

Agile Testing

Vergiss Testen
Qualität als Haltung

Die Bank

Komplexität: gering

Teamgröße: klein



Die Versicherung

Komplexität: gering

Teamgröße: groß



Der Automobilkonzern

Komplexität: hoch

Teamgröße: groß



Der Gerätehersteller

Komplexität: hoch

Teamgröße: klein



Das StartUp

Komplexität: hoch

Teamgröße: klein



Teamgröße und
Komplexität machen
nicht den Unterschied...

Methodik

Fünf Fallen

#1 keine Struktur

Teststufen

- Unittests
- Integrationstests
- Akzeptanztests
- E2E-Tests
 - Unterschiedlich:
 - Fokus
 - Testziel
 - Gefundene Fehler

Unittests

- Feedback!
- Klasse/Methode/funktionale Einheit
- keine Abhängigkeiten (keine DB, Files, Systeme)
- schnell (fast feedback)
- Test one thing (single assumption)
- Qualität über Quantität (Testentwurfsmethoden, ~~viel hilft viel~~)
- Fehlerzustand schnell erreichbar
- Testbasis: Code & Struktur

Design for Testability!

Integrationstests

- Verschiedene Ebenen:
 - Integration der Units
 - Integration der Komponenten
 - Integration der Systeme
- Schnittstellentests
- Testbasis: Architektur/Design

Akzeptanztests

- Prüfung der Akzeptanzkriterien
- + links, rechts, negativ
(Testentwurfsmethodik)
- Verknüpfung Fachlichkeit & Code
 - Behavior-Driven Development
 - Acceptance Test-Driven Development

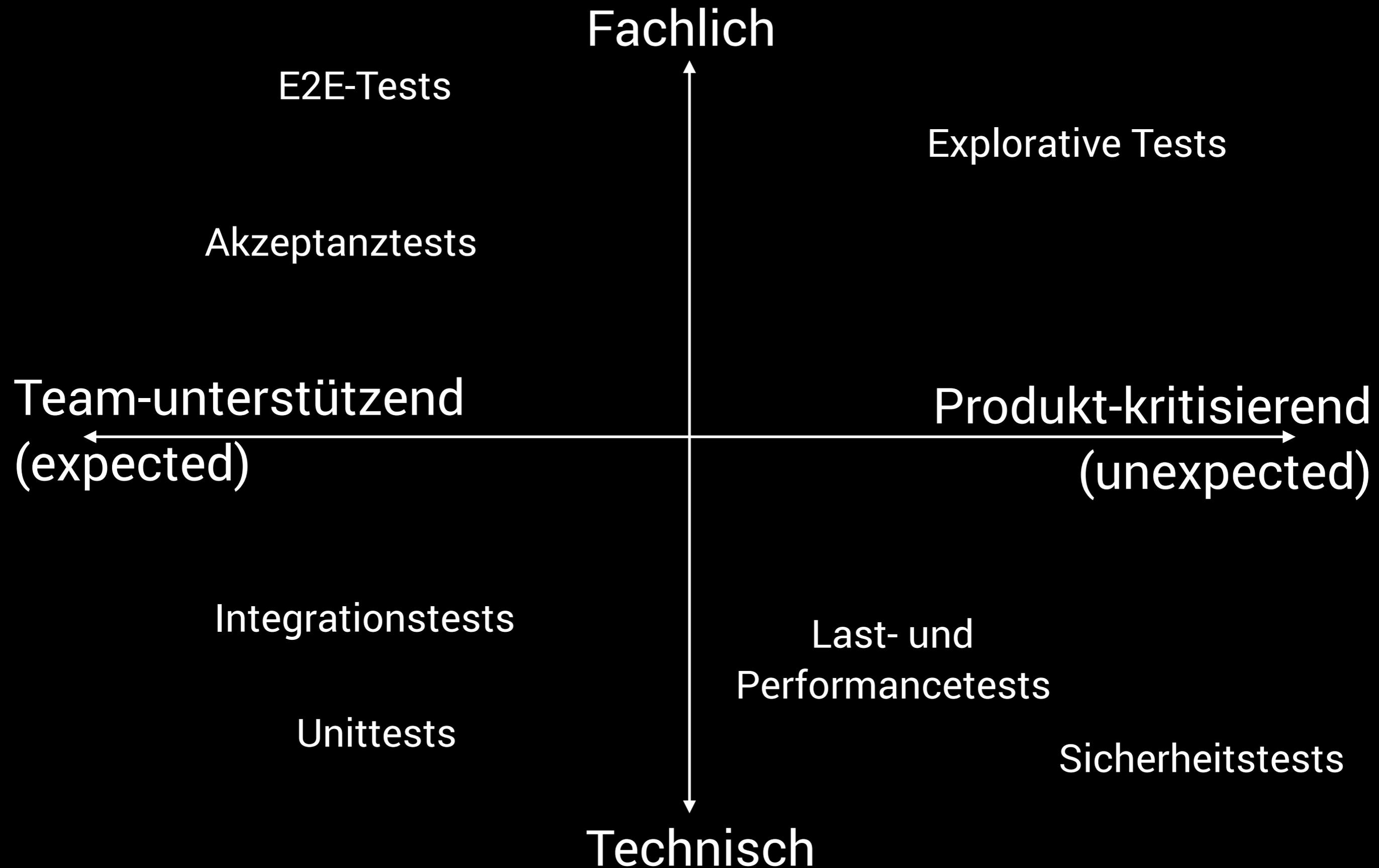
E2E-Tests

- Test auf der Oberfläche
- (oder knapp darunter -> Framework)
- Testbasis: Geschäftsprozesse, User-Stories, Epics
- Stark (geschäfts)-risikobasiert

Testarten

	Funktionalität	Effizienz	Sicherheit	...
Unittest	x			x
Integrations- test	x	x		
Akzeptanz- tests	x	x	x	
...				

Testquadranten



#2 Teststrategie

Strategie:

Risikobasiertes Testen

Unit- und Integrationstest: technisch

Akzeptanz- und E2E-Test: fachlich

#3 Testautomatisierung

E2E-Tests

expl. Tests

Akzeptanztests

Integrationstests

Unittests

Toolunabhängig!

Testautomatisierung fachlicher Tests

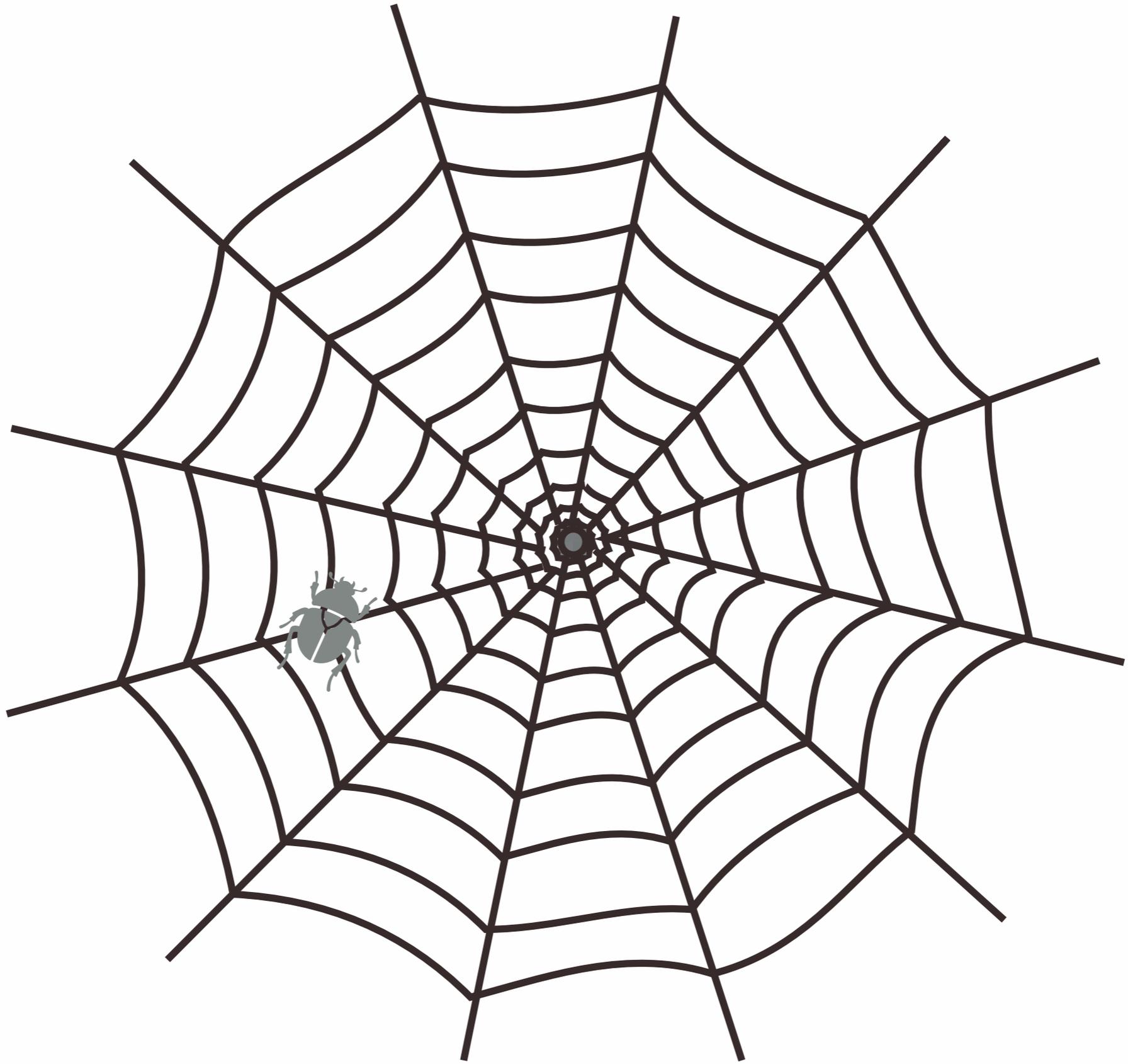
- Anforderungen/Akzeptanzkriterien um TF erweitern (z.B. BDD)
- „If you automate a mess, you get automated mess.“
- Data & Keyword-Driven
- Fachlichkeit nicht nachimplementieren
- Je Testfall
 - Definierter Start- und Endpunkt
 - Vorher/Nachher aufräumen
- Unabhängige Testfälle

#4 Statische Analysen

Statische Analysen

- Frühzeitig Fehlerzustände/Mängel finden
- Robustheit & Wartbarkeit
- Einhaltung Programmierregeln
- Prüfung gegen State-of-the-Art-Pattern
- Metriken für die Zeit
 - Quantität
 - Qualität
 - Komplexität

#5 Testmethodik



- Äquivalenzklassen
- Grenzwerte
- Entscheidungstabellen
- Zustände
- Abläufe
- Pairwise

Mindset

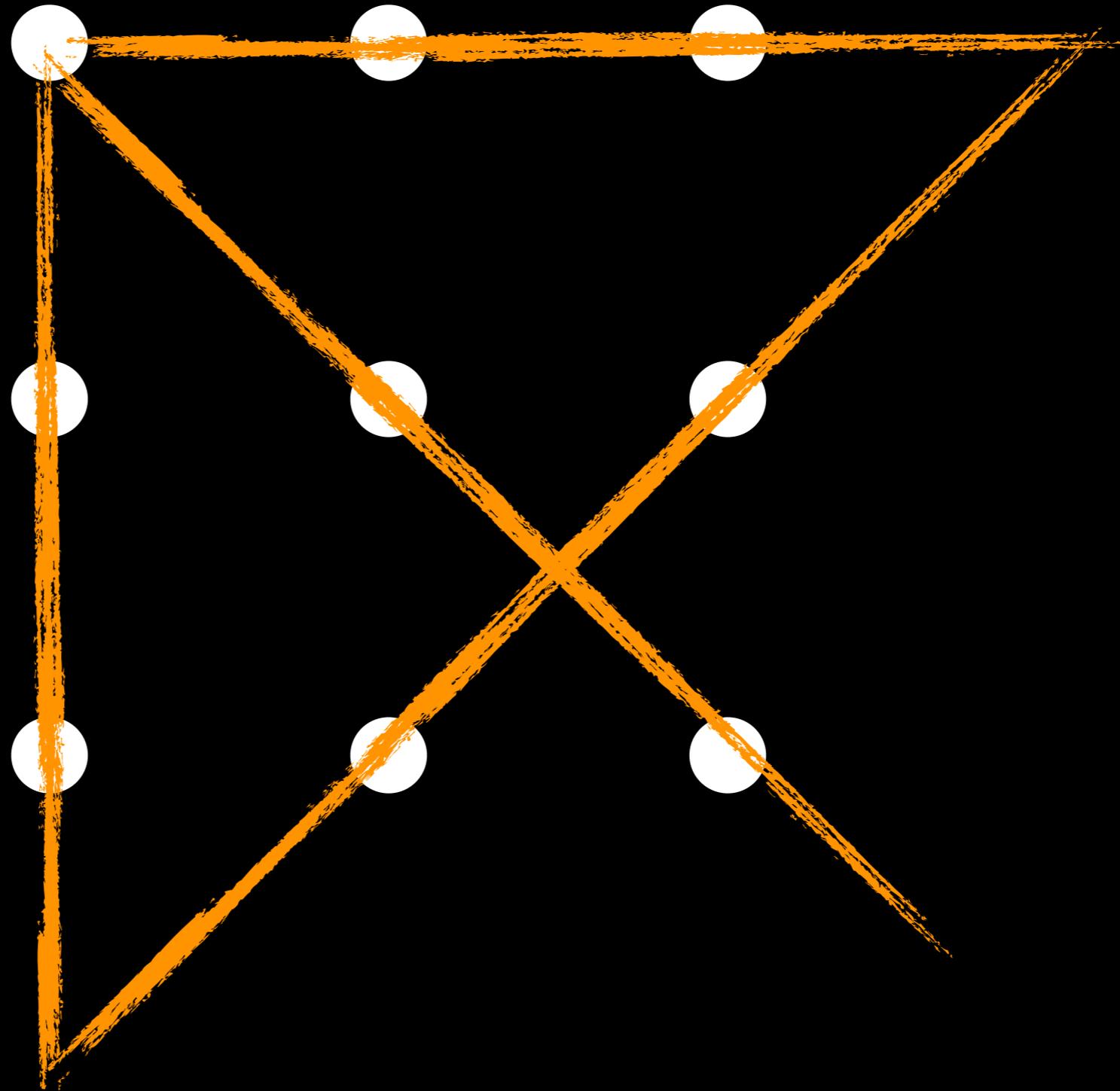
- Qualität wird ganzheitlich gesehen
- hohe Selbstverantwortung
- Testen wird nicht mehr wahrgenommen
- Testen im Flow
- Disziplin

Reflexion

- hohe Selbstreflexion im Team
- Prozesse, Vorgehen und Methoden werden laufend hinterfragt und angepasst

Integrales Team

- Rollen und Themen werden durchlässig
- Interdisziplinär
- Wertschätzung, keine Gleichmacherei



Think
Out of the Box

- Nichtwissen
„You don't know, what you don't know“
- „Probleme kann man niemals mit der Denkweise lösen, durch die sie entstanden sind“
- Albert Einstein
- Achtung vor Denkfallen (Bias)

Wie verwurzelt man nun
Qualität in agilen Teams?

Es braucht

Zeit + Energie + Geduld

Coaching

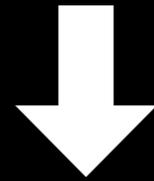
Partizipation
&
Selbstverantwortung

Retrospektiven

Agile Ideen leben statt
Dogmen zu folgen

Let's see what works

Testmethodik



Qualität als Haltung

*„Testen ist wie herumklicken
- nur krasser“*

www.richard-seidl.com

mail@richard-seidl.com

