

SIDION

Zuhören. Analysieren. Beraten.



AGILE-DRIVEN INTEGRATIONSTESTS

Azmir Abdi & Alexander John-Anacker

JAVAFORUM2021
stuttgart

WER SIND WIR?

Langjährige Erfahrung und Zusammenarbeit bei **sidion GmbH**



Azmir Abdi

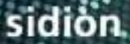
Software Architect

Team Lead Architecture, Integration
& Microservices



Alexander John-Anacker

Senior Full-Stack Developer



#wirarbeitendigital

Continuous Migration

Schrittweise und kalkulierbare Migration von Legacy Anwendungen

☐ Ihre Legacy Anwendungen sind komplex und veraltet?
 ☐ Sie wollen das Risiko einer Neuentwicklung minimieren?
 ☐ Sie wollen schneller verfügbare, stabile, flexible und skalierbare IT Lösungen?
 ☐ Sie wollen neue Services sofort produktiv nutzen?



Direkt zur Homepage

Unsere Leistungsbausteine

sidion hat langjährige Erfahrung in der Neu- und Weiterentwicklung von komplexen Anwendungslandschaften bei Groß- und Mittelstandkunden. Die Herausforderung, anspruchsvolle Legacy Applikationen schrittweise in Micro-Service-Architekturen zu überführen, haben wir schon mehrfach erfolgreich gemeistert.

Continuous Migration Ansatz

Wir verwenden einen erprobten Mix aus konventionellem Design bzw. Entwicklung und Agiler Methoden:

- Stringentes Domain Driven Design
- Architektur Blueprint (Self-contained Systems, Micro-Services, Consumer Driven Contracts) und Toolchain für fachliche und technische Shared Services
- Agiles Entwicklungs- und -Projekt-Governance-Modell
- Soviel Design wie nötig, soviel Implementierung wie möglich
- Bau der neuen Lösung Service für Service, Inbetriebnahme der fertigen Services inklusive der Kommunikation mit den Legacy Applikationen
- Schrittweiser Rückbau der Legacy Anwendung ("Continuous Decommissioning")
- Priorisierung und Bau der zu entwickelnden Services:
 - Nach Shared Services für das technische und fachliche Fundament
 - Gemäß IT-Strategie und dem angestrebten business value inklusive Kopplung mit Ihren Legacy Anwendungen



SIDION DIENSTLEISTUNG

CONTINUOUS MIGRATION

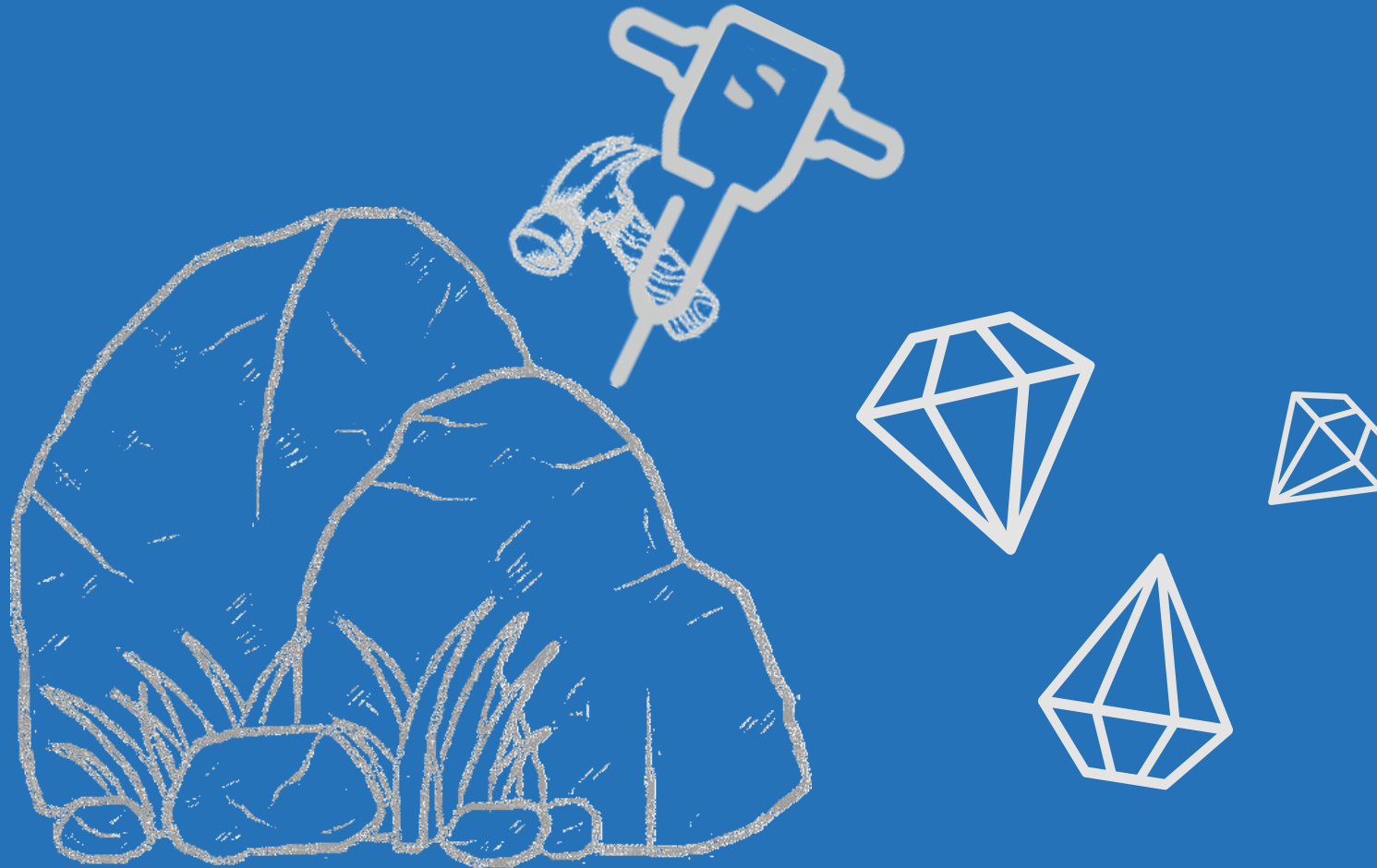
STETIGE MIGRATION AUF DIE STATE-OF-THE-ART
 METHODEN, ANSÄTZE UND TECHNOLOGIEN

AGILE. DDD. MICROSERVICES

CONTINUOUS MIGRATION

VON MONOLITHEN ZU MICROSERVICES

AGILE SCHRITTWEISE MODERNISIERUNG DER LEGACY SYSTEME



AUSGANGSSITUATION

STATUS QUO

TESTING IN MICROSERVICES PROJEKTEN

NICHT INTEGRIERTES TEST TEAM

Ein separates Test Team testet die Implementierung bevor diese Live gehen kann



DEDIZIERTE TESTER IN SCRUM TEAM

Jedes Scrum Team hat einen eigenen Tester, der die Implementierung teste



RÜCKBLICK

AGILE WERTE

AUCH TEST-ENTSCHEIDUNGEN IM AGILEN UMFELD
SOLLTEN SICH AN AGILEN WERTEN & PRINZIPIEN ORIENTIEREN



Individuen und
Interaktionen
mehr als
Prozesse und
Werkzeuge



Funktionierende
Dienstleistungen
mehr als
umfassende
Dokumentation



Zusammenarbeit
mit dem Kunden
mehr als
Vertrags-
verhandlung



Reagieren auf
Veränderung
mehr als das
Befolgen eines
Plans

RÜCKBLICK

AGILE PRINZIPIEN

AUCH TEST-ENTSCHEIDUNGEN IM AGILEN UMFELD
SOLLTEN SICH AN AGILEN WERTEN & PRINZIPIEN ORIENTIEREN



FRÜHES FEEDBACK

Lieferungen unter Berücksichtigung von Kundenrückmeldungen



VERÄNDERUNG

Auch späte Änderungen sind willkommen



REGELMÄSSIG

Liefere innerhalb weniger Wochen aus



CROSS-FUNCTIONAL

Verschiedene Bereiche in der täglichen Zusammenarbeit



INDIVIDUEN

Errichte Projekte rund um motivierte Individuen



KOMMUNIKATION

Führe Gespräche von Angesicht zu Angesicht



QUALITÄT

Liefere funktionierende Software



RHYTHMUS

Kunde und Scrum-Team halten ein gleichmäßiges Tempo



EXPERTISE

Jedes Teammitglied bringt ausreichend fachliches, soziales und organisatorisches Können mit



VERMEIDE MUDA

Eliminiere Verschwendung und unnötige Tätigkeiten



SELBST-ORGANISATION

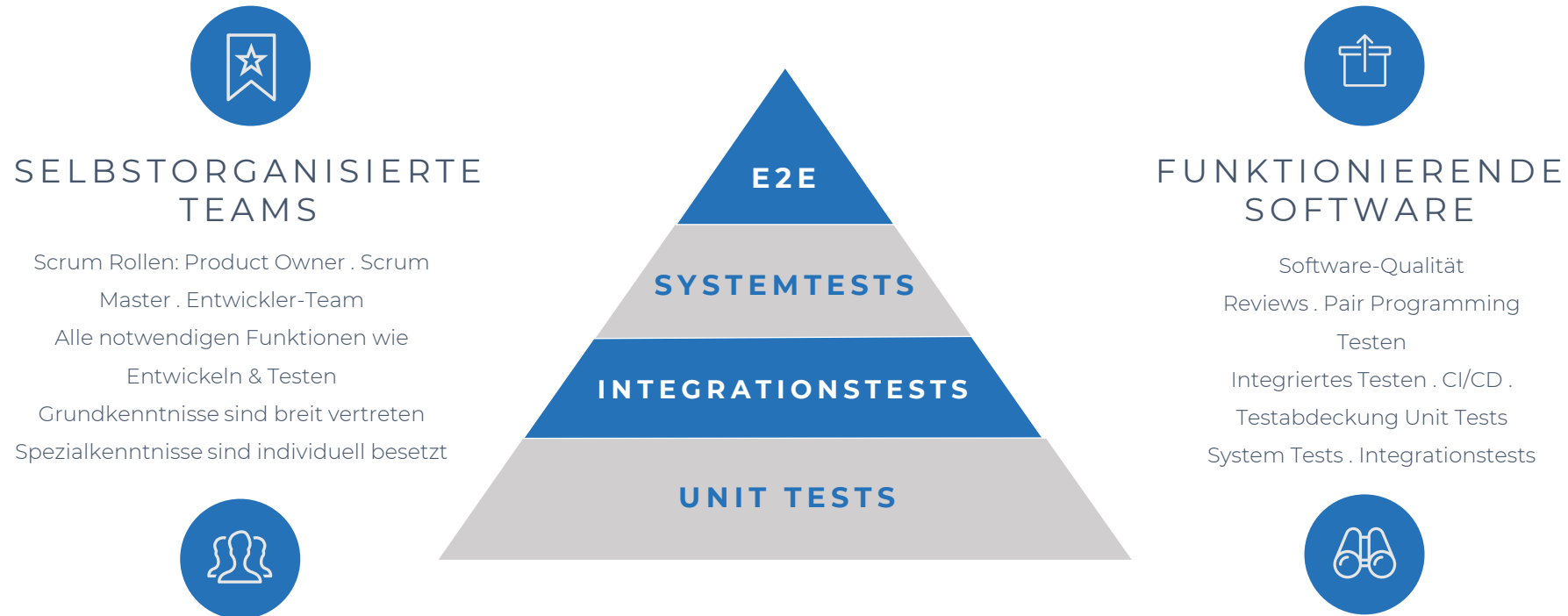
Die besten Ergebnisse entstehen durch selbstorganisierte Teams



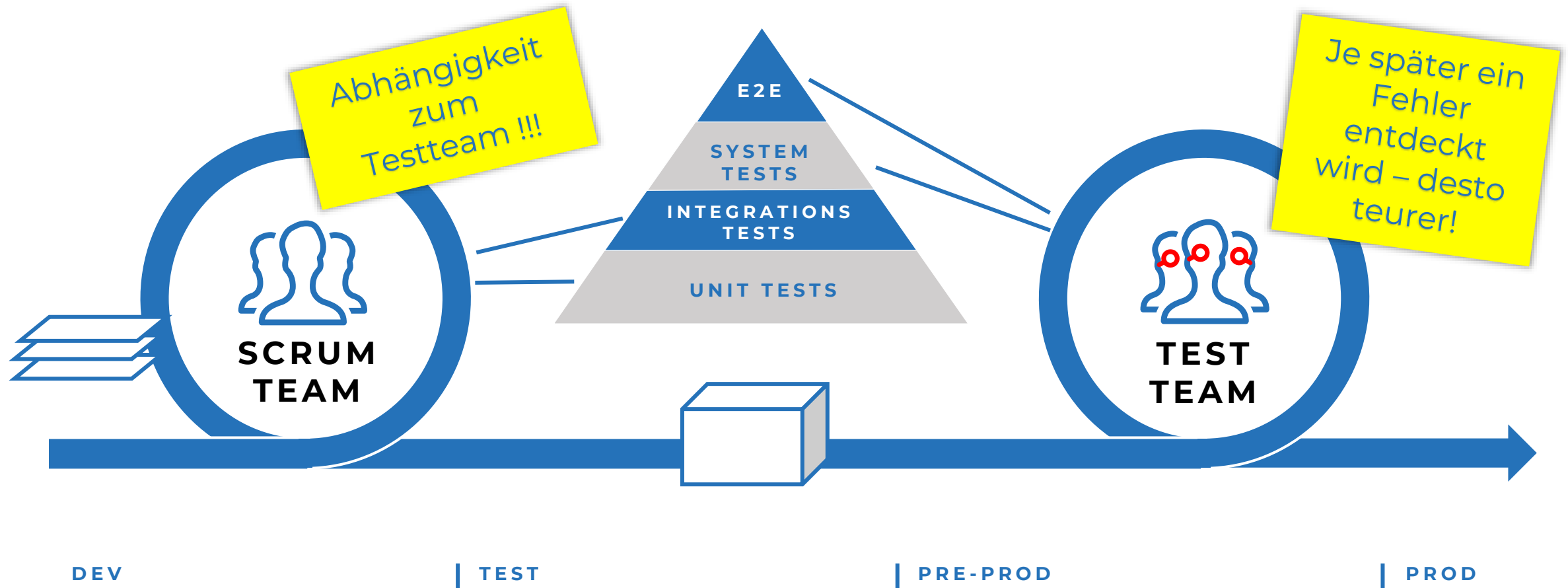
RETROSPEKTIVE

Team reflektiert regelmäßig wie es effektiver werden kann und passt sich entsprechend an

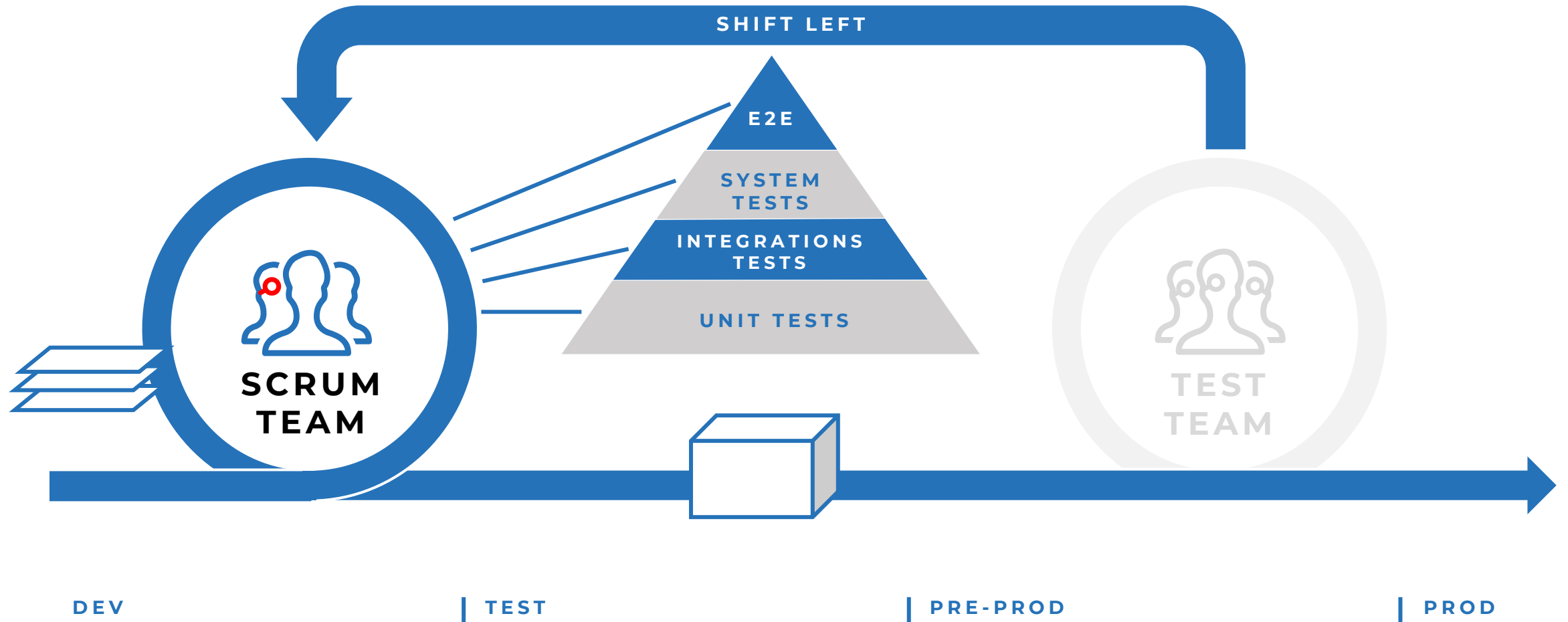
AGILE-DRIVEN INTEGRATIONSTEST

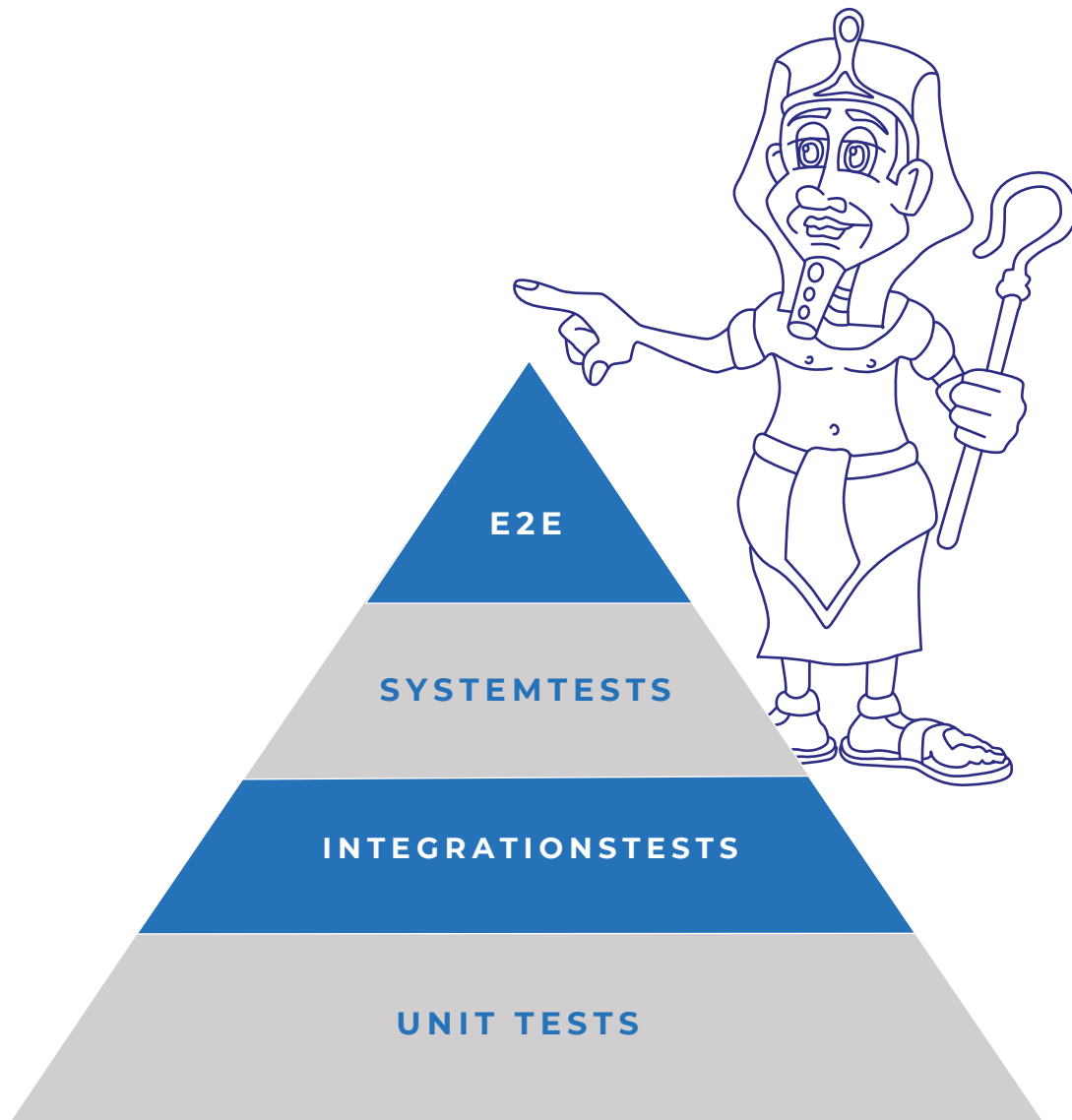


PROBLEMSTELLUNG



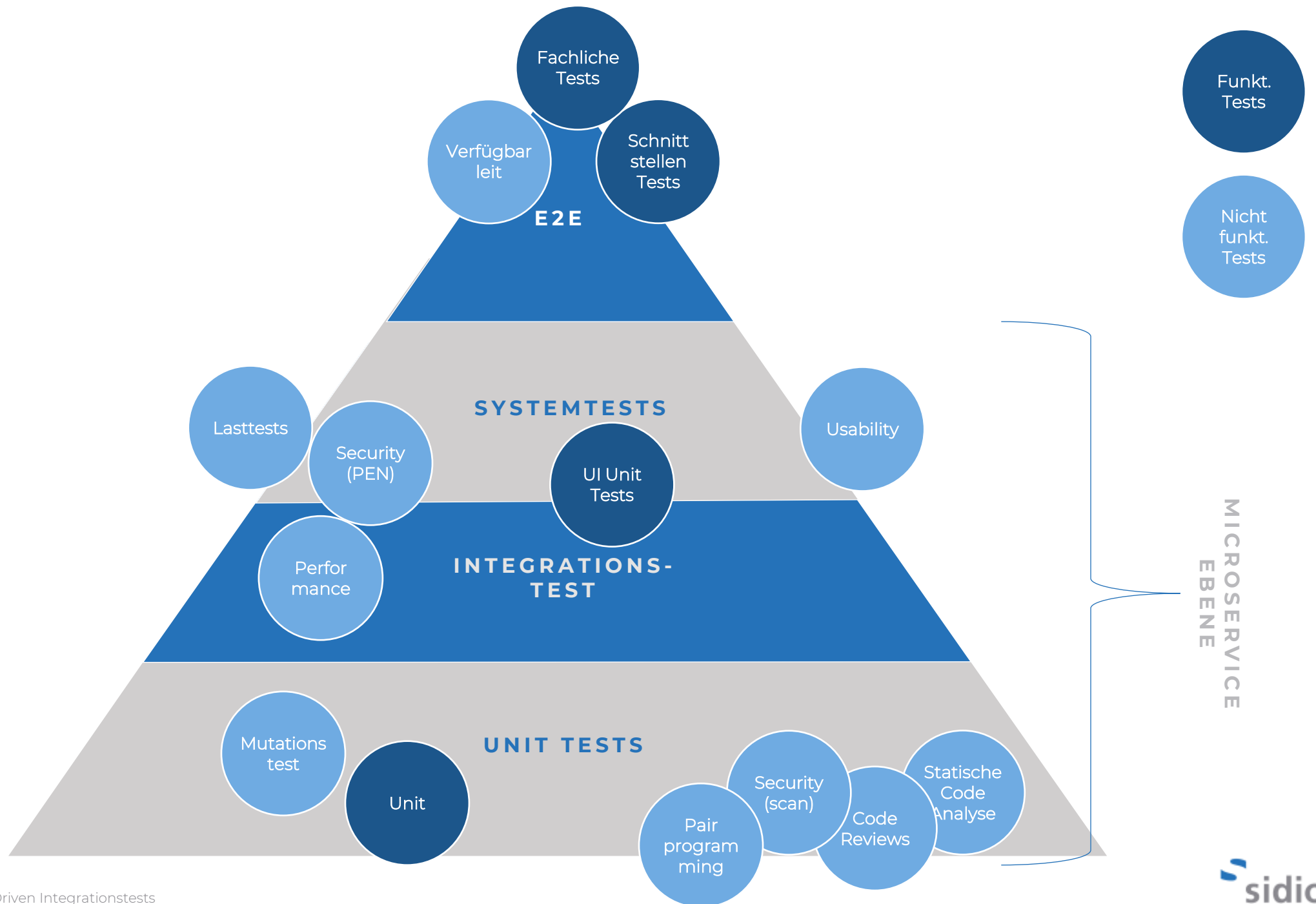
SHIFT LEFT

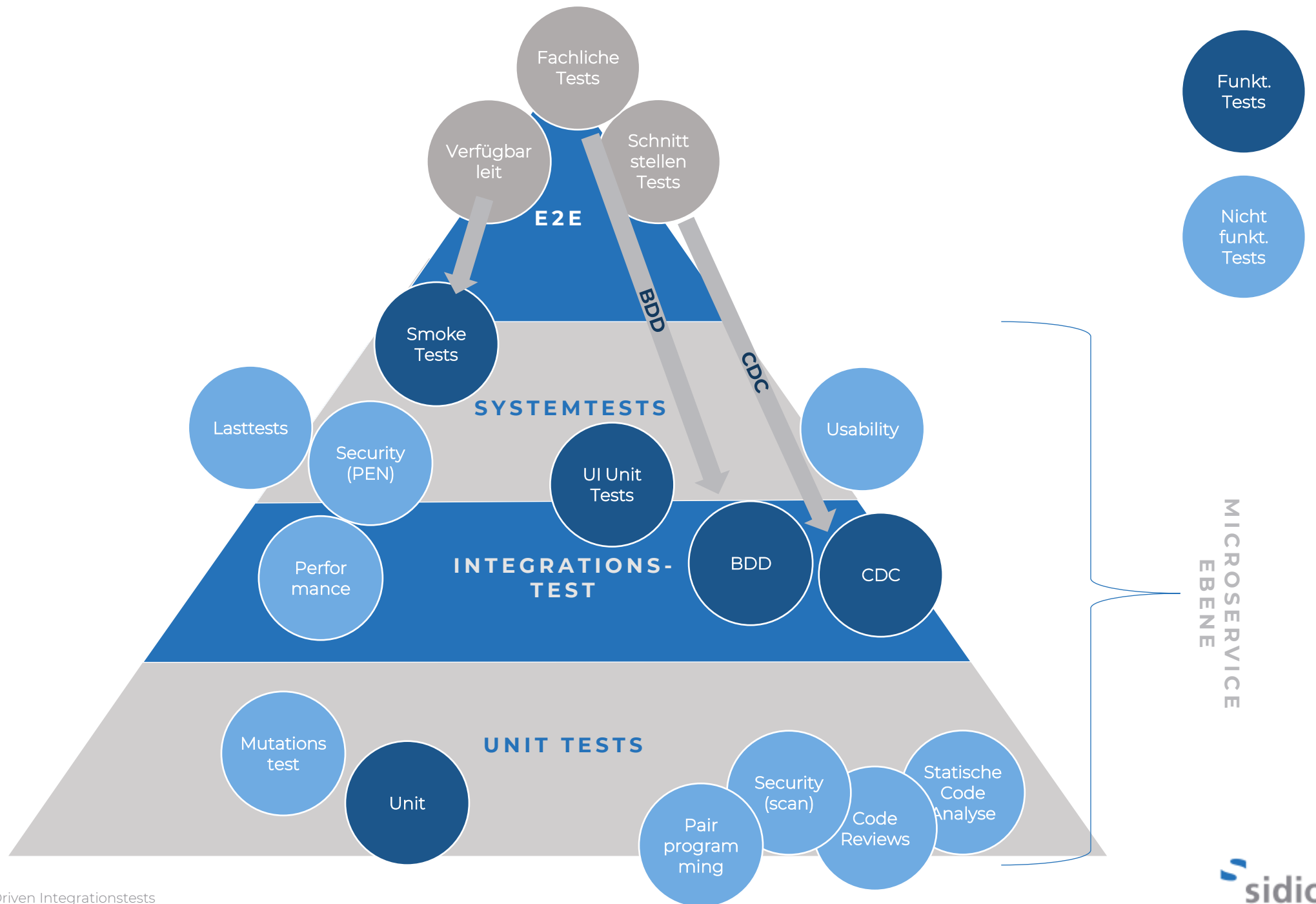




TESTPYRAMIDEN SHIFT LEFT







BEHAVIOUR-DRIVEN DEVELOPMENT

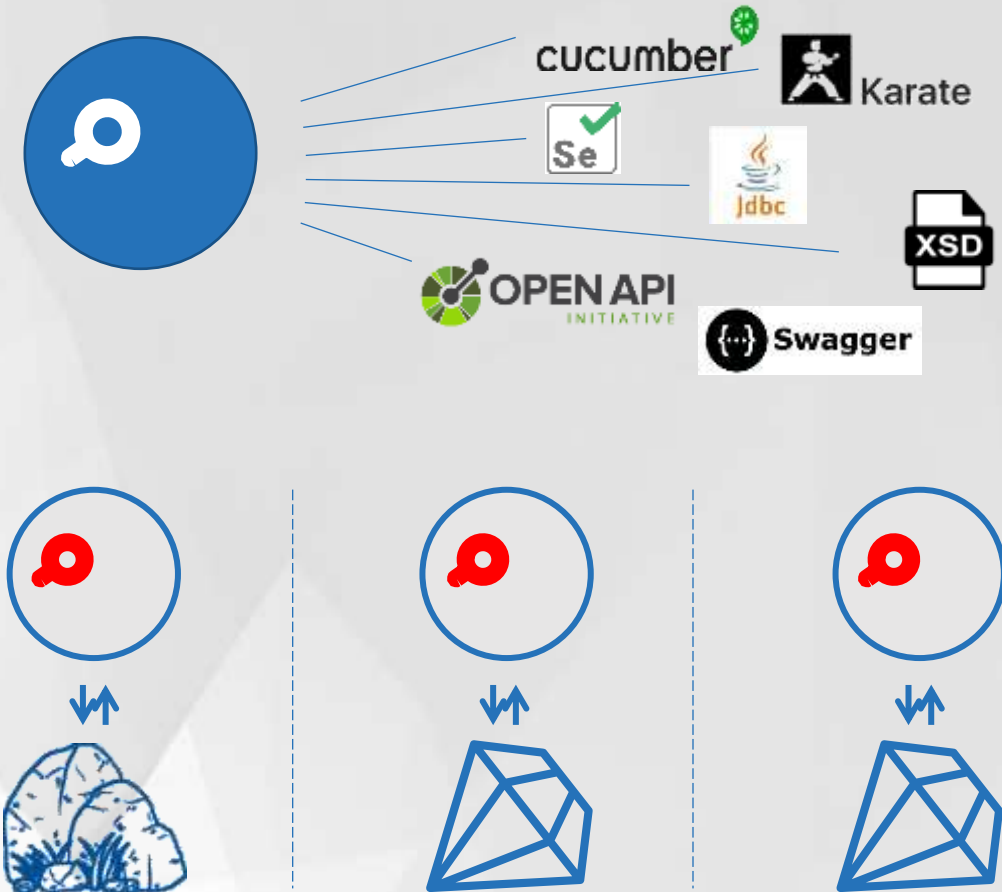
Ziel: Sicherstellung der richtigen fachlichen Umsetzung (do the right things)

Anwendungsgebiet

- Anforderungsspezifikation und Tests

Grundidee

- Anforderungen als Testspezifikationen
 - Ausführbare Akzeptanzkriterien
 - Testdaten
- Acceptance Test-Driven Development (ATDD)
- Einsatz vom Domain-Specific / Ubiquitous Language
- Erhöht Unabhängigkeit des Scrum Teams
- Technologieunabhängig



CONSUMER-DRIVEN CONTRACTS

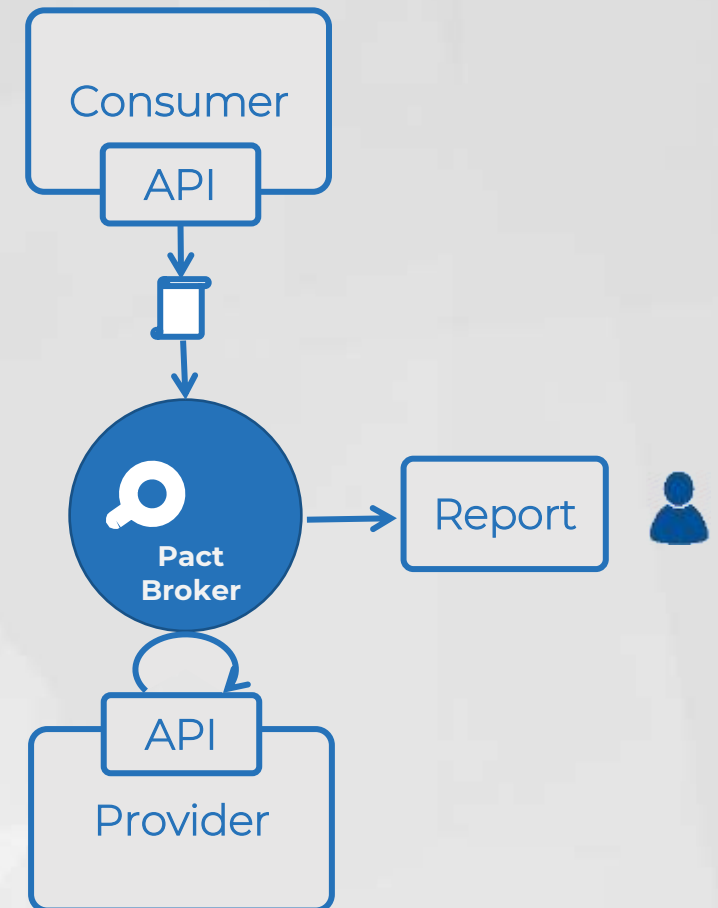
Ziel: Entkoppelte Schnittstellentests

Anwendungsgebiet

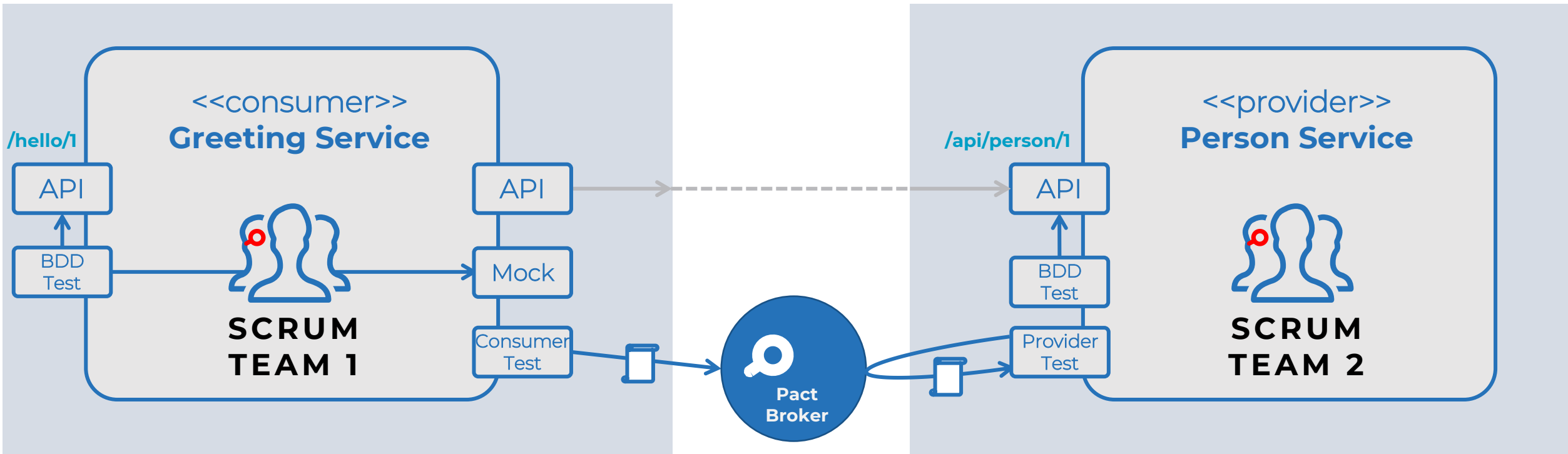
- Integrationstests

Grundidee

- Der Consumer weiß, was er benötigt, um seine Anforderungen zu erfüllen.
- Daraus ergeben sich Anforderungen an den Schnittstellen-Vertrag (Pact). Dies wird in einem File dokumentiert
- Der Consumer stellt den Vertrag dem Provider zur Verfügung. Zum Beispiel wird es auf einen zentralen Punkt veröffentlicht (Pact-Broker).
- Der Provider verifiziert bei jeder eigenen Änderung alle Verträge seiner Konsumenten. Ein fehlgeschlagener Test bedeutet Schnittstellenbruch.



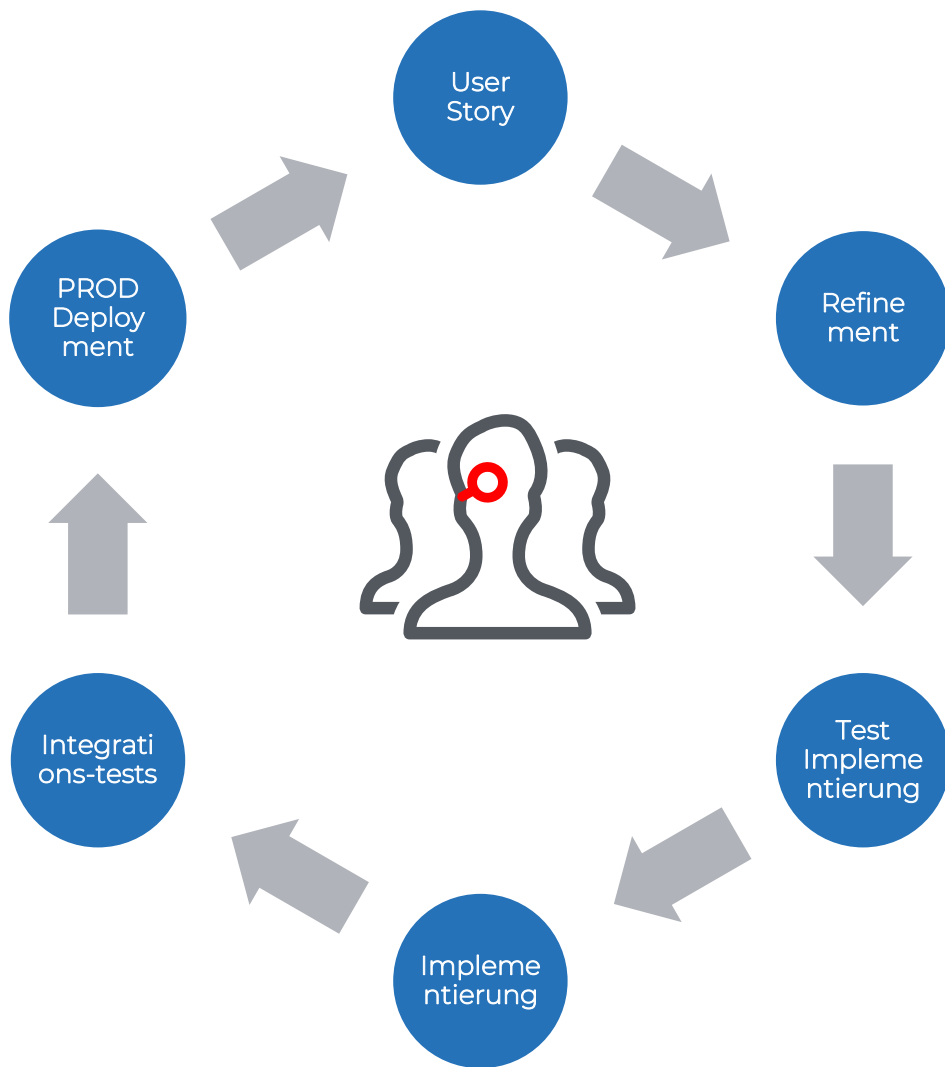
CONSUMER-DRIVEN CONTRACTS DEMO



<https://gitlab.com/sidion/javaforumstuttgart/2021/agiledrivenintegrationstests>



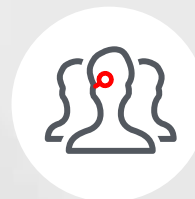
DEMO



INTEGRATIONSTESTS IN AGILEM UMFELD

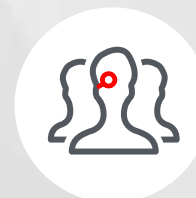
CONTINUOUS DELIVERY

Mission: DevTestOps



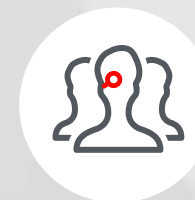
DevTestOps

Testing in allen DevOps Phasen
Verantwortung über die Integrationstests
des gesamten Teams



Tester als Know-How Träger

Refines User Story zur Testable Stories
Berät Entwickler über die
Testimplementierung



Entwickler

Implementiert Integrationstests



FRAGEN

Mitdenken
Mitgestalten
Mitarbeiten
bei sidion

✉ azmir.abdi@sidion.de
✉ alexander.john-anacker@sidion.de



[https://gitlab.com/sidion/javaforumstuttgart/
2021/agiledrivenintegrationstests](https://gitlab.com/sidion/javaforumstuttgart/2021/agiledrivenintegrationstests)