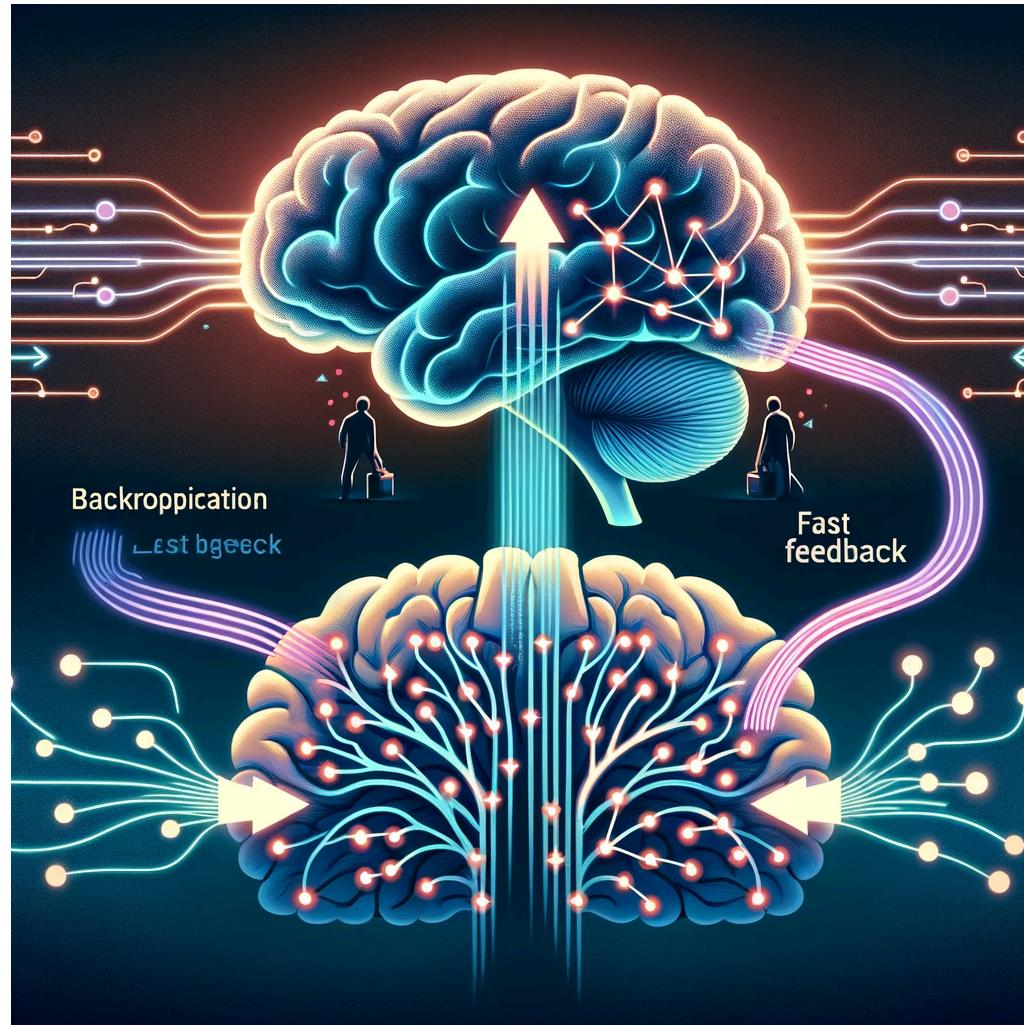
- schreiben lernen
- eine Fremdsprache lernen



## FRAGE:

Wer von euch hat etwas schwieriges  
mühsam gelernt und erst viel später den  
Nutzen verstanden ?





# Wie lernt Mensch (oder eine Maschine) trotzdem ohne schnelles Feedback?

- Maschine: historische Daten sammeln
- Mensch: historische Daten sammeln (von Menschen vor euch, die die Erfahrung gemacht haben)



Wie lernt Mensch (oder  
eine Maschine) trotzdem  
ohne schnelles Feedback?

# Warum ist TDD so (scheinbar) schwierig zu lernen?

-> kein schnelles Feedback!

und

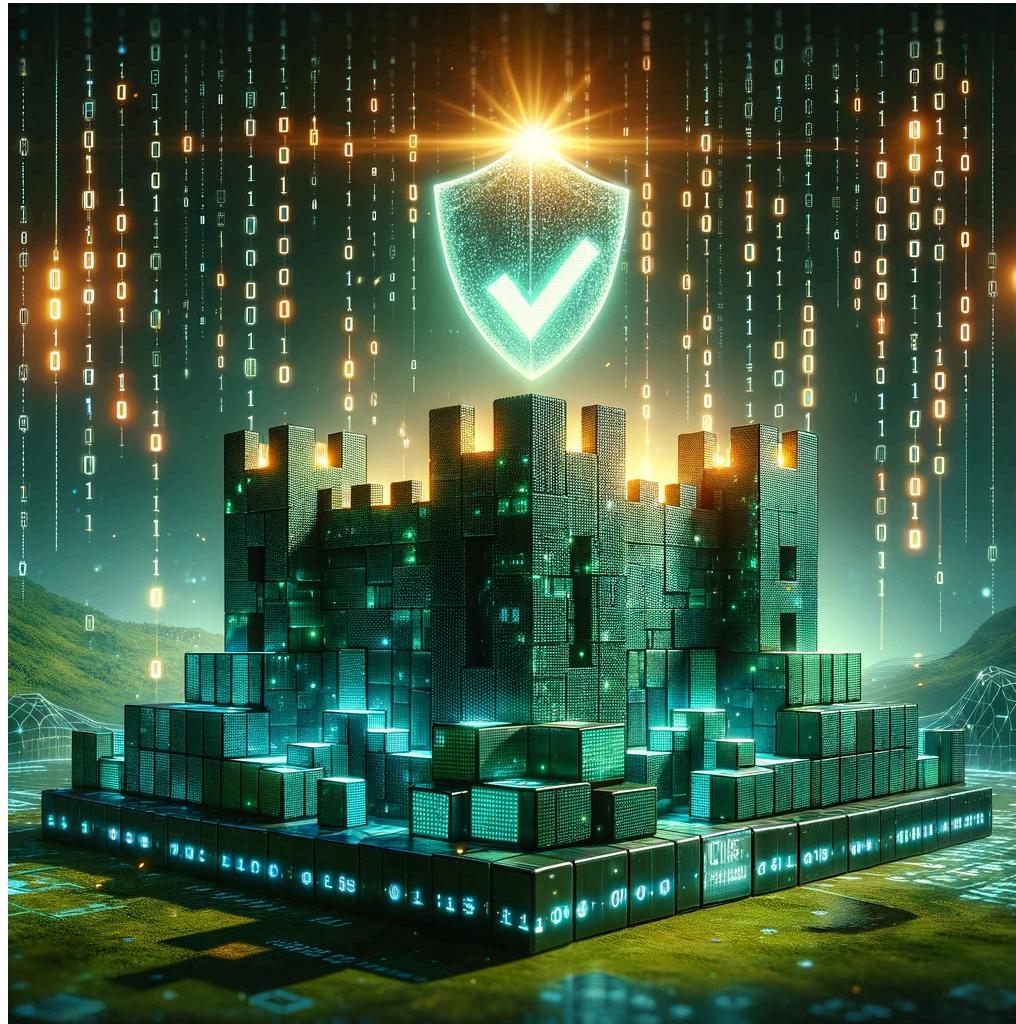
- das Thema hat viele Aspekte.
- Nimm dir Zeit dich dem Thema anzunähern
- Rückschläge sind zu erwarten
- Der Weg ist das Ziel

# Wer kam schon mal zu einem schon länger bestehenden großen Softwareprojekt dazu, das nicht nur...

- vorbildliche Testabdeckung hatte,
- also sehr stabil im Betrieb war,
- sondern auch durchdachte leicht verständliche Architektur hatte
- die auch noch leicht erweiterbar war ?
- [Mentimeter](#)

<https://www.menti.com/al9bcx57niwx>





## Was soll TDD (angeblich) leisten ?

- offensichtlich Stabilität und Qualität
- ... und weniger offensichtlich ? ...

# TDD -> fast automatisch bessere Architektur

- ... durch die der Code besser verständlich ist,
- ... besser wartbar bleibt
- ... und sich leichter an neue Anforderungen anpassen lässt





## TDD ist NICHT Unittests !

... sondern eher das Gegenteil,  
wenn man nach der Definition  
von "Unittests" in Wikipedia  
geht !



## Wie geht TDD ?

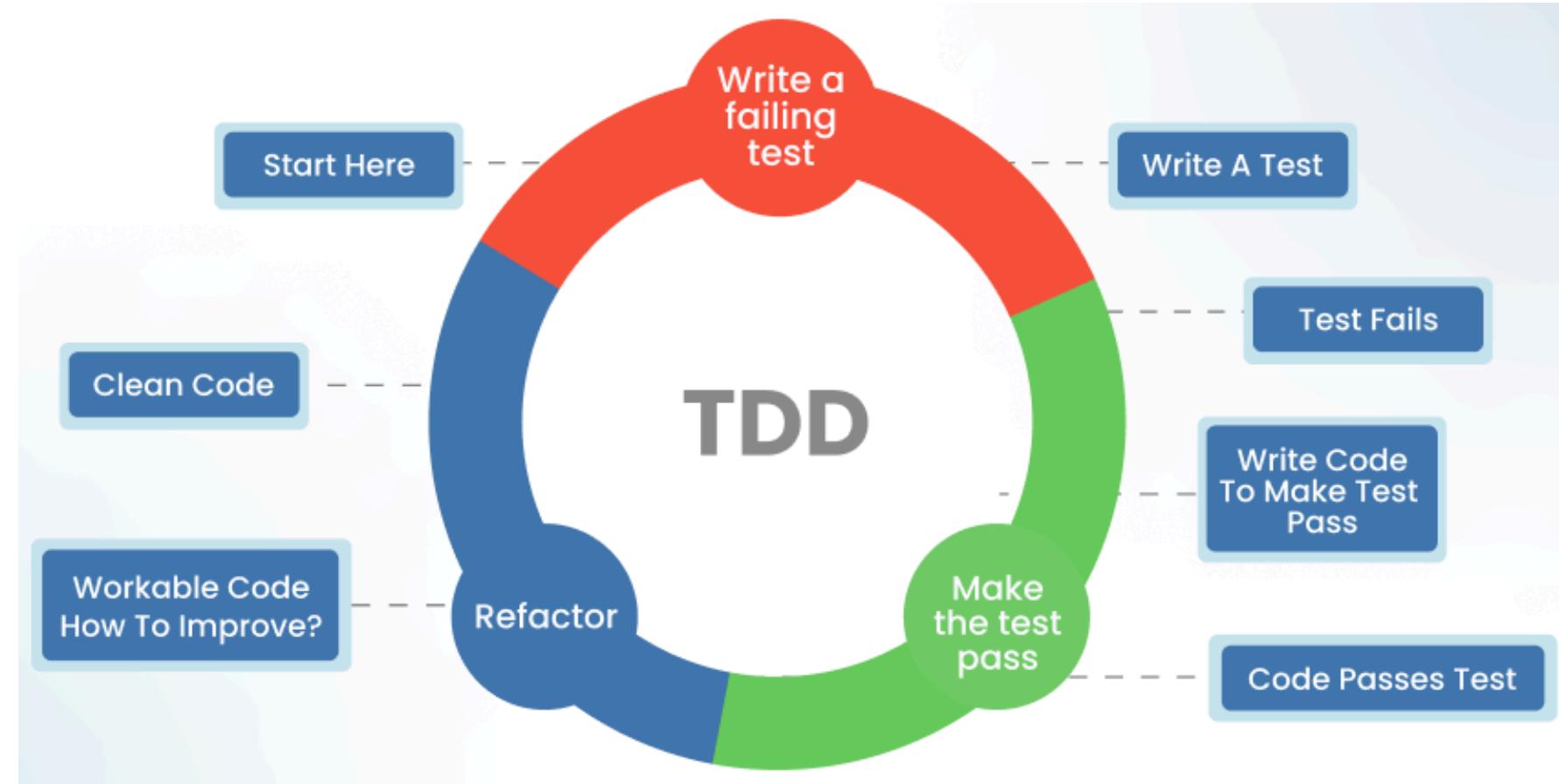
- TDD ist **NICHT** Testklasse:Implementationsklasse 1:1 -> führt zu viele Problemen !
- TDD ist **NICHT GLEICH** Unitests !
- Junit ist **nur** ein gutes Werkzeug und definiert nicht die Methode mit der man vorgeht!

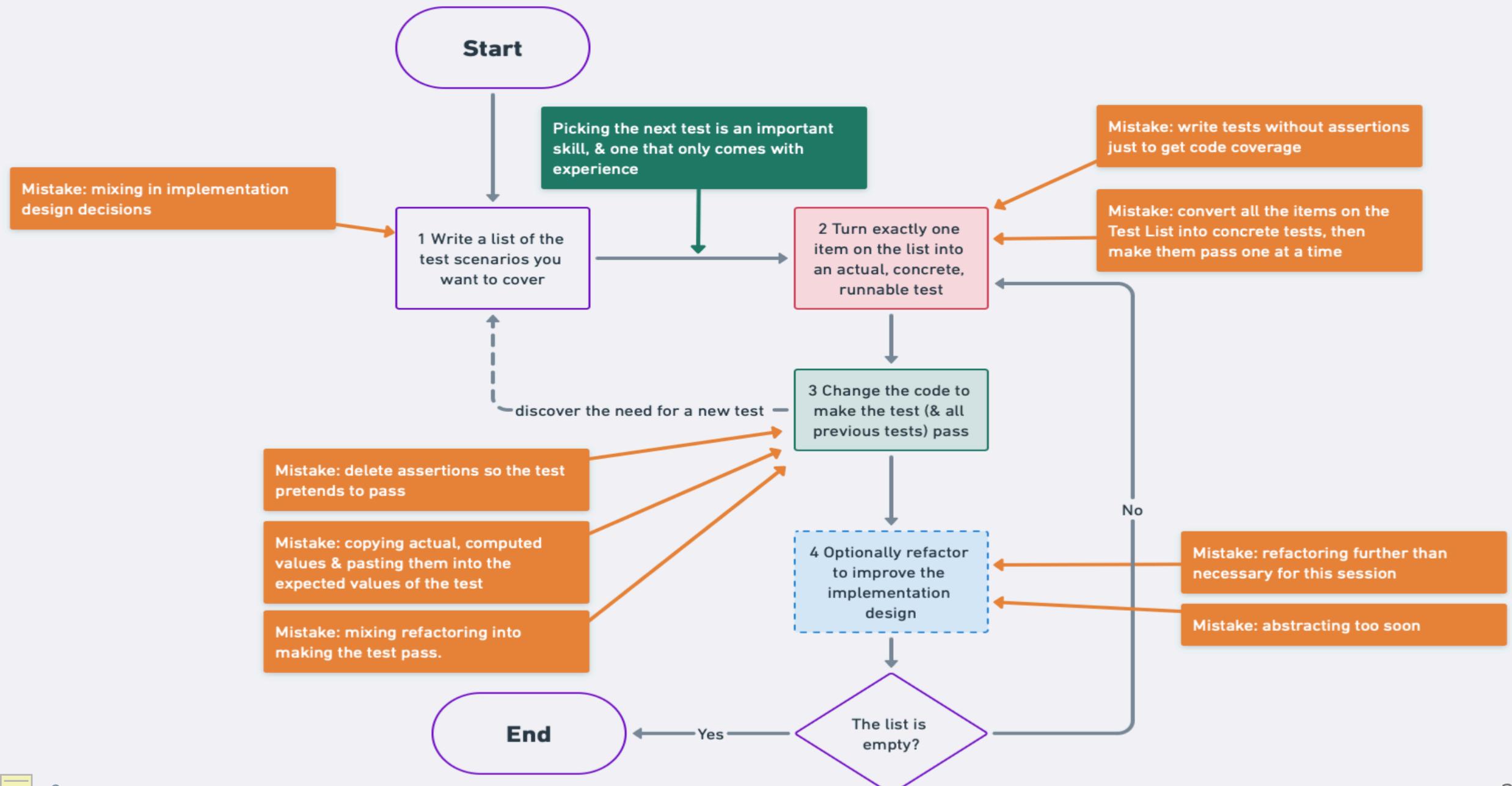
## "JUnit" als Tool auch für TDD hilfreich

... "Unittest" als Begriff wird oft auch abseits der Definition von Wikipedia verwendet, z.B: Unit-Test-tool "xUnit", z.B. "JUnit"

# Wie geht TDD ?

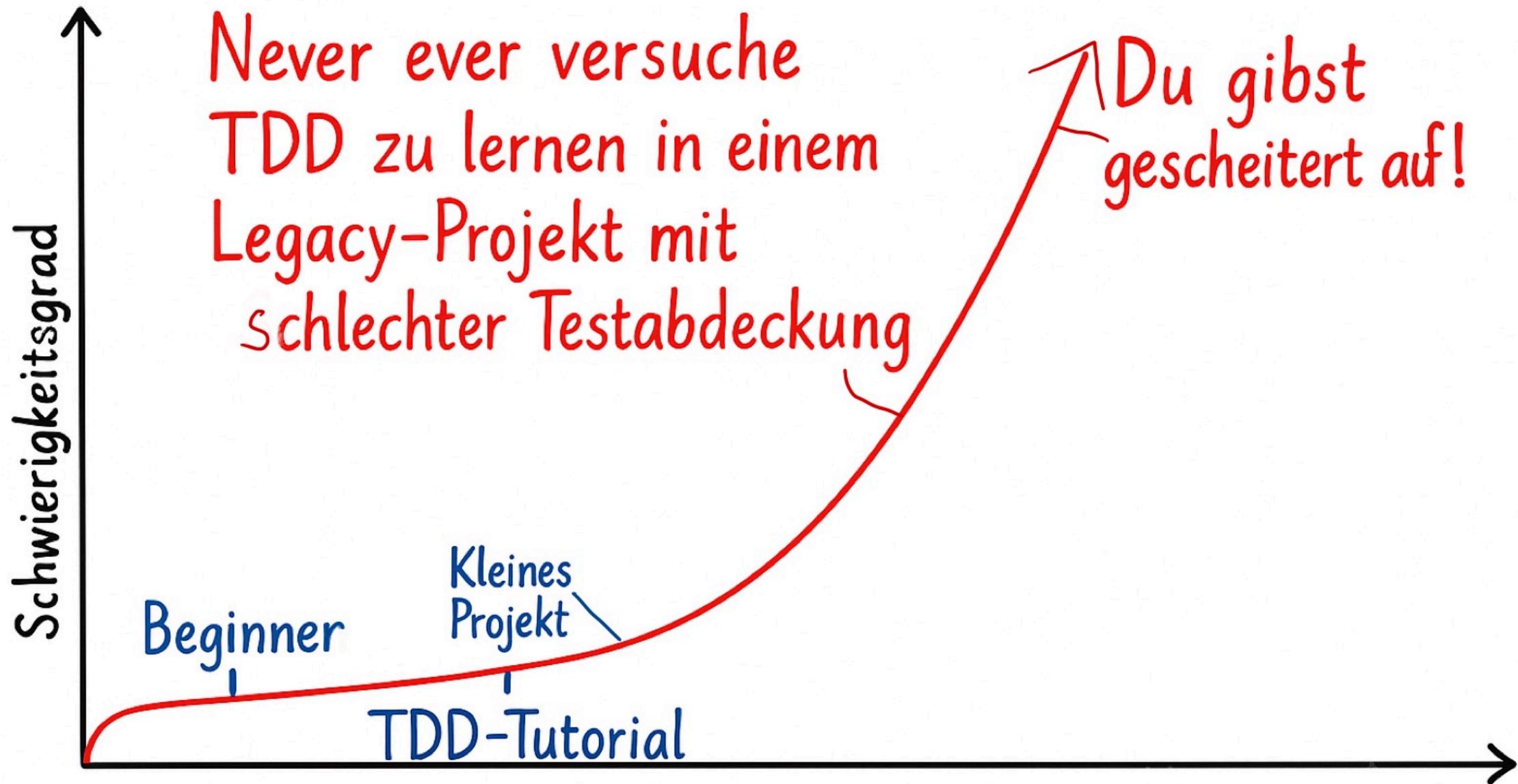
... ganz einfach  
(wirklich ?)







**Wann passt TDD nicht ?**





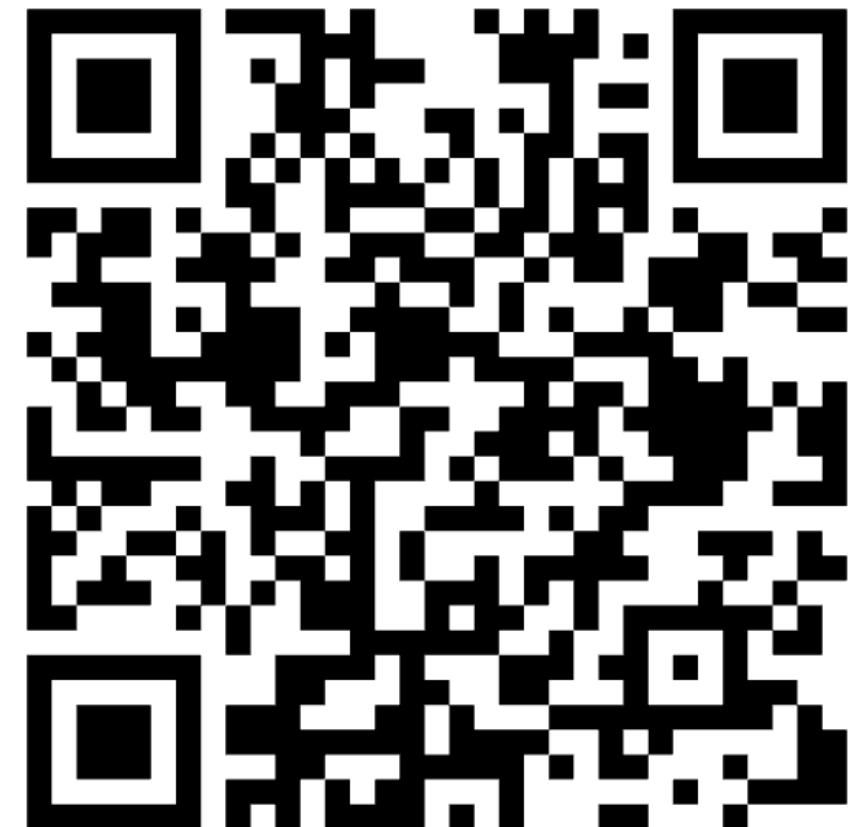
- TDD ist aber nicht so einfach zu lernen, wie es aussieht
- du kannst auch viel falsch machen.
- TDD macht große Versprechungen



# Beweise, Beweise !

# TDD führt (automatisch) zu besserer Softwarearchitektur ?

- wirklich ?
- Ja, siehe ausführlich unter  
[bodote.github.io](https://bodote.github.io): TDD Blog Teil 2:  
Architektur



## Hier nur ein Auszug

... alle Links findest du auf  
[bodote.github.io](https://bodote.github.io): TDD Blog Teil 3: Beweise  
etc.



# Studien über TDD

- "The Spring team advocates test-driven development, TDD ."

"Das Spring Team unterstützt und befürwortet TestDrivenDevelopment.

- VMware Pivotal Labs Website

"Wir praktizieren selbst TestDrivenDeveopment. Da heißt, bevor wir die Arbeit an einem neuen Feature beginnen, schreiben wir zuerst einen Test, der das gewünschte Verhalten genau beschreibt."

- IBM: Assessing test-driven development

50% weniger Bugs als bei "test last", [bei zunächst] minimaler geringere Produktivität

## Studien über TDD (2)

- Microsoft: Evaluating the Efficacy of Test-Driven Development: Industrial Case Studies
  - | Bugrate sinkt mit TDD um 62%-77% , [initial] 15% -35% mehr Zeit für TDD notwendig.
- Metastudie, die 6 unterschiedliche andere Studien ausgewertet hat
  - | Ergebnis: deutlich Positiv, auch bezüglich Software-Architektur
- An Experimental Evaluation of the Effectiveness and Efficiency of the Test Driven Development
  - | deutlich bessere Gesamt-Produktivität und Codequalität

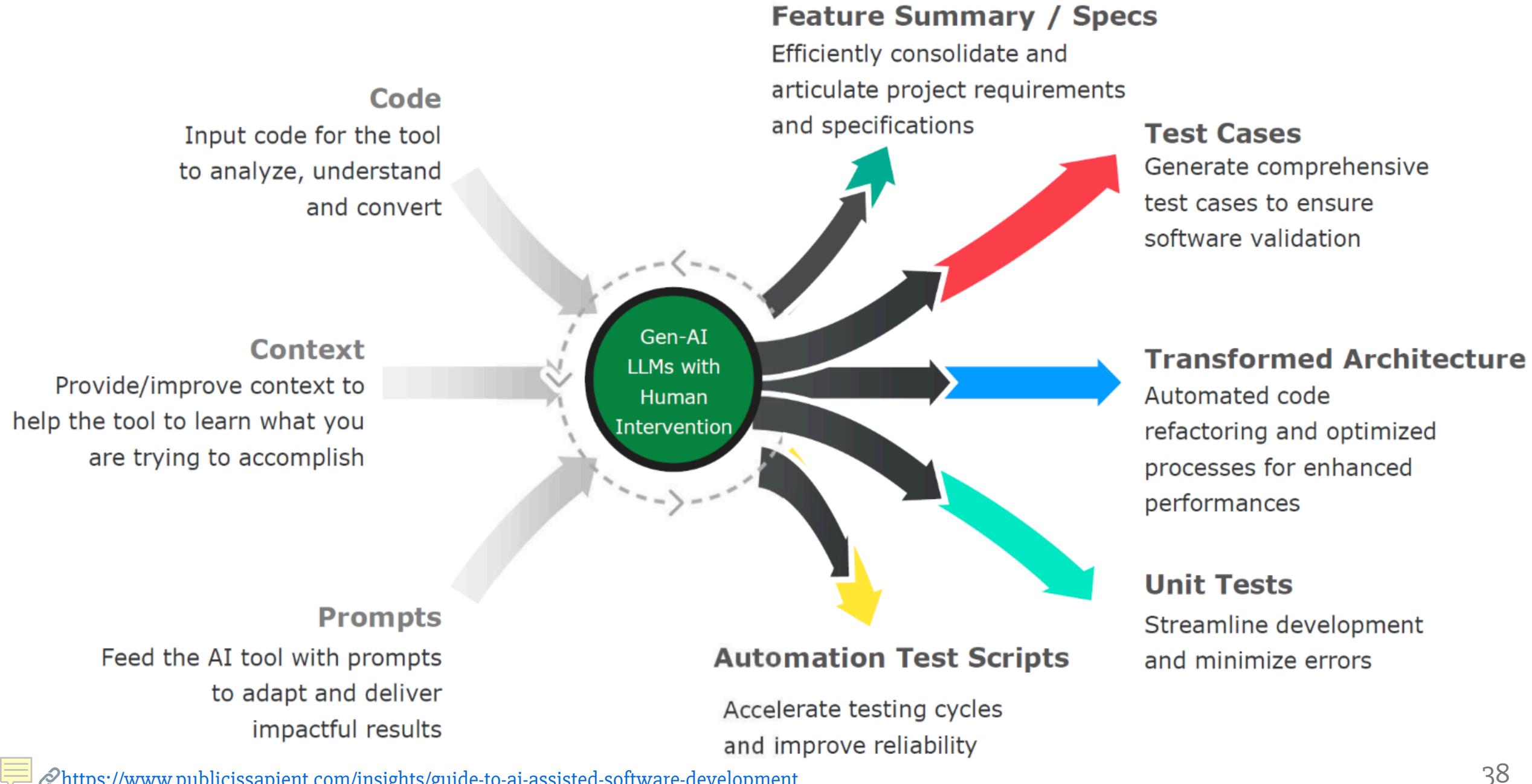
# Studien über TDD (3)

- noch eine Metastudie
  - Gesamturteil deutlich positiv, aber Hinweis: "schwer zu meistern" , **Fazit: Empfehlung für TDD**
- Empirische Studie zu Test-Driven Development
  - " *Test First* ist besser darin, **lose gekoppelte** Softwarekomponenten hervorzubringen als *test last*." *Lose gekoppelt* bewirkt zum Beispiel bessere Wartbarkeit, Verständlichkeit für die Entwickler, ist also was sehr gutes.



## TDD und AI-assisted coding

Ist TDD in Zeiten von KI-unterstützter Softwareentwicklung noch sinnvoll ?

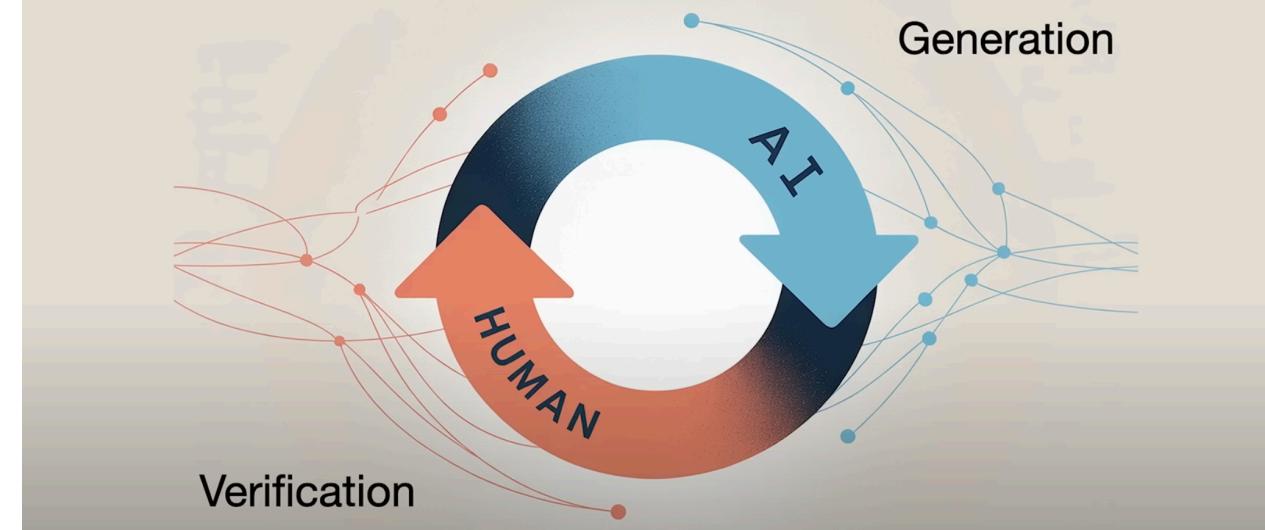


# TDD und AI-assisted coding

Zitat aus :[Andrej Karpathy: Software Is Changing \(Again\) vom 19.6.2025:](#)

"[AI] is doing the generation , the humans are doing the verification"

Consider the full workflow of partial autonomy UIUX





Neugierig geworden?

-> [zur Abstimmung](https://bodore.github.io)

# BRANDAD Development GmbH

- Wir suchen Softwareentwickler, Scrummaster, Product Owner, UI/UX-ExpertInnen
- viel Angular und Spring-boot, aber auch anderen Frameworks, offen für neues
- <https://brandad.dev>

BRANDAD

Development



## BRANDAD Development GmbH

- Wir suchen Softwareentwickler, Scrummaster, Product Owner, UI/UX-ExpertInnen
- viel Angular und Spring-boot, aber auch anderen Frameworks, offen für neues
- <https://brandad.dev>



Danke für eure Aufmerksamkeit!

Fragen ?

Präsentation unter

<https://bodore.github.io/vorträge/TDD-Nuernberg-Digital> (QR-Code links)