



● Workflow Analytica

**Früh testen.
Modular bauen.
Intelligent automatisieren.**

Wie ein moderner Testansatz Qualität, Tempo und Kosten gleichzeitig verbessert.

Was Sie in den nächsten 30 Minuten erwartet

- | | | |
|-----------|-------------------------------------|---|
| 01 | Grundlagen | Was ist Testen — und warum es uns alle angeht |
| 02 | Frühes Testen | Qualität entsteht im Sprint, nicht am Ende |
| 03 | Komponentenbasiert | Modularer Aufbau statt monolithischer Tests |
| 04 | Intelligente Automatisierung | Parameter, GeVos, End-to-End |
| 05 | Unser Ansatz | Verzahnung mit Entwicklung, BA und Projekt |

Was ist Testen — und was ist es nicht?

Definition (ISTQB)

Testen ist ein Prozess innerhalb des Software-Lebenszyklus, der Risiken sichtbar macht, Annahmen überprüft und Vertrauen in das Produkt aufbaut — vor der Auslieferung an den Kunden.

Testen IST ...

- Risiko-Management mit Methode
- ein integrierter Bestandteil der Entwicklung
- eine Investition in Stabilität und Tempo

Testen IST NICHT ...

- ein Schritt am Ende des Projekts
- die alleinige Aufgabe einer Abteilung
- ein Beweis für Fehlerfreiheit

Warum wir testen — und was passiert, wenn wir es lassen

100×

höhere Behebungskosten
in Produktion vs. Anforderung

Boehm / IBM Systems Sciences

60 %

der Software-Defekte
entstehen während der Entwicklung

Capers Jones, Software Quality

30 %

der Projektkosten
im Schnitt für Bugfixing

NIST · Branchenstudien



Was passiert, wenn wir nicht (oder zu spät) testen?

Defekte erreichen den Kunden · Releases verzögern sich · Vertrauen in das Produkt sinkt · Aufwände für Bugfixing verdrängen Innovation · regulatorische Risiken (z. B. DORA, MaRisk).

Wann beginnt Testen wirklich? — Shift Left



← **SHIFT LEFT**

Hier liegt unser Hebel



Mit dem Lesen der Anforderung

Reviews und Testbarkeits-Checks finden Lücken, bevor Code entsteht.



Mit dem ersten Modell oder Prototyp

Frühes Feedback zu Logik und Schnittstellen statt späte Überraschungen.



Mit jedem User Story Refinement

Akzeptanzkriterien werden testbar formuliert (Given · When · Then).

Was sich ändert, wenn Tests früh starten



Fehler früh gefunden

Defekte werden im Sprint behoben — nicht in der Abnahmephase.



Kosten reduziert

Bugfixing-Aufwände sinken signifikant; Budget bleibt für Features.



Enge Zusammenarbeit

Test, Entwicklung und BA arbeiten parallel — direkte Kommunikation, kurze Wege.



Komponenten parallel reif

Module entstehen testbereit und werden im Baukastensystem zusammengeführt.



Schnellere Releases

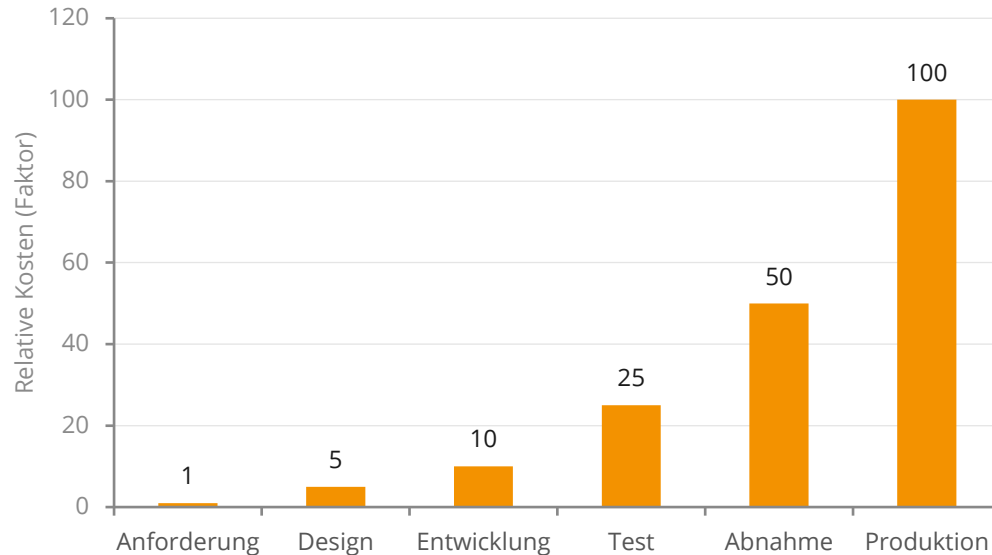
Qualität steckt im Inkrement — Release-Phasen verkürzen sich messbar.



Höhere Stabilität

Risiken werden über Iterationen reduziert statt am Ende kumuliert.

Was kostet ein Defekt — abhängig vom Zeitpunkt?



Die Logik dahinter

Je später ein Defekt entdeckt wird, desto mehr Code, Doku und Abhängigkeiten müssen mit-angefasst werden.

Faktor 100

Anforderung vs. Produktion

Quelle: Boehm · IBM Systems Sciences

Tests wie ein Baukasten — modular und wiederverwendbar

Statt monolithischer Testfälle bauen wir kleine, eigenständige Bausteine, die einzeln und in Kombination funktionieren.



Atomar

Jede Komponente prüft genau eine fachliche Funktion.



Wiederverwendbar

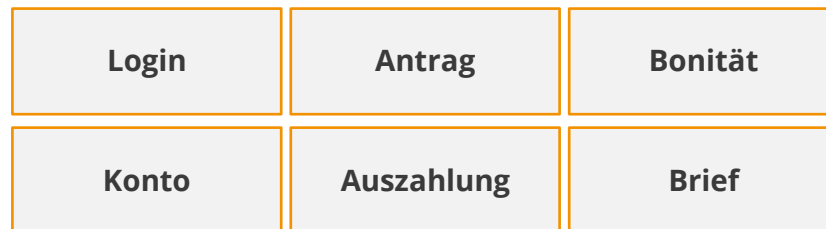
Gleicher Baustein in vielen Geschäftsvorfällen.



Kombinierbar

Komponenten reihen sich zu End-to-End-Strecken.

Komponenten-Bibliothek



GeVo: Online-Kreditantrag

Login · Antrag · Bonität · Konto · Auszahlung

Ein Geschäftsvorfall — beliebig viele Varianten

Beispiel: Kfz-Versicherungsantrag — gleicher Ablauf, andere Daten

Variante	Kunde	Fahrzeug	Schadenfreiheit	Erwartetes Ergebnis
V1	Privatkunde, 35 J.	VW Golf, 110 PS	SF 12	Tarif A, Standardprämie
V2	Fahranfänger, 19 J.	BMW M3	SF 0	Risikozuschlag, manuelle Prüfung
V3	Gewerbekunde	Sprinter, Flotte	n/a	Flottentarif, individuelle Police
V4	Bestandskunde	E-Auto	SF 25	Öko-Rabatt + Treuebonus



Ein Testablauf · vier Varianten · null doppelte Pflege — Daten kommen aus einer Tabelle, der Ablauf bleibt identisch.

End-to-End: Schnittstellen einfach mitgetestet



End-to-End-Strecke

✓ Vorteil

Bestehende Komponenten werden um Schnittstellen-Aufrufe erweitert. Aus Komponententests entstehen E2E-Strecken — ohne komplette Neuentwicklung.

⚠ Voraussetzung

Zugang zu Drittsystemen (Test-Stages, Stubs oder produktive APIs mit Testdaten) muss früh organisiert sein — ein klassischer Engpass in Projekten.

Eng verzahnt mit Entwicklung, BA und Projekt

Unser Test-Team ist kein nachgelagertes Silo — wir arbeiten parallel mit allen Projektrollen, vom Refinement bis zum Release.

Business Analyse

- Anforderungs-Reviews gemeinsam
- Akzeptanzkriterien testbar formulieren
- Komponentengrenzen mitdefinieren

Lücken werden vor der Entwicklung sichtbar.

Entwicklung

- Tester sitzen in Daily und Refinement
- Test-Komponenten parallel zum Code
- Defekte direkt im Sprint klären

Kein Bug-Backlog am Sprint-Ende.

Projektleitung & PO

- Transparente Risikolage jederzeit
- Belastbare Release-Aussagen
- ISTQB-konforme Berichterstattung

Entscheidungen auf belastbarer Basis.

Was Sie als BA, PO oder Projektleiter beitragen können

Business Analyst

- Anforderungen testbar formulieren, z.B. Gherkins: (Given · When · Then)
- Akzeptanzkriterien gemeinsam mit Test reviews
- Fachliche Komponentengrenzen mitdefinieren

Product Owner

- Test als Definition of Ready / Done verankern
- Testergebnisse als Entscheidungsgrundlage nutzen
- Refinements für Test öffnen — keine separate Schleife

Projektleitung / Channel Manager

- Zugang zu Drittsystemen früh organisieren
- Testdaten und -umgebungen als Lieferobjekt einplanen
- Risiko-basierte Release-Entscheidung statt Termin-Druck

Erprobt bei führenden Versicherern und Banken

Auszug unserer Referenzen

Versicherungen

Allianz, MunichRE
Versicherungskammer Bayern
HDI

Banken

DEKA, UBS
Commerzbank
Raiffeisen Schweiz

Sonstige

BMW, MINI
V-ZUG, MAN Trucks & Busses
BWI – Bundeswehr IT, IT.N – IT Niedersachsen
uvm.



Was unsere Kunden eint: regulierter Markt, hohe Qualitätsanforderungen, komplexe Systemlandschaften.

● AUF EINEN BLICK

Drei Hebel — eine Wirkung.

01

Früh testen

Defekte werden im Sprint behoben — bevor sie teuer werden.

02

Modular bauen

Komponenten parallel zur Entwicklung — wiederverwendbar, parametrisiert.

03

Intelligent automatisieren

GeVos in jeder Variante · End-to-End ohne Neubau.



QUALITYONE

HQ

Tallinn Estland

www.quality.one

Offices

München, Valencia

hello@quality.one

QualityOne . pure play testing