

Effizientes  
SAP® Testmanagement

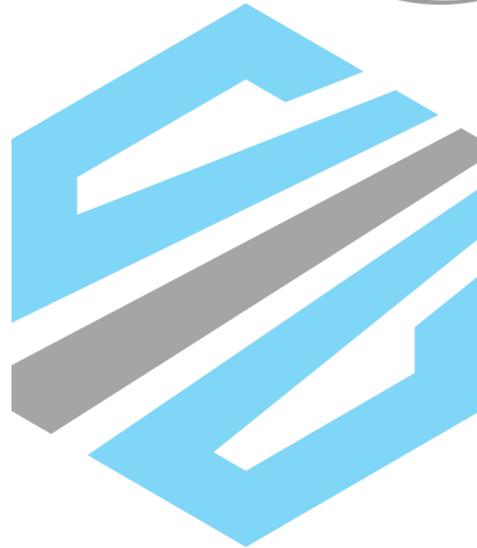


Synaworks  
Academy



SKYWAY

# Kennenlernen





## Daniel Kohl | Synaworks GmbH

Als Gründer von Synaworks bringt er mehr als 15 Jahre SAP-Beratungserfahrung in nationalen und internationalen Kontexten ein, mit einem besonderen Fokus auf die Verbindung von Menschen und Technologie für erfolgreiche SAP ERP Transformationen.

Mit umfassendem Wissen im SAP ALM und einer Historie als Consulting Manager, ist er eine treibende Kraft bei der Entwicklung und Optimierung von IT-Strategien, die sowohl den wichtigen Faktor Mensch als auch Technik in den Vordergrund stellen.



## Dr. Björn Gelhausen | SKYWAY GmbH

Strategie, Prozess- und Organisationsberatung sowie ALM und IT for Business sind seine Fokusthemen basierend auf 25 Jahren Erfahrung in der Beratung und auf Kundenseite.

Als DSAG-Sprecher für die Arbeitsgruppen Testmanagement und Testautomatisierung setzt er sich seit mehr als einem Jahrzehnt für die ganzheitliche Weiterentwicklung des SAP Solution Managers und SAP Cloud ALM ein und bereichert diese Themen mit zahlreichen eigenen Erfahrungswerten.



## Holger Stumm | Log2

Dies sind spannende Zeiten. Das Business wird sich in Schockwellen verändern und als Unternehmer bin ich mit den Themen KI, Security und SAP-Unternehmenslandschaften aktuell engagiert.

Mit mehr als 30 Jahren SAP-Erfahrung und mehr als 10 Jahren Erfahrung in internationalen SAP-Sicherheitsprojekten bieten wir Pen-Tests für SAP-Systeme. Aber auch die Absicherung der klassischen IT-Landschaft sowie alle Themen von Sicherheit, Strategie und Cyber-Resilience gehören zu meinen Kern-Kompetenzen.



## Markus Scheuber | Synaworks GmbH

Markus ist Mitgründer von Synaworks und als Consulting Manager verantwortlich für den Bereich Technology.

Mit mehr als 15 Jahren Erfahrung im SAP-Umfeld betreut er nationale und internationale Kunden bei SAP ERP-Transformations- und Migrationsprojekten im Application Lifecycle Management (ALM) mit Schwerpunkt im Release-, Prozess- & Test Management.

Markus ist zertifiziert als ISTQB Certified Tester und Test Manager.



End-to-End  
Transformationsbegleitung  
für die SAP-IT seit 2018

## Weil digitale Transformation nicht an der Technologie scheitert...

Die digitale Transformation ist Teamsport für IT und Fachbereiche. Wir helfen IT-Entscheidern den Reifegrad Ihrer SAP-IT Organisation auf das nächste Level zu bringen und sich als Innovations-partner im Unternehmen zu etablieren.

## Wir gestalten die SAP-IT der Zukunft und begleiten bei der Umsetzung



Unsere Academy bietet ein umfangreiches Seminar und Trainingsangebot

Mach die  
Fachbereiche zum  
besten Freund  
der SAP-IT.  
Termin buchen:



Sprich mit uns, um das volle  
Potential der SAP-IT zu entfalten.

consulting@synaworks.com

+49 6221 64704 30

www.synaworks.com



## Connecting People and Technology



### Unsere Kunden

30+ Kunden aus allen Branchen  
Von KMU bis DAX-30 in DACH



### Unsere Projekterfahrung

100+ Projekte in 5 Jahren  
120+ Jahre SAP-Expertise im Team



### Unser Partner-Netzwerk

15 spezialisierte SAP-Lösungsanbieter  
150+ Fachexperten



### Unsere Community

30+ IT Leiter im aktiven Austausch



### Unser Team

- Change Advisor •
- SAP Test Manager •
- SAP (Cloud) ALM Expert •
- Agile Coach •



## Portfolio:

- Organisations- und Prozessberatung
- Programm- und Projektmanagement, Methodik und Governance für S/4HANA-Einführungen
- Fiori Architekt, Entwicklungen und Fiori Administration
- Applikation Lifecycle Management (mit dem SAP Solution Manger, ServiceNow, ...) für hybride Systemlandschaften von der Organisation bis zur Technik
- Prozessmodellierung und Dokumentation (mit dem SAP Solution Manager, Signavio, Symbio, Cubetto, ...)
- Testmanagement in all seinen Facetten bis zum Testmanagement der neusten Generation
- ITSM – mit unterschiedlichsten (integrierten)
- Kundenentwicklungen auf höchstem Niveau (auf Basis zahlreicher Programmiersprachen)
- Integrationslösungen, Digitalisierung und IoT



SKYWAY

- Gegründet 12.08.2021 in Grünstadt / Pfalz
- Aufnahme der Geschäftstätigkeit am 01.10.2021
- Zufriedene Kunden: 20
- 9 gelebte Partnerschaften
- Vollausslastung seit Anfang an
- Beteiligung an der Firma HyPrint – IoT-Sensorik



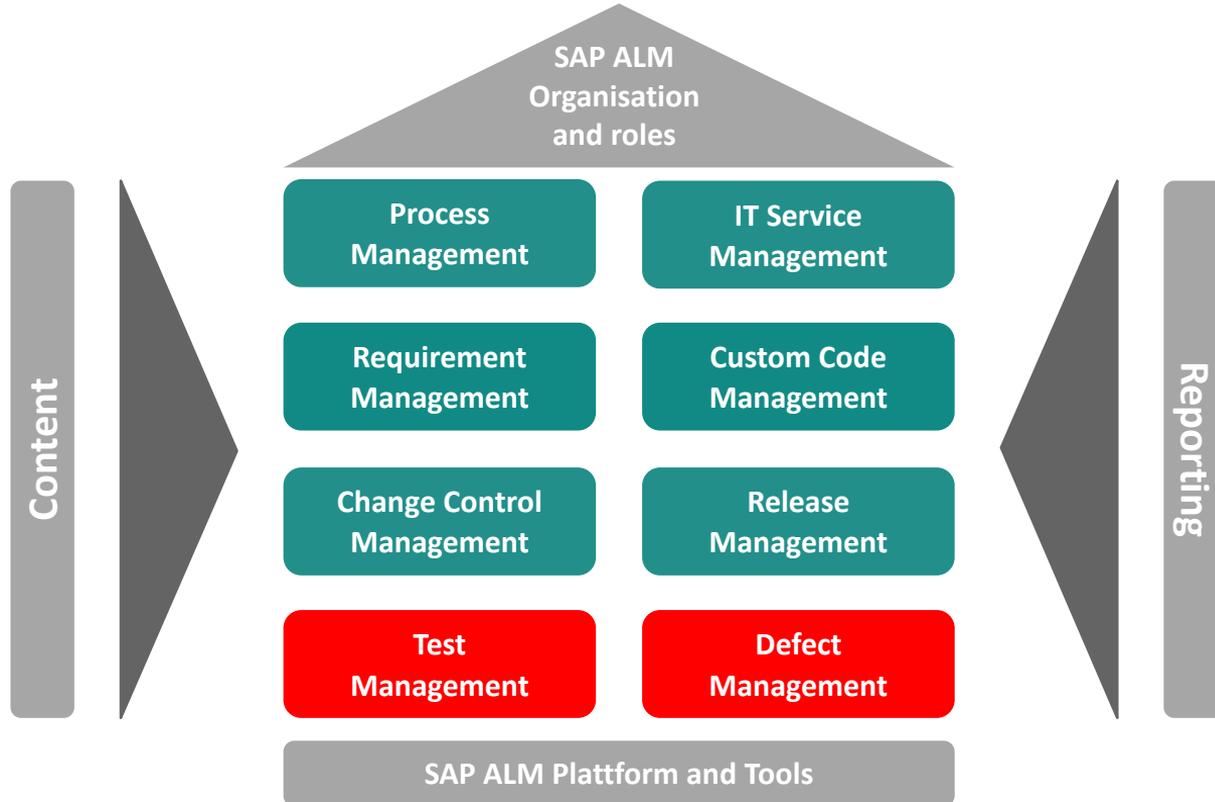
# Wie checke ich hier heute ein?



Erwartung an dieses Seminar

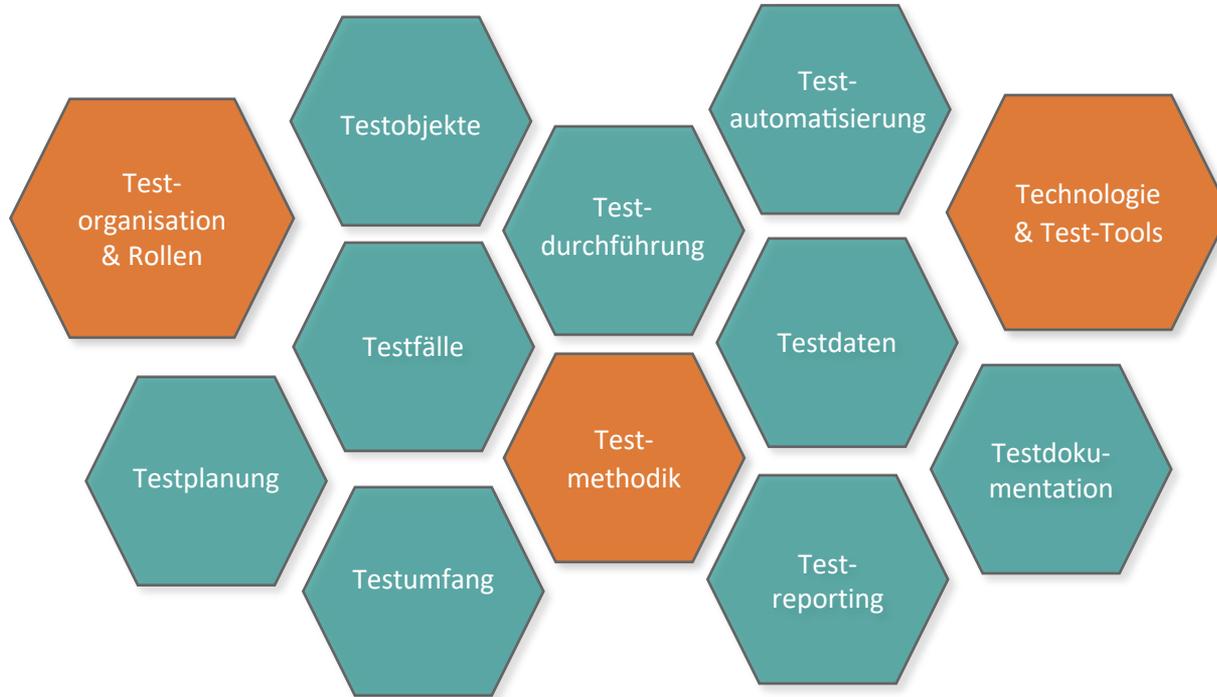
Hindernisse von Veränderung (spez. im Testmanagement)

Wo stehen Sie aktuell? Selbsteinschätzung



# SAP Testmanagement Landkarte

## 12 Handlungsfelder



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Test-Reporting

## Vormittag

Die Testorganisation

Das „Richtige“ Testen

Erfahrungsberichte II.

Mittagspause

## Nachmittag

Exkurs: Penetrationstest

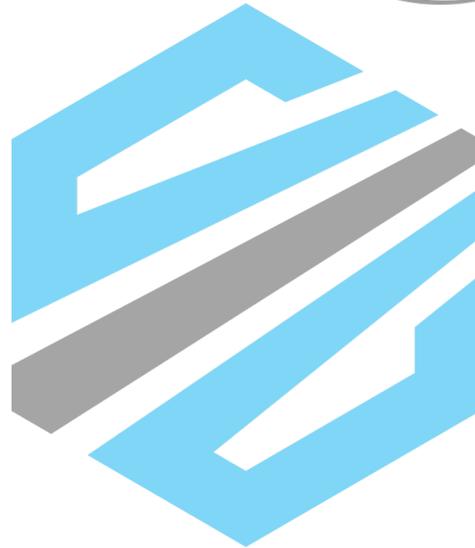
Toolvergleich

Abschluss

Kaffeepause



# Das 1x1 des Testens



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Test-Reporting

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement

## Fragen, die dieses Kapitel beantwortet:

- Wozu testen wir?
- Warum sollten Fehler möglichst früh gefunden werden?
- Welche Testarten gibt es?
- Welche Teststufen gibt es und wie unterscheiden sie sich?
- Wie entsteht ein priorisierter Testumfang?
- Welchen Unterschied zwischen Fehlhandlung, Fehlerzustand und Fehlerwirkung gibt es?

## Wozu testen wir?

- Arbeitsergebnisse bewerten
- Verifizieren der Anforderungen
- Vollständigkeit prüfen
- Vertrauen in das Testobjekt schaffen
- Fehler aufdecken

## Effizienz

- Zeitverhalten
- Verbrauchsverhalten

## Funktionalität

- Angemessenheit
- Richtigkeit
- Interoperabilität
- Ordnungsmäßigkeit
- Sicherheit

## Zuverlässigkeit

- Reife
- Fehlertoleranz
- Wiederherstellbarkeit

## Benutzbarkeit

- Verständlichkeit
- Erlernbarkeit
- Bedienbarkeit

## Änderbarkeit

- Analysierbarkeit
- Modifizierbarkeit
- Stabilität
- Prüfbarkeit

## Übertragbarkeit

- Anpassbarkeit
- Installierbarkeit
- Konformität
- Austauschbarkeit

- Testen zeigt die Anwesenheit von Fehlern
- Vollständiges Testen ist nicht möglich
- Mit dem Testen frühzeitig beginnen
- Häufung von Fehlern
- Zunehmende Testresistenz
- Testen ist abhängig vom Umfeld
- Trugschluss: Keine Fehler bedeutet brauchbares System

## Management-Sicht

Testen ist teuer

Keine ausreichenden Ressourcen vorhanden

Zeit nicht ausreichend

## Individuelle-Sicht

Testobjekt ( z.B. Prozesse) nicht klar definiert

Meine Arbeit als Entwickler wird bewertet

Schwachstellen werden aufgedeckt

Persönliche Auslastung und kurze Vorlaufzeiten

Mehrwerte nicht erkennbar

# Die zehn größten Fehler im Testmanagement



1. Verzicht auf Testmanagement
2. Unterschätzte Komplexität und Vorbereitungsdauer
3. Schlechtes Testdatenmanagement
4. Fehlende Automatisierung
5. Unterschätzter Know-How Bedarf
6. Verzicht auf methodisches Vorgehen
7. Ungeeignete Testwerkzeuge
8. Unpassende, zu viele oder zu wenig Testfälle
9. Last- und Performance-Test erst am Projektende
10. Unterschätzte Folgen fehlerhafter Software

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

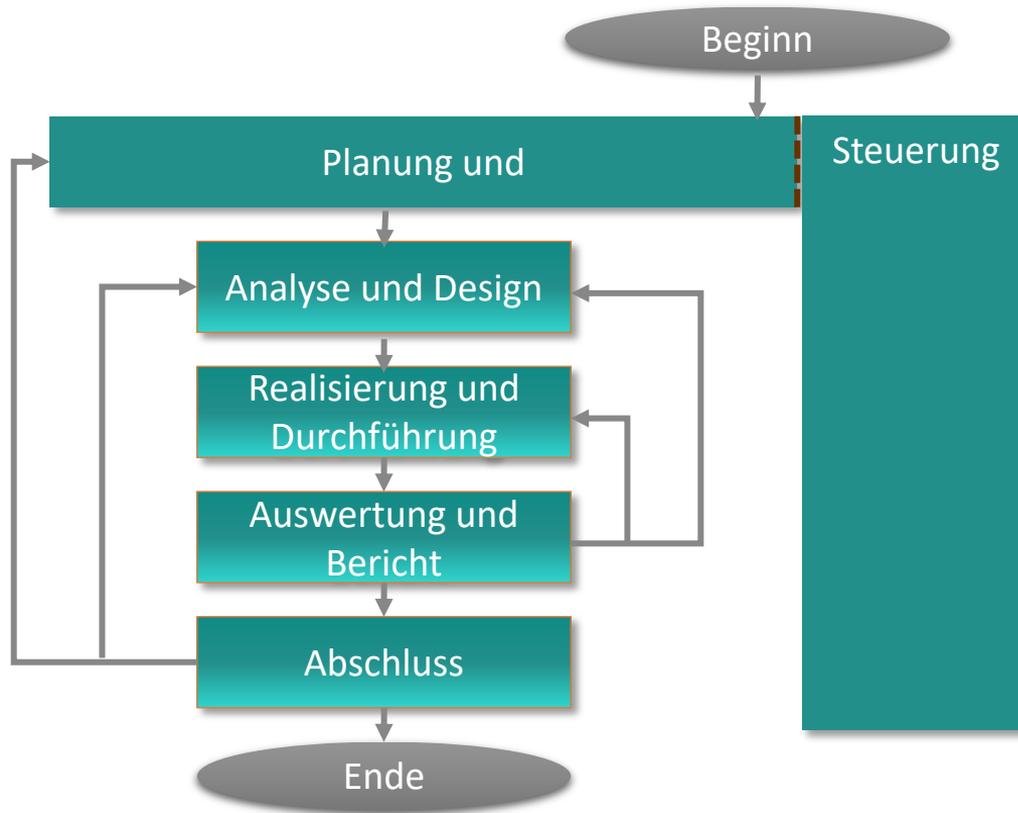
Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement



- Aufgaben und Zielsetzung festlegen
- Benötigte Ressourcen planen (Mitarbeiter, Zeitraum, Dauer, Aufwand)
- Schulung der Mitarbeiter
- Zu den Aufgaben des Testmanagers gehören
  - Verwaltung des Testprozesses (Teststrategie)
  - Testinfrastruktur
  - Testmittel
  - Überwachung Testfortschritt
  - Testaktivitäten nach Risikoeinschätzung verteilen
- Testendekriterien
  - Der Überdeckungsgrad dient als Kriterium, um ein Ende der Tests festzulegen
  - Priorisierung der Tests (kritische Testfälle am Anfang!)

Überprüfung der Durchführbarkeit der Kernprozesse und Nachweisbarkeit der Systemqualität

Verifikation der Integration der SAP Module untereinander und nach außen zu Drittsystemen

Verifikation, dass die Umsetzung den Anforderungen entsprechend realisiert wurde

Identifikation von Abweichungen, die ohne Behebung dazu führen würden, dass die Produktivsetzung verhindert wird

Verifikation der nicht-funktionalen Anforderungen an das SAP S/4HANA-System

Wiederverwendbarkeit von Testfällen und Testszenarien für effizientes Testen im Tagesgeschäft

Unklare  
Verantwortlichkeiten

Fehlschätzung  
Testaufwände

Kapazitäten und  
Ressourcenengpässe

Nicht performante  
Testsysteme

Verzögerte  
Bereitstellung  
Testumgebung &  
Testdaten

Fehlende SAP-  
Berechtigungen

Testfälle: Nicht  
rechtzeitig da, nicht  
qualitativ hochwertig

Aufwand der  
Fehlerbehebung

Innere Ablehnung  
des Testteams

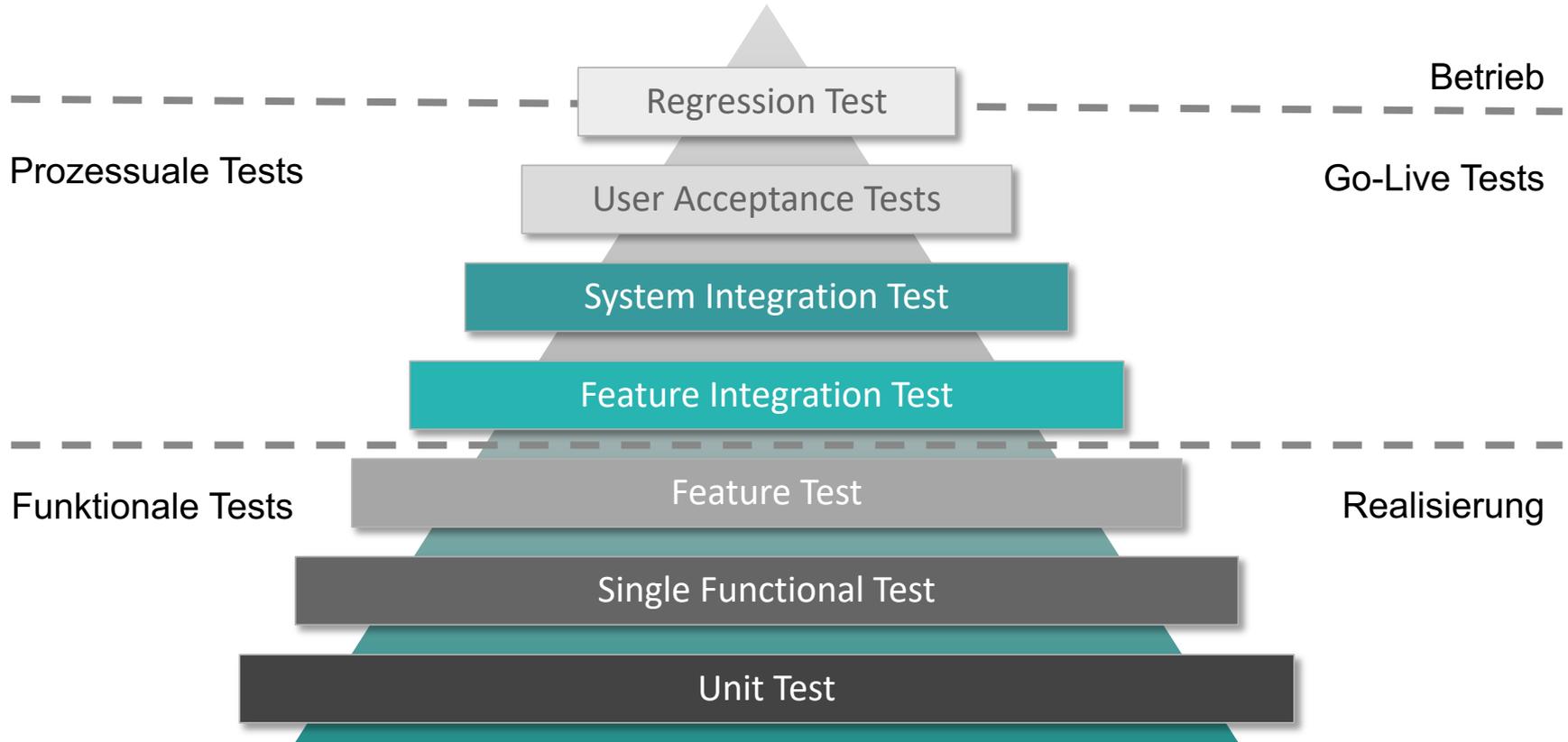
Fehlen einer stabilen  
Testbasis



Identifizierung und Management von Risiken  
zentral durch den Testmanager

- Testumfänge ermitteln
- Testfall-Erstellung
  - logische Testfälle aufbauen
  - konkrete Testfälle mit Testdaten ableiten
- Testfälle für erwartete und unerwartete Eingaben (Negativ-Tests)
- Testinfrastruktur zeitnah aufbauen, integrieren und prüfen!

# Teststufen - Übersicht

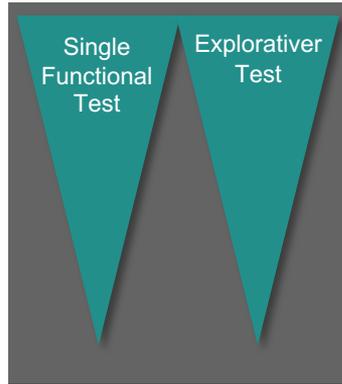


# Teststufen – Übersicht aus der Praxis

## Modulbasierter Test

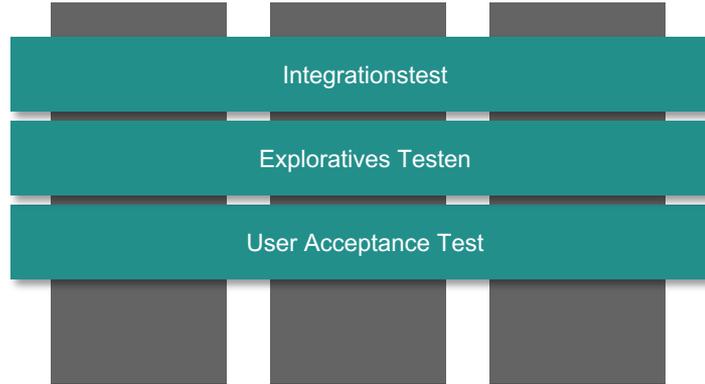


SAP Modul



SAP Modul

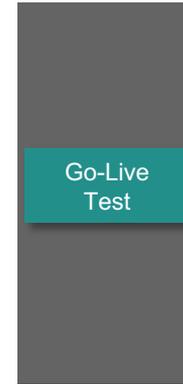
## Modulübergreifender Test



SAP Modul

SAP Modul

SAP Modul



Integrativ

Kundenbeispiel

Entwicklungssystem

Qualitätssicherungssystem

Produktionssystem

# Teststufen Übersicht

Kundenbeispiel

Teststufen	Wann	Wer testet	Testgegenstand	Funktion
Unit Test (UT)	Während Entwicklung	Entwickler	Durchgeführte Entwicklung / Konfiguration	Technische Korrektheit sicherstellen (nicht Formal) Bestätigung über Statuswechsel auf „zu Testen“
Single Functional Test (SFT)	Nach Umsetzung	Tester Projektorganisation	Anforderungen / Akzeptanzkriterien einer User Story	Abnahme der Funktionalität
Feature Test (FT)	Am Ende eines Entwicklungszyklus	Tester	Feature / Komponente / Modul	Softwarequalität des Features sicherstellen. Vorbereitung für weitere integrative Tests bzw. prozessuale Tests.
Feature Integration Test (FIT)	Nach dem Feature Test	Tester	Schnittstellen & Feature- übergreifende Prozesse	Integration zw. Features & Funktion der Einzelprozesse sicherstellen
System-Integrationstest (SIT)	Nach Ende der Umsetzung	Fachabteilung / Linienorganisation	End2End Prozesse inkl. Schnittstellen zu umgebenden Systemen	Integration zw. Systemen & Funktion der End2End Prozesse sicherstellen
Regressionstests (RT)	Start nach erstem Entwicklungszyklus sinnvoll	wenn möglich automatisiert	Fertiggestellte Prozesse / Applikationen	Sicherstellung, dass Anwendung/Prozess ordnungsgemäß funktioniert
User Acceptance Test (UAT)	zwischen SIT und Go-Live.	komplette Linienorganisation	Kernprozesse mit Vorgaben und exploratives Testen	Sicherstellen bzw. erhöhen der User Akzeptanz für das neue System


 Wichtiger Indikator, ob Testeingangsparameter für weitere Teststufen erreicht wurden.

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement

# Was können Testobjekte sein?

- Was wird getestet?

Anforderungen

Business  
Prozesse

Funktionen

Executables

Schnittstellen

Berechtigungen

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

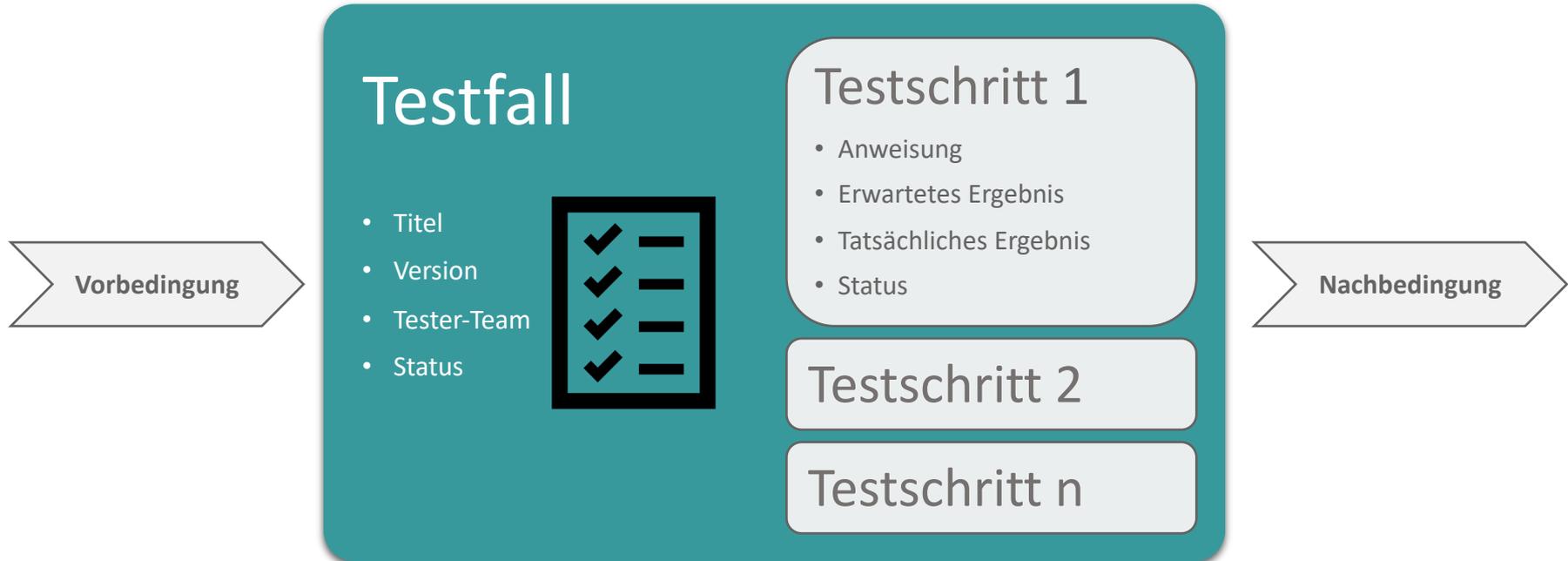
Testobjekte

Testfälle

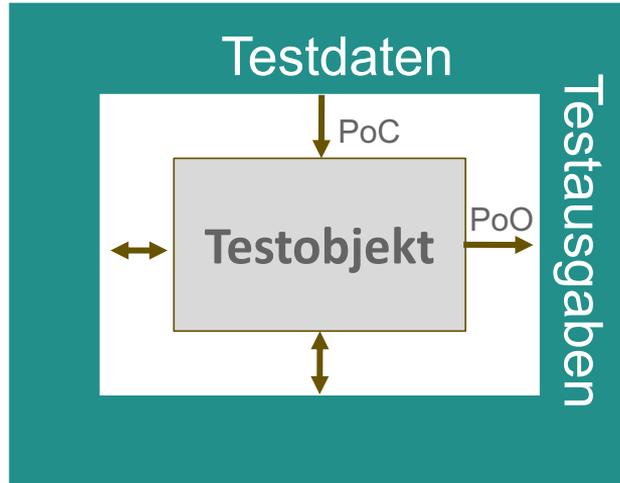
Testumfang identifizieren

Testprozess

Fehlermanagement

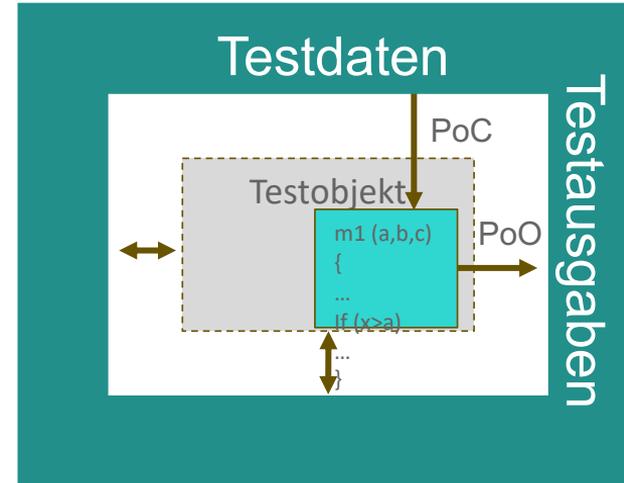


## Blackbox-Verfahren



PoC und PoO „außerhalb“  
des Testobjekts

## Whitebox-Verfahren



PoC und/oder PoO  
„innerhalb“ des Testobjekts

**Point of Control (PoC):** Schnittstelle, über die das Testobjekt mit Testdaten versorgt wird.

**Point of Observation (PoO):** Schnittstelle, an der die Reaktionen und Ausgaben des Testobjekts beobachtet und aufgezeichnet werden.

## Technische Tests



### Unit Test

- Wer: Entwickler
- Wann: Während der Umsetzung

### Statische Testmöglichkeiten

- Code Analyse (z.B. Virtual Forge)

### Beispiele:

- Doppelter Code
- Unerreichbarer Code
- Ineffiziente Algorithmen

## Fachliche Tests



- Wer: Anwender (Fachbereich)
- Was wird getestet:
  - Order-Formular (Funktion)
  - Order (Prozess)
  - Order2Cash (End2End Prozess)

## Manuelle Tests

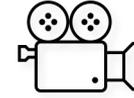


- Testfälle, die manuell ausführbar sind, benötigen für jeden Durchlauf personelle Ressourcen.
- Nachteile:
  - Ausführung ist aufwändig
- Vorteile:
  - Pflegeleichter
  - Kein Automatisierungstool notwendig
  - Kein zusätzlicher Schulungsbedarf für das Testautomatisierungstool



Geeignet sind Testfälle die selten wiederholt werden oder deren Anpassungshäufigkeit hoch ist

## Automatisierte Tests



- Testfälle, die automatisiert sind, können mit geringen Ressourcen wiederholt ausgeführt werden.
- Nachteile:
  - Testfallerstellung ist aufwändiger
  - Pflege ist aufwändiger
- Daher geeignete Auswahl notwendig!
- [Mehr Informationen im Kapitel Testautomatisierung](#)

## Funktionale Tests



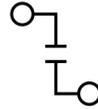
- Allg. Anwendungsfunktionen
  - Eingabe einer Order
  - Query aller offenen Orders
- Berechtigungen
  - Rolle ist berechtigt alle nötigen Schritte durchzuführen
  - Rolle darf keine weiteren Funktionalitäten verwenden
- Daten (z.B. bei Migrationen)
  - Stammdaten sind aufgebaut

## Nicht-Funktionale Tests



- Beispiele:
  - Effizienz
  - Performance (Last)
  - Ressourcenverbrauch (z.B. Speicherverbrauch)
  - Wartbarkeit
  - Stabilität
  - Prüfbarkeit
  - Übertragbarkeit
  - Wiederverwendbarkeit
  - Austauschbarkeit
  - Sicherheit → **Mehr Informationen im Kapitel Penetrationstest**

## Funktionale Tests



- Allg. Anwendungsfunktionen
  - Eingabe einer Order
  - Query aller offenen Orders
- Berechtigungen
  - Rolle ist berechtigt alle nötigen Schritte durchzuführen
  - Rolle darf keine weiteren Funktionalitäten verwenden
- Daten (z.B. bei Migrationen)
  - Stammdaten sind aufgebaut

## Prozessuale Tests



- Prozessabläufe (Sequenzen) werden getestet und Schnittstellen stehen im Fokus
- Funktionsfähig einzelner Transaktionen wird gegeben angenommen und ist obligatorisch
- Testdaten Management und Testdatenübergabe wichtige Grundlage
- Voraussetzung ist eine hohe Softwarequalität bei den Funktionstests, um eine akzeptable Durchlaufzeit zu erreichen

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement

## Dilemma:

„eigentlich“ soll alles getestet werden  
hierzu reichen Zeit und Budget nicht aus



Auswahl oder Priorisierung von Testfällen



**Welche Kernprozesse müssen laufen?**



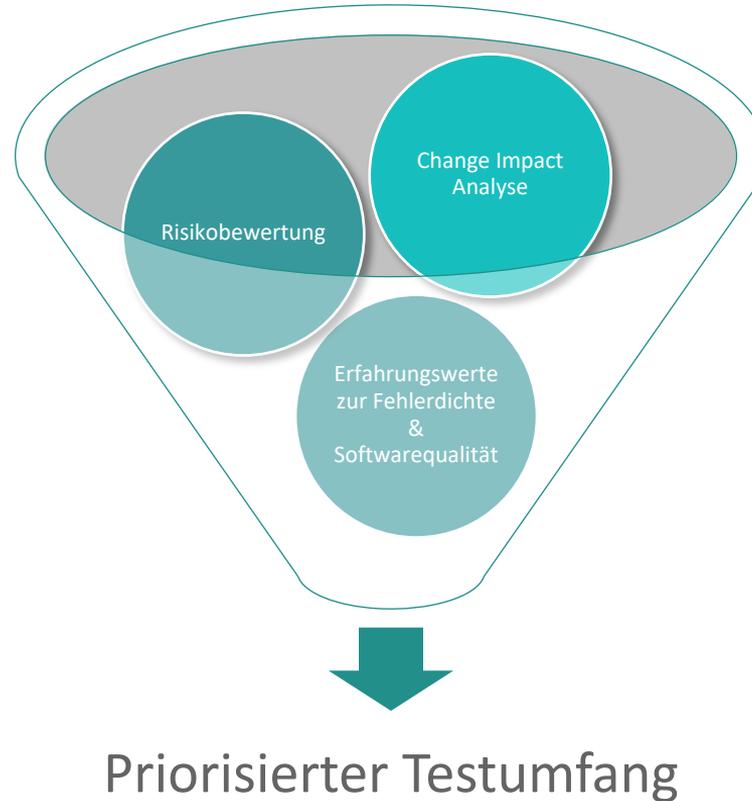
Allgemeine Prinzipien von  
Softwaretests:  
„Vollständiges Testen ist  
nicht möglich“

## Gesamttest

- Test aller Systemfunktionalitäten (?) (sinnvoll und erforderlich bei Einführungsprojekten)
- Hoher Testaufwand

## „Test der Änderung“

- Welche Auswirkungen hat die Änderung?
- Was muss ich im Umfeld auch Testen?



## Risiko

- Wahrscheinlichkeit des Auftretens eines Ereignisses, einer Gefahr, einer Bedrohung oder einer Situation und seiner unerwünschten Konsequenzen.
- Potenzielles Problem

## Produktisiko

- Mögliche Fehlerwirkungsbereiche in der Software oder im System sind bekannt, da sie Risiken für die Qualität des Produktes darstellen
  - Zum Beispiel:
    - Fehleranfällige Software
    - Schlechte Softwarecharakteristika (Funktionalität, Sicherheit etc.)
    - Schlechte Datenintegrität und –Qualität

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

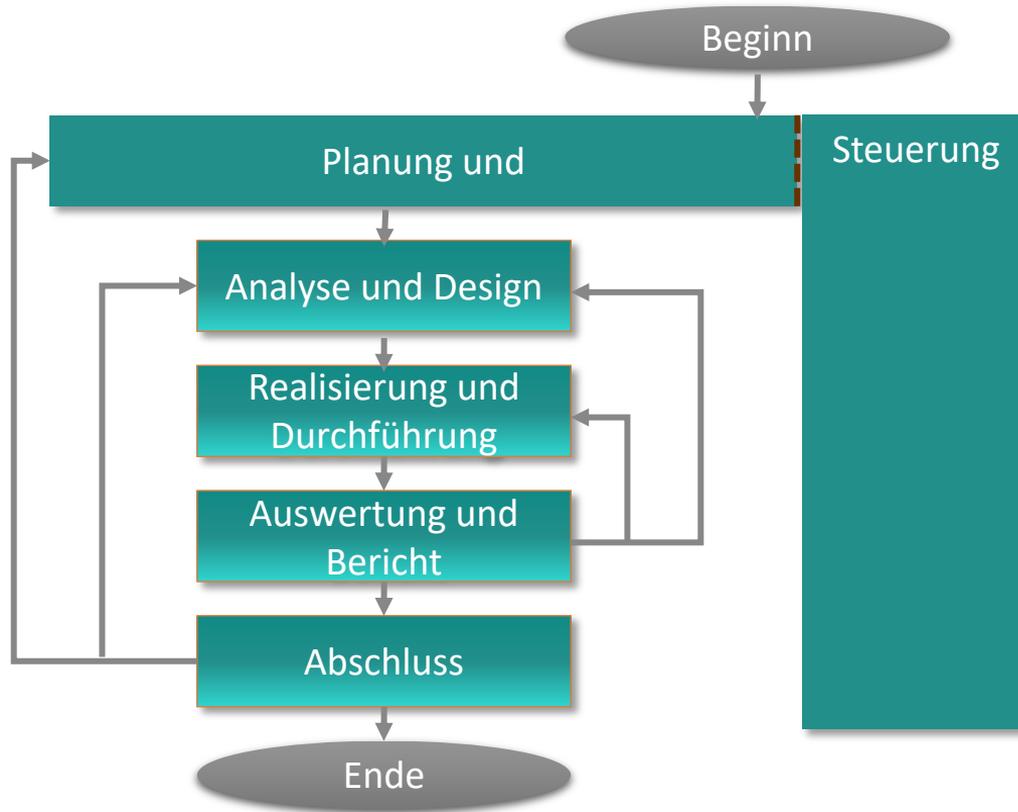
Testobjekte

Testfälle

Testumfang identifizieren

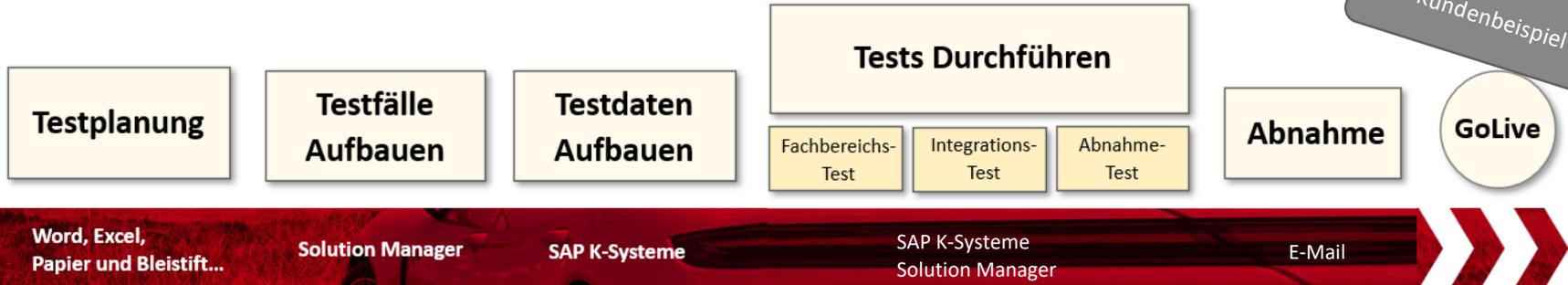
Testprozess

Fehlermanagement



# Test Management Prozess aus der Praxis

Kundenbeispiel



- Was will ich testen?
- Welche Daten habe ich?
- Muss ich Testdaten aufbauen?
- Wie ist meine Testabdeckung?



- Testfälle im SAP Solution Manager anlegen
- Auch Berechtigungen testen
- Negativtests nicht vergessen



- Im Testsystem vor Beginn der Tests Testdaten aufbauen
- Bei Schnittstellen: Prüfen, dass die Daten in den verschiedenen Systemen zueinander passen



- Tests im K-System durchführen
- Testergebnis im SolMan dokumentieren
- **Wichtig:** Auch i.O.-Fälle dokumentieren
- **Wichtig:** Für jeden Fehler einen Defect anlegen



- Keine Prio1-Fehler mehr vorhanden
- Alle Testfälle durchlaufen
- Abnahme an IT-PL per E-Mail, dass Transport ins P-System erfolgen kann

## Vormittag

### Das 1x1 des Testens

Grundlagen

Testplanung

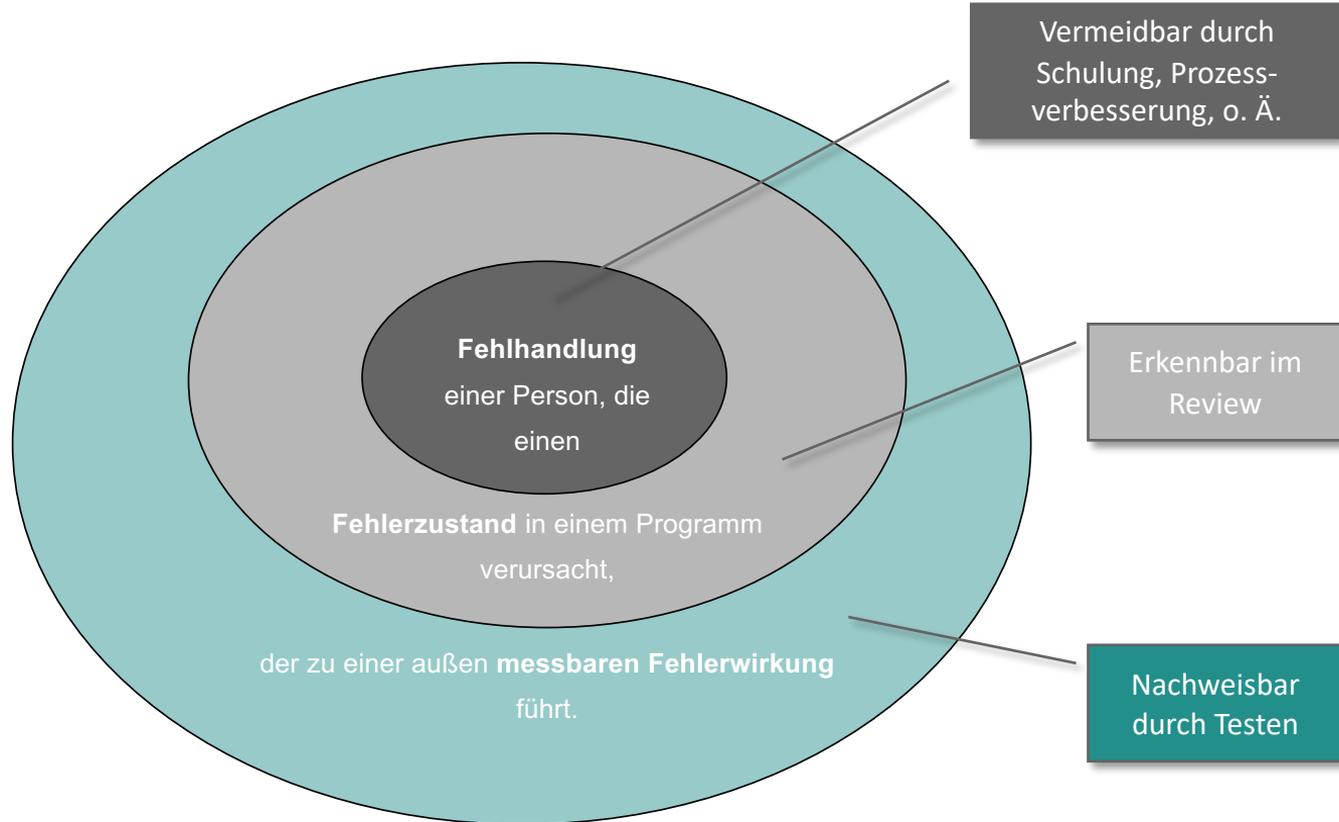
Testobjekte

Testfälle

Testumfang identifizieren

Testprozess

Fehlermanagement



**Nichtererkennung von Fehlern = hohes Risiko** beim Einsatz der Software

## Fehlerbegriff (Was gilt als Fehler oder Mangel?)

- **Fehler:** Nichterfüllung einer festgelegten Anforderung, Abweichung zwischen dem Ist-Verhalten und dem Soll-Verhalten (Spezifikation)
- **Mangel:** gestellte Anforderung oder berechnete Erwartung nicht angemessen erfüllt. Beeinträchtigung der Verwendbarkeit bei gleichzeitiger Erfüllung der Funktionalität oder die Nichterfüllung einer angemessenen Erwartung

## Fehlerwirkung

- Wird für den Test oder den Anwender bei dem Betrieb nach außen sichtbar
- Austreten der Fehlerwirkung und deren Ursache unterscheiden!
- Fehlerwirkung hat ihren Ursprung im **Fehlerzustand** (Bug)

## Fehlerzustand in Software lokalisieren

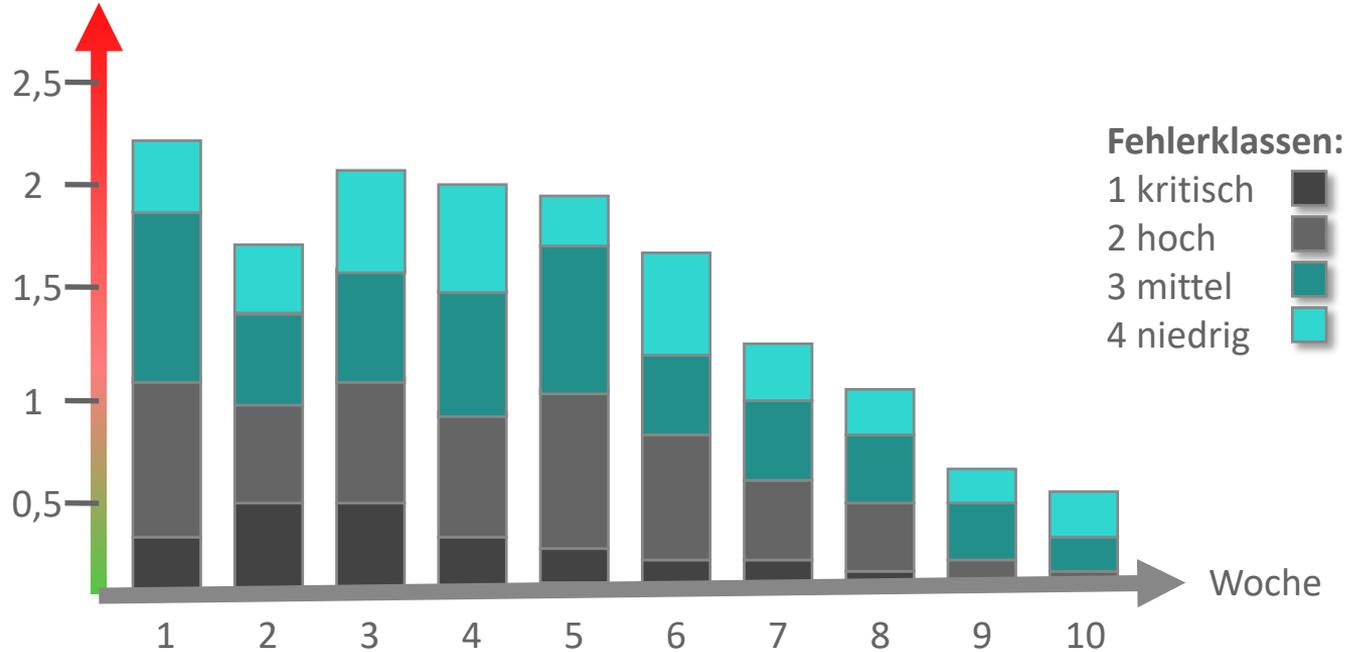
- Zuerst nur Wirkung bekannt, aber nicht genaue Stelle in SW, die Defekt darstellt
- Beheben des Defekts ist Aufgabe des Entwicklers
- Debugging = Fehlersuche & Fehlerbereinigung & Korrektur
- Behebung des Fehlerzustands führt zur Qualitätsverbesserung des Produkts

# Anlage von Fehlern - Ermittlung der Fehlerschwere

Priorität Fehler	Auswirkung fehlende Fehlerbehebung auf Go-Live	Beschreibung	Beispiel
<b>Sehr Hoch</b>	Kein Go-Live	<ul style="list-style-type: none"> <li>System / Testobjekt ist in dieser Form nicht einsetzbar, Test kann nicht fortgeführt werden</li> </ul>	<ul style="list-style-type: none"> <li>SAP-System nicht verfügbar</li> <li>Systemabsturz mit ggf. Datenverlust</li> </ul>
<b>Hoch</b>	Kein Go-Live	<ul style="list-style-type: none"> <li>System / Testobjekt ist nur mit großen Einschränkungen einsetzbar</li> <li>Test kann fortgeführt werden.</li> <li>Es liegt eine sicherheitsgefährdende Situation vor</li> <li>Wesentliche Funktionen sind betroffen und können nicht ausgeführt werden</li> </ul>	<ul style="list-style-type: none"> <li>Prozess nicht testbar</li> <li>Dump in der Transaktion</li> <li>Anforderung nicht beachtet oder falsch umgesetzt</li> </ul>
<b>Mittel</b>	Go-Live nach Risikoanalyse denkbar	<ul style="list-style-type: none"> <li>Wesentliche Funktionen sind betroffen, können aber mit Einschränkungen ausgeführt werden</li> <li>Die Durchführung nachfolgender Tests ist nicht betroffen und kann weiter erfolgen</li> </ul>	<ul style="list-style-type: none"> <li>Funktionale Abweichung bzw. Einschränkung;</li> <li>Anforderung fehlerhaft oder nur teilweise umgesetzt</li> <li>Drucken nicht möglich, aber PDF Erstellung funktioniert</li> <li>Ein Bestelltyp verursacht einen Fehler</li> </ul>
<b>Niedrig</b>	Go-Live unkritisch	<ul style="list-style-type: none"> <li>Die Funktionalität ist unwesentlich betroffen</li> <li>Es gibt keine signifikanten Auswirkungen auf die getestete Funktion</li> <li>Die Durchführung nachfolgender Tests ist nicht betroffen und kann weiter erfolgen</li> </ul>	<ul style="list-style-type: none"> <li>Typos nicht korrekt formatiert</li> <li>Mangel im Maskenlayout</li> <li>Exportspalten in falscher Reihenfolge</li> </ul>

# Fehlerfindungsrate als Indikator für das Testende

Fehler / Teststunde



# Defect Management - Prozessablauf aus der Praxis

Kundenbeispiel



Was nimmst du mit?

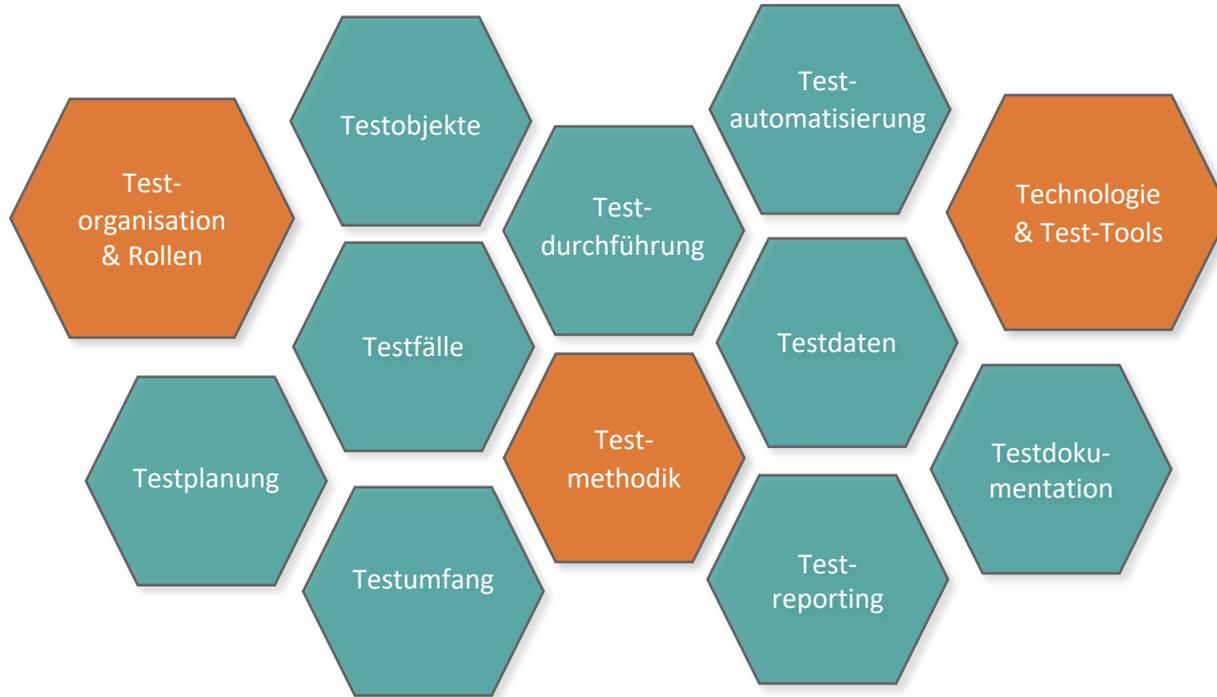


# Offene Fragen zum 1x1 des Testens?



# SAP Testmanagement Landkarte

## 12 Handlungsfelder

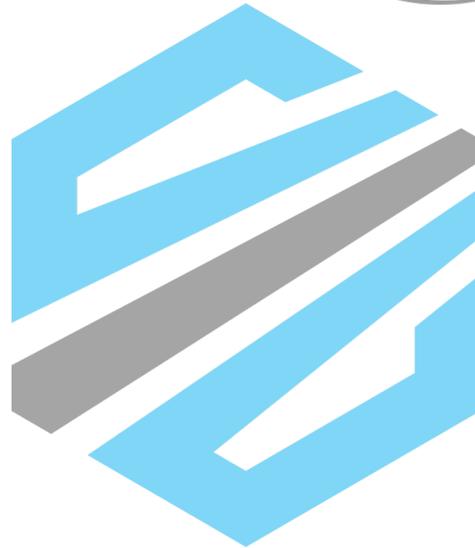


Kaffeepause





# Testautomatisierung



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Test-Reporting

## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

## Fragen, die dieses Kapitel beantwortet:

- Was sind automatisierte Tests?
- Wie funktionieren automatisierte Tests?
- Wann sind automatisierte Tests sinnvoll?
- Was sind die Elemente und wie wird vorgegangen (Best Practice)?

## Definition der Testautomatisierung:

Automatisierung des Prozesses zum Testen von Softwareanwendungen, um ihre Funktionalität, Leistung und Sicherheit mit minimaler menschlicher Intervention zu überprüfen.

## Allgemeine Bedeutung/Vorteile der Testautomatisierung:

Erhöht

- **Effizienz**
- **Genauigkeit**
- **Konsistenz**

der Softwaretests.

# Grundlagen: Ziele der Testautomatisierung

**Qualitätsverbesserung:** Verbesserung der Softwarequalität durch Erhöhung der Genauigkeit und Verringerung der menschlichen Fehler bei Tests.

**Schnellere Markteinführung:** Beschleunigung des Release-Zyklus durch Reduzierung der Zeit für Regressionstests.

**Konsistenz und Wiederholbarkeit:** Gewährleistung gleichbleibender Testausführungen und Ergebnisse bei wiederholten Testzyklen.

**Ressourcenoptimierung:** Freisetzung menschlicher Ressourcen für komplexere Testaufgaben und Analysearbeiten.

**Effizienzsteigerung:** Automatisierte Tests führen Routineaufgaben schneller aus und erhöhen die Testabdeckung.

**Frühzeitige Fehlererkennung:** Früheres Auffinden von Defekten im Entwicklungszyklus durch kontinuierliche Tests.

Qualitäts-  
verbes-  
serung

Schnelle  
Markt-  
einführung

Konsistenz  
und  
Wiederhol-  
barkeit

Ressourcen-  
optimierung

Effizienz-  
steigerung

Frühzeitige  
Fehler-  
erkennung

## Manueller Test

**Ausführung:** Durch Menschen von Hand

**Genauigkeit:** Risiko durch menschliche Fehler

**Zeitbedarf:** Zeitaufwendiger

**Verwendung:** Manuelle Tests sind für einmalige oder seltene Tests sinnvoll

**Skalierbarkeit:** schlechter verwendbar bei groß angelegten Tests über verschiedene Geräte

## Kosten

**Kosten:** Die Anfangsinvestitionen sind für automatisierte Tests höher, aber langfristig können sie kosteneffizienter sein, besonders bei regelmäßig wiederkehrenden Testreihen.

## Automatisierter Test

**Ausführung:** Skripte und Werkzeuge um den Test zu Automatisieren

**Genauigkeit:** sehr präzise, weil Skript- und Codebasis

**Zeitaufwand:** Schneller als manuell, weil sie automatisch ablaufen

**Verwendung:** Einfach replizierbar, daher für wiederholende Tests sinnvoll

**Skalierbarkeit:** gut verwendbar bei groß angelegten Tests über verschiedene Geräte

## Testskripte

- Automatisierte Anweisungen, die Testaktionen ausführen
- Simulieren Benutzerinteraktionen mit der Software
- Bestehen aus vordefinierten Eingaben, Ausführungsbedingungen und erwarteten Ergebnissen

## Testfälle

- Spezifische Szenarien zur Überprüfung von Funktionen der Software
- Enthalten detaillierte Schritte zur Durchführung eines Tests
- Prüfen Anforderungen und helfen bei der Identifizierung von Fehlverhalten

## Testpläne

- Sammlungen von Testfällen, die zusammen gruppiert sind
- Zielgerichtet für Regressionstests, Funktionstests, etc.
- Können selektiv basierend auf Testzielen ausgeführt werden

## Assertions

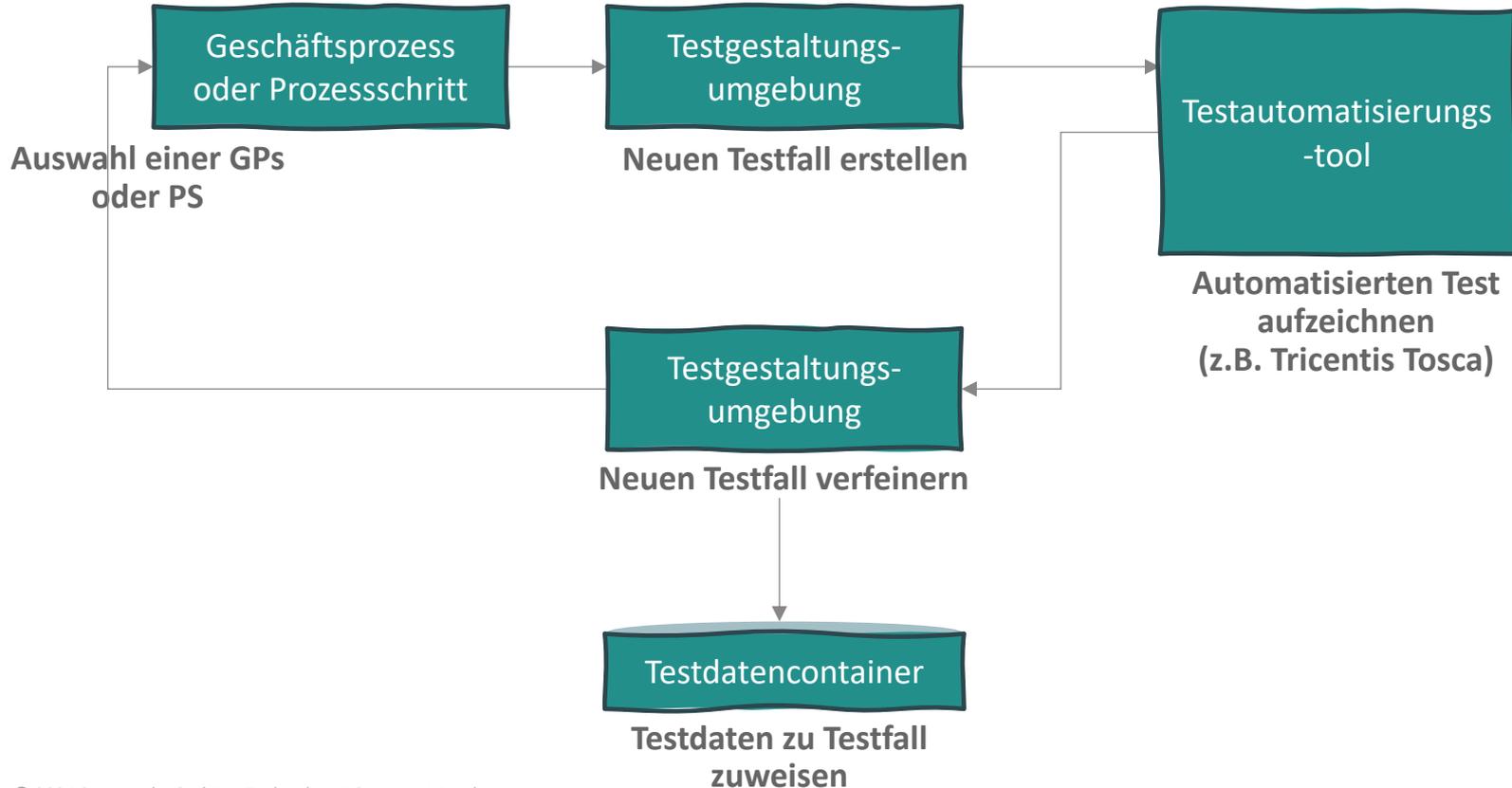
- Überprüfungen innerhalb eines Testskripts zur Bestätigung, dass das Softwareverhalten den Erwartungen entspricht

## Testdaten

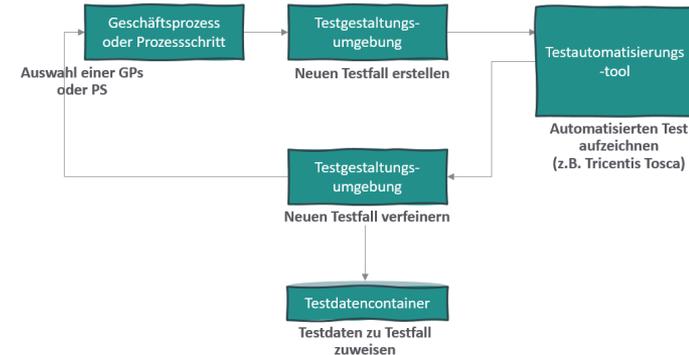
- Daten, die von Testskripten verwendet werden, um Testfälle unter verschiedenen Bedingungen auszuführen

## Testframework

- Software die Testskripte ausführt, Ergebnisse überwacht und Berichte generiert



- **Geschäftsprozess oder Prozessschritt:** Auswahl des zu testenden System- oder Softwarebereichs.
- **Testgestaltungsumgebung - Neuen Testfall erstellen:** Definition und Erstellung eines Testfalls basierend auf dem ausgewählten Prozess.
- **Neuen Testfall verfeinern:** Detaillierung des Testfalls zur Sicherstellung der vollständigen Testabdeckung.
- **Testdatencontainer:** Zuweisung von Testdaten zum Testfall, einschließlich aller relevanten Variablen und Benutzereingaben.
- **Testautomatisierungstool - Automatisierten Test aufzeichnen:** Einbindung in ein Automatisierungstool wie Tricentis Tosca oder CBTA, zur Ausführung und Dokumentation der Testergebnisse.
- **Ausführung der Geschäftstransaktion:** Automatisierte Durchführung des Tests in der realen Umgebung, inklusive SAP- und Nicht-SAP-Transaktionen.



**Nicht immer ist das automatisierte Testen sinnvoll.**

**Kriterien für die Auswahl:**

- **Viele Wiederholungen:** Tests, die häufig und regelmäßig durchgeführt werden müssen.
- **Hoher manueller Aufwand:** Zeitintensive Tests, die automatisiert werden können, um Ressourcen zu sparen.
- **Stabilität:** Tests von Features, die sich nicht häufig ändern.
- **Kritische Geschäftsfunktionen:** Tests, die kritische Komponenten der Anwendung betreffen.

**Beispiele für sehr gut automatisierbare Testfälle:**

Regressions-  
tests

Belastungs-  
Tests /  
Performance  
Tests

Smoke  
Tests

# Grundlagen: Wann automatisiert Testen?

## Viele Wiederholungen

Tests, die häufig und regelmäßig durchgeführt werden müssen, sind ideale Kandidaten für die Automatisierung. Automatisierung hilft, menschliche Fehler zu reduzieren, die bei monotonen, wiederholten Aufgaben auftreten können. Außerdem spart es Zeit und sorgt für Konsistenz über die Testdurchläufe hinweg.

## hoher man. Aufwand

Zeitintensive Tests können sehr gut automatisiert werden, um Ressourcen zu sparen. Nicht nur Zeit, sondern auch die Kosten für die manuelle Ausführung werden reduziert. Sobald der Test automatisiert ist, kann er schnell und häufig ohne zusätzlichen Aufwand ausgeführt werden.

## Stabilität

Tests von Features, die sich nicht häufig ändern, sind ebenfalls gute Kandidaten für Automatisierung. Da diese Tests seltener aktualisiert werden müssen, bietet die anfängliche Investition in die Automatisierung langfristige Vorteile.

## Kritische Geschäftsfunktionen

Automatisiertes Testen ist bei kritischen Geschäftsfunktionen besonders sinnvoll, weil diese Funktionen oft das Rückgrat einer Anwendung darstellen und ihre korrekte Funktionsweise entscheidend ist. Automatisierte Tests können rund um die Uhr ausgeführt werden, um sicherzustellen, dass diese Funktionen jederzeit wie erwartet funktionieren. Darüber hinaus können sie dabei helfen, Regressionen schnell zu identifizieren, was bei manuellen Tests möglicherweise nicht mit der gleichen Geschwindigkeit möglich wäre.

Welche Kriterien könnten noch relevant sein für die Wahl automatisiert zu testen?  
Sammeln Sie weitere Kriterien in der Gruppe. Welche Geschäftsaspekte und welche technischen Aspekte könnten hineinspielen?

## 1. Geschäftsaspekte:

1. Stabilität des Prozesses in Bezug auf Geschäfts- und technologische Aspekte.
2. Vorteilhaftigkeit der Testausführung, zum Beispiel während Wartungszyklen.
3. Möglichkeit, Testergebnisse ohne Mehrdeutigkeiten zu verifizieren.
4. Geschäftskritikalität und Priorität des Prozesses.
5. Häufigkeit der Durchführung im täglichen Geschäftsbetrieb.

## 2. Technische Aspekte:

1. Aufwand für die Automatisierung des Testfalls.
2. Wiederverwendbarkeit der zu automatisierenden Transaktionen in anderen Tests.
3. Involvierung von Drittsystemen.
4. Möglichkeit zur mehrmaligen Ausführung automatisierter Tests.
5. Notwendigkeit für manuelle Add-On-Tests zur Unterstützung automatisierter Tests.

Eine Möglichkeit, um sich final für eine Testautomatisierung und Testautomatisierungstools zu entscheiden ist die Verwendung einer Bewertung, in der die Kriterien durchgegangen werden:

Nr.	Frage	Hauptkriterium	Erklärung/ Beispiel	Antwort (J/N)	Zusatzkommentar
1	Ist der Aufwand der Testfallautomatisierung niedrig?		<b>Erklärung:</b> Je geringer der Aufwand ist, desto höher die Benefits		
2	Können die zu automatisierenden Transaktionen, wie sie sind, wiederverwendet werden oder mit nur kleinen Anpassungen		<b>Erklärung:</b> Die Wiederverwendung von Tests die bereits aufgezeichnet wurden (mit unterschiedlichen Daten) minimiert den Aufwand der Automatisierung von zukünftigen Tests <b>Beispiel:</b> <ul style="list-style-type: none"><li>• Bestellung erstellen</li><li>• Wareneingang anlegen</li></ul>		

## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

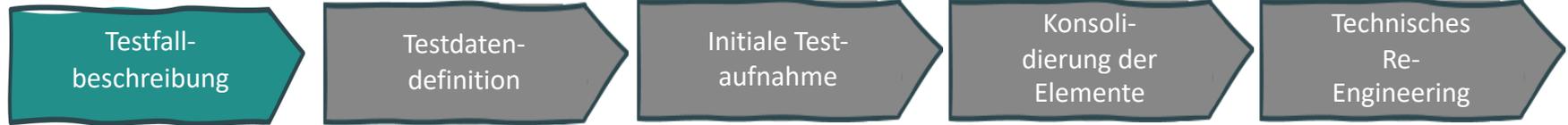
Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

Methodische Vorgehensweise ist in 5 Schritten aufgeteilt:



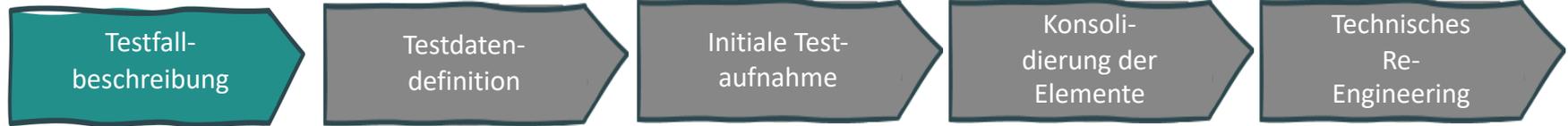


Eine Testfallbeschreibung ist ein Dokument, das einen einzelnen Testfall detailliert darstellt. Sie umfasst alle erforderlichen Schritte, um eine bestimmte Funktion eines Softwareprodukts zu überprüfen.

- Definiert klare Testkriterien und Ziele
- Ermöglicht, Zweck, Visionen und die schrittweise Abfolge der Elemente eines Testfalls zu verstehen
- Ermöglicht eine konsistente Testdurchführung
- Dient als Basis für Testautomatisierung

### Best Practice Tipp:

Jeden Testfall mit einer eindeutigen Identifikationsnummer versehen und neben der Beschreibung auch Vorbedingungen, erwartete Ergebnisse und Nachbedingungen dokumentieren, um Transparenz und Nachvollziehbarkeit zu gewährleisten.



## Bestandteile einer Testfallbeschreibung:

- 1. Testfall-ID:** Eindeutige Kennzeichnung eines Testfalls.
- 2. Beschreibung:** Klare und präzise Beschreibung des Testziels.
- 3. Vorbedingungen:** Zustand des Systems vor Testdurchführung.
- 4. Testdaten:** Daten, die für die Durchführung des Tests erforderlich sind.
- 5. Testvorgang:** Schritt-für-Schritt-Anweisungen zur Durchführung des Tests.
- 6. Erwartetes Ergebnis:** Definierte Ausgabe oder Zustand, der nach Testausführung erwartet wird.
- 7. Nachbedingungen:** Zustand des Systems nach Testdurchführung.
- 8. Tatsächliches Ergebnis:** Das während des Tests tatsächlich erhaltene Ergebnis.

# Testautomatisierung

## Schrittweises Vorgehen: Testdatendefinition



Testdaten sind Daten, die während eines Tests verwendet werden, um sowohl auch normale als auch außergewöhnliche Betriebsbedingungen zu definieren. Sie ermöglichen es, das Verhalten einer Anwendung unter verschiedenen Datenkombinationen zu überprüfen.

### Inhalt Testdatendefinition:

- Definition von Testdaten inkl. verschiedener Testdatenvarianten
- Relevante Testdaten als verschiedene Testdatenvarianten müssen als Eingabe für manuelle und automatische Tests eingerichtet werden.

(mehr Informationen zu Testdaten im Kapitel *Testdaten*)

### Best Practice Tipp:

Testdaten sollten repräsentativ, datenschutzkonform und in einem Format vorliegen, das eine einfache Wiederverwendung und Anpassung ermöglicht, um unterschiedliche Test-Szenarien effizient abzudecken.

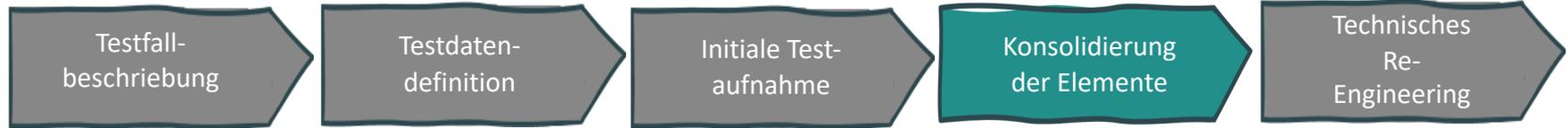
# Testautomatisierung

## Schrittweises Vorgehen: Testdatendefinition



- Initiale Aufnahme mit Erfassung der Bearbeitung
- Ermöglicht die Aufzeichnung detaillierter Bearbeitungsschritte (Pop-ups, Nachrichten, Datentransfer usw.).

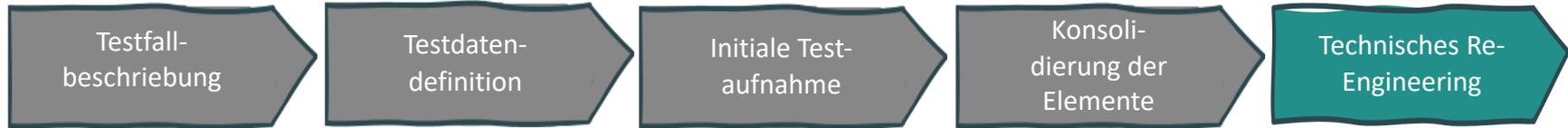
**Best Practice Tipp:**  
Einsatz von Tools für die Testautomatisierung, die eine visuelle Aufzeichnung von Benutzerinteraktionen unterstützen, um die Erstellung von Testskripten zu beschleunigen und zu vereinfachen.  
**z. B. Tricentis Tosca**



- Konsolidierung aller Elemente, um zusammenzupassen
- Ermöglicht die Konsolidierung und Integration von Erfahrungen während der Einrichtung.

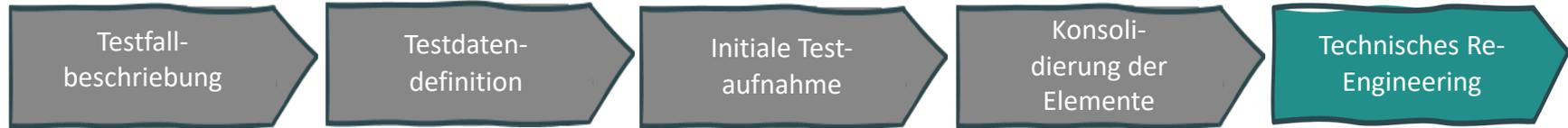
### Best Practice Tipp:

Erstellung eines einheitlichen Repositorys für Testskripte, -bibliotheken und -daten, um die Wiederverwendbarkeit zu erhöhen und Redundanzen zu reduzieren. Dafür gibt es ebenfalls Toolunterstützung.



- Technische Überarbeitung, um die Nutzung von Lego-Bausteinelementen zu ermöglichen
- Überarbeitung der Testautomatisierungsaufzeichnung, um einzelne Elemente zu verketteten und Lego-Bausteinelemente für eine effiziente Wiederverwendung einzurichten.
  - Beispiele: Entfernung redundanter Codezeilen, Nutzung von Design Patterns, Kapselung von häufig verwendeten Funktionen in Hilfsklassen.
- Optimierung der Testumgebung:
  - Anpassen der Testumgebung, um sie robuster gegenüber Veränderungen in der zu testenden Software zu machen.

**Best Practice Tipp:**  
Periodische Überprüfung und Anpassung der Testautomatisierungs-Skripte, um mit der weiterentwickelten Anwendungsarchitektur Schritt zu halten und die Testeffizienz zu maximieren.



Weitere Überarbeitungsmöglichkeiten:

- Aktualisierung der Testwerkzeuge
- Verbesserung der Testdaten
- Performance Tuning (Analyse und Optimierung der Performance)
- Verbessern der Integration der Testautomatisierung in den Gesamtentwicklungsprozess
- Sicherstellung der Testabdeckung

## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

- Der Testdatencontainer erlaubt die Hinterlegung mehrerer Testdatensätze in ein einheitliches Format in einer tabellenartigen Abbildung (wie eine Excel-Liste)
- Er lässt sich in Test Steps über Testdatensätze mit einem Test Step verbinden
- Die Verbindung erfolgt über den zu erstellenden Testplan und das Testpaket auf indirektem Wege
- Alle Testdatensätze sind so mit dem Test Step verknüpft und kein einzelner Testdatensatz
- Der Testdatencontainer liefert über das Test Automation Framework (TAF) auch Daten für die Testautomatisierung

**Leitsatz: Welche Information muss an welcher Stelle eingetragen werden, um den Test durchzuführen?**

**Nachteil:** Testdatencontainer müssen regelmäßig aktualisiert werden, damit sie Anwendbar bleiben und die Wahrheit zeigen, wenn sie zeigen immer die Vergangenheit an.



Das Wort

Testdatencontainer ist stark Solution Manager geprägt – aber prinzipiell ist es allgemein verwendbar

**Testdatencontainer AGS\_GBC\_TD\_ODATA\_BPMN\_AUT (1) anzeigen**

Menü  Zurück Beenden Abbrechen System  Anzeigen <-> Ändern Anderes Objekt bearbeiten Dokumentation Varianten herunterladen Versions

Testdatencontainer  Version  Zielsystem

Titel  Instanz

SAP Solution Manager

Parameter Varianten Attribute

Parameter	Beschreibung	Parameterwert	Bezug des Parameters	Zielsystem	ABAP ...	Länge DezSt...	Gruppe
AGSGBC_GRAPH	GBC: Grafische Tabelle	<VALUE>	AGSGBC_T_GRAPH				
AGSGBC_ENTI...	GBC: Entitätstabelle	<VALUE>	AGSGBC_T_ENTITIES				
AGSGBC_ENT_...	GBC: Tabelle Entitätslinks	<VALUE>	AGSGBC_T_ENT_LINK				
AGSGBC_ENT_...	GBC: Tabelle Entitätslayout	<VALUE>	AGSGBC_T_ENT_LAYOUT				
AGSGBC_ENT_...	GBC: Sprachabhängige Texta...	<VALUE>	AGSGBC_T_ENT_TEXT				
AGSGBC_ENT_...	GBC: Entitätensammlung	<VALUE>	AGSGBC_T_ENT_COLL				
AGSGBC_ODAT...	OData-Testdatenstruktur	<VALUE>	AGSGBC_S_ODATA_TEST_DATA				
AGSGBC_ROLES	GBC: Rollentabelle	<VALUE>	AGSGBC_T_ROLES				
AGSGBC_ROLE...	GBC: Rollentexttabelle	<VALUE>	AGSGBC_T_ROLES_TEXT				
AGSGBC_DATA...	GBC: Datenspeichertabelle	<VALUE>	AGSGBC_T_DATA_STORE				
AGSGBC_DATA...	GBC: Datenspeichertabelle	<VALUE>	AGSGBC_T_DATA_STORE_TEXT				
AGSGBC_DATA...	GBC: Datenobjektabelle	<VALUE>	AGSGBC_T_DATA_OBJECT				
AGSGBC_DATA...	GBC: Datenobjektabelle	<VALUE>	AGSGBC_T_DATA_OBJECT_TEXT				
AGSGBC_FREE...	GBC: Texttabelle für freie Kom...	<VALUE>	AGSGBC_T_FREE_COMP_TEXT				
AGSGBC_FREE...	GBC: Tabelle für freie Kompon...	<VALUE>	AGSGBC_T_FREE_COMP				





### Change Test Configuration ZMYCONFIGURATION

External Test

Test Configuration: ZMYCONFIGURATION

Title: Sample test configuration

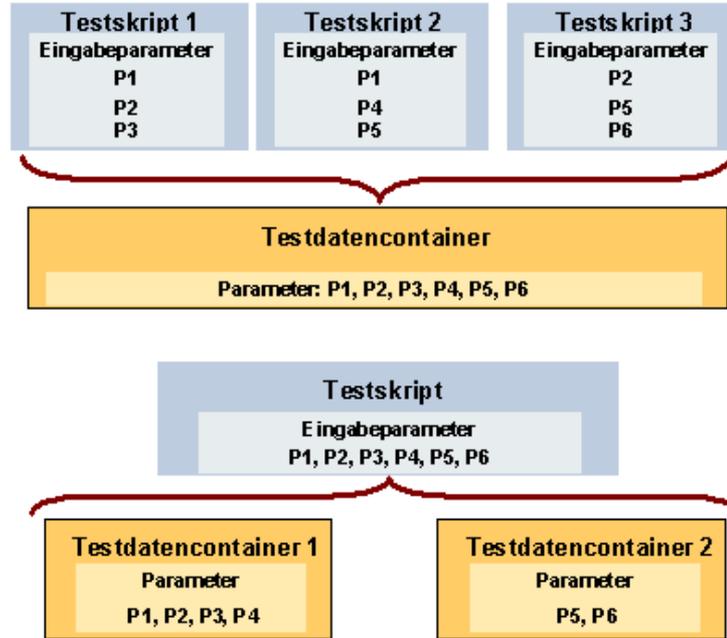
SAP Solution Manager: Landscape SOLMAN\_BP -> Branch MAINTENANCE -> Role: Development System

Configuration | Variants | Test Data | External Test | Attributes

Mode of external data

Execute	Variant	Description	Attachment	SALES_DOC_TYPE	SALES_ORGANIZATION	DISTRIBUTION_CHANNEL	DIVISION
<input checked="" type="checkbox"/>	ECATTDEFAULT			OR	1000	10	00
<input checked="" type="checkbox"/>	VARIANT1			OR	0006	10	08

Hier ist die Liste zu sehen





## Testdatencontainer

### Parameter

Parameter	Wert	Referenz	System	ABAP-Typ	Länge
Flugges.	LH	S_CARR_ID			
Datum				D	8

### Varianten

Variante	Beschreibung	Flugges.	Datum
ECATTDEFAULT		LH	
Var1	Test 1	AA	01.04.2002
Var2	Test 2	LH	01.04.2002

## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

## Manuelles Testen:

Hierbei werden die Testfälle von einem Menschen durchgeführt. Testdatencontainer werden verwendet, um konsistente Testbedingungen zu schaffen, indem man vordefinierte Daten für Testläufe bereithält.

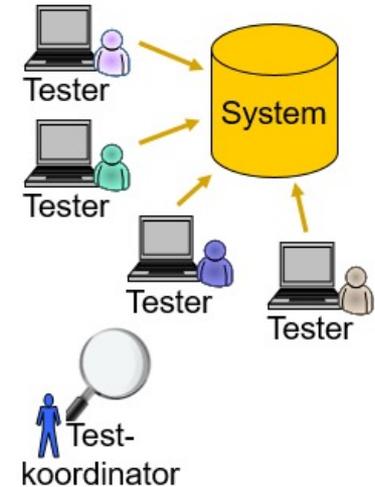
Bei manuellen Tests mit dem Testdatencontainer wird im Solution Manager die MyExecution-App verwendet.

## Vorteile bei Nutzung des Testdatencontainers:

- Schneller und einfacher Zugriff auf Informationen
- Mehr Fokus auf Ausführung und Beobachtung der Testergebnisse
- Zeitersparnis (keine Vorbereitung und Konfiguration von Testdaten)

## Einsatzgebiete manuelles Testen:

- Exploratives Testen
- Usability Tests
- Regressionstests
- ...



## Automatisches Testen:

Hierbei werden die Testfälle von einem Computer ausgeführt.

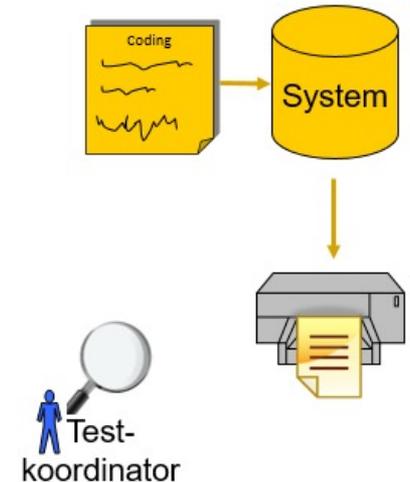
Der einzutragende Wert muss zuvor für den Computer festgelegt werden.

## Vorteile bei Nutzung des Testdatencontainers im automatisierten Testen:

- **Konsistenz:** Gleiche Testdaten für wiederholte Testszenarien sorgen für standardisierte Ergebnisse.
- **Einsatz von Testdaten:** Automatisierte Skripte nutzen Testdatencontainer für unterschiedliche Teststufen und Datenvariationen.

## Einsatzmöglichkeiten automatisiertes Testen:

- Datenwiederverwendung
- Testskript-Effizienz
- Paralleltests
- ...



Manuelle und automatische Tests können beide den Testdatencontainer verwenden

## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

**Definition:** Testparameter ermöglichen es, Testfälle dynamisch und flexibel zu gestalten, indem festgelegte Parameterwerte durch variable Inhalte ersetzt werden. Dies ist besonders hilfreich, um Testläufe an unterschiedliche Testumgebungen und Szenarien anzupassen, ohne jeden Testfall manuell bearbeiten zu müssen.

**Vorgehensweise:** Zuerst werden die Testparameter definiert. Das bedeutet, dass Sie für jede Variable einen Namen und gegebenenfalls einen Standardwert festlegen. Im Kontext von SAP könnten dies Kundennummern, Produktdaten oder spezifische Konfigurationseinstellungen sein.

## Was wird gemacht?

- Ersetzen von Parametern in Testfällen durch andere Parameter
- Im Beispiel wird Plant mit WDF zugeordnet
- In der My Test Execution wird beim öffnen des Test-Steps live der Wert in der Klammer mit WDF ersetzt

Die Vorgehensweise im Solution Manager wird in den folgenden Folien demonstriert



- Der SAP Solution Manager bietet grundsätzlich die zwei Funktionen zur Unterstützung des Themas Testdatenmanagement an:
  - Testparameter
  - Testdatencontainer
- Beider Ansätze sind nicht dazu geeignet alleine explizite Testdaten auf einfachem Wege Testern zur Verfügung zu stellen:
  - Testdatencontainer bietet nur eingeschränkt die Auswahl eines Testdatensatzes an und sind nur indirekt mit dem Testfall verknüpft
  - nicht für die Testautoautomatisierung wiederverwendbar
  - Eine Kombination aus beiden Lösungen könnten sowohl den Tester selbst unterstützen und ebenfalls für die Testautomatisierung genutzt werden





Voraussetzung  
Ein Test-Step pro Variante und Testpaket

Testdatencontainer wird ausgelesen und Test-Step Parameter aufgrund des Testdatencontainers gefüllt.  
Manuell durch Report ausgelöst.

Ohne technische Unterstützung durch einen Report ist eine Kombination der beiden Funktionalitäten unrealistisch wenn:

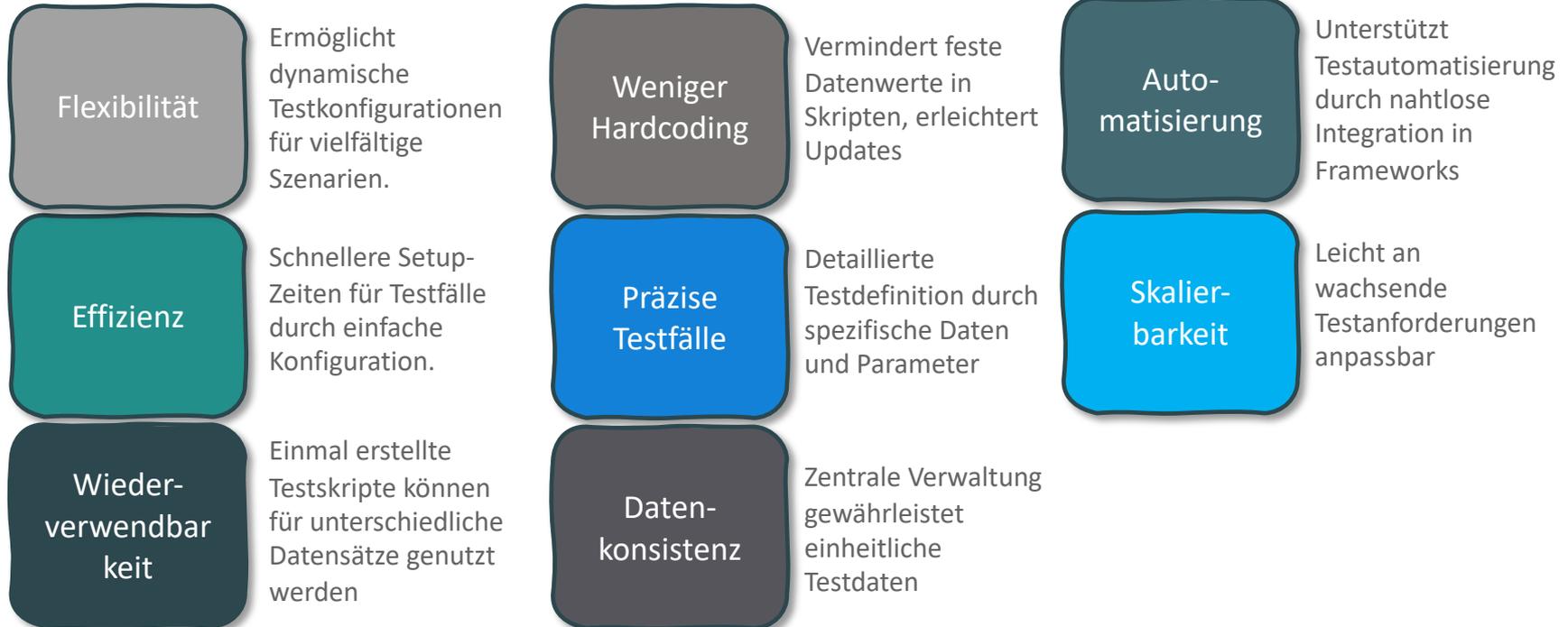
Inhalte von Testplänen flexibel sein müssen

Daten sich kontinuierlich ändern

Umfang und Testdaten dynamisch sind

# Testdatencontainer und Testparameter: Vorteile

Die Kombination aus Testdatencontainer und Testparametern bietet eine Reihe von Vorteilen, die besonders in komplexen Testumgebungen wie denen von SAP relevant sind. Hier sind einige der Schlüsselvorteile:



## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

## Folgende Vorteile entstehen durch eine Testautomatisierung (nach ISTQB):

**Wiederholbarkeit:** Automatisierte Tests können beliebig oft wiederholt werden, was bei manuellen Tests zeitaufwendig wäre

**Zuverlässigkeit:** Sie eliminieren das menschliche Fehlerpotenzial bei wiederholten Testdurchläufen

**Schnelligkeit:** Tests können schnell ausgeführt werden, was die Testzyklen verkürzt

**Kosteneffizienz:** Langfristige Kostenersparnis durch wiederholte Nutzung der automatisierten Tests

Wiederhol-  
barkeit

Zuverlässig-  
keit

Schnelligkeit

Kosten-  
effizienz

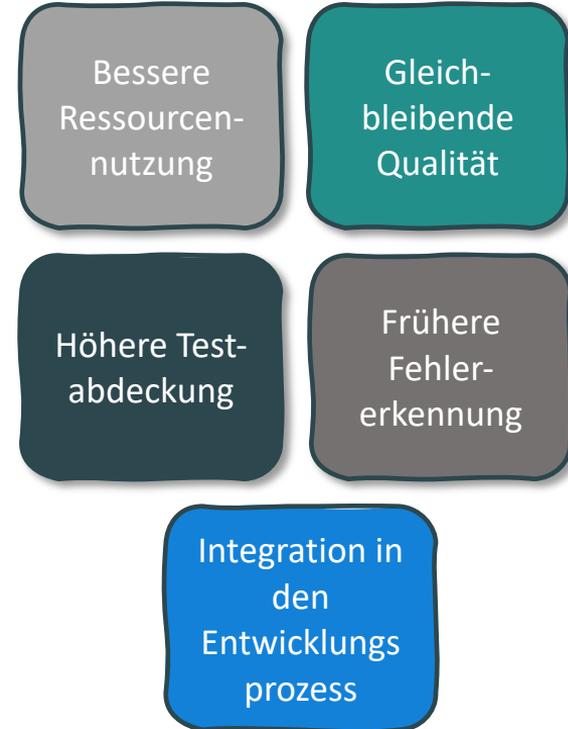
**Bessere Ressourcennutzung:** Tester können sich komplexeren Aufgaben widmen, während automatisierte Tests laufen

**Gleichbleibende Qualität:** Einheitliche Testdurchführung sichert konsistente Qualität

**Höhere Testabdeckung:** Automatisierung ermöglicht es, mehr Tests in kürzerer Zeit durchzuführen

**Frühere Fehlererkennung:** Fehler können durch kontinuierliche Tests schneller identifiziert werden

**Integration in den Entwicklungsprozess:** Automatisierte Tests können leicht in Continuous Integration und Continuous Delivery Prozesse integriert werden



## Vormittag

### Testautomatisierung

Grundlagen

Schrittweises Vorgehen

Testdatencontainer

Verwendung für manuelles und automatisiertes Testen

Testdatencontainer und Testparameter in Kombination

Vorteile

Allgemeine Best Practice Tipps

## Planung und Strategie

- **Zielsetzung und Scope definieren:**  
Klare Ziele für die Testautomatisierung festlegen, einschließlich des Umfangs, welcher Teil der Anwendung automatisiert werden soll.
- **Auswahl der richtigen Tools:**  
Basierend auf der Technologie der Anwendung, Budget, Ressourcen und vorhandenen Skills das passende Testwerkzeug auswählen.
- **Aufbau eines skalierbaren Frameworks:**  
Entwicklung eines modularen und wiederverwendbaren Frameworks, das mit der Anwendung mitwachsen kann.
- **Priorisierung der Testfälle für Automatisierung:**  
Kritische Geschäftsfunktionen und häufig verwendete Features zu priorisieren

## Entwicklung und Wartung

- **Wartbare und lesbare Skripte schreiben:**  
Code-Konventionen und Best Practices befolgen, um die Wartung und Lesbarkeit der Testskripte zu erleichtern.
- **Verwendung von Page Objects und anderen Designmustern:**  
Reduzierung von Wiederholungen und Erleichterung der Wartung durch den Einsatz von Designmustern wie Page Objects.
- **Regelmäßige Überprüfung und Aktualisierung der Testfälle:**  
Sicherstellen, dass die Testskripte aktuell bleiben und mit den Änderungen der Anwendung Schritt halten.
- **Datenmanagement:**  
Verwendung von Testdatenmanagementstrategien, um mit variablen Datensätzen effektiv zu testen.

## Integration und Berichterstattung

- **Continuous Integration (CI) einbinden:**  
Testautomatisierung in CI/CD-Pipelines integrieren, um Tests bei jedem Commit auszuführen.
- **Flaky Tests minimieren:**  
Instabile Tests identifizieren und korrigieren, um die Zuverlässigkeit der Testergebnisse zu gewährleisten.
- **Ausführliche Berichte und Dashboards nutzen:**  
Einsatz von Tools zur Generierung detaillierter Berichte und Dashboards, um Testergebnisse transparent und nachvollziehbar zu machen.
- **Feedback-Schleifen etablieren:**  
Regelmäßige Reviews der Testergebnisse und Anpassungen der Teststrategien basierend auf Feedback.

## Beispiele für aktuell häufig genutzte Testautomatisierungstools:

- Tricentis Test Automation & Tricentis Tosca
- Leapwork
- Suxxesso
- Worksoft
- Cypress



suxxesso

WORKSOFT®



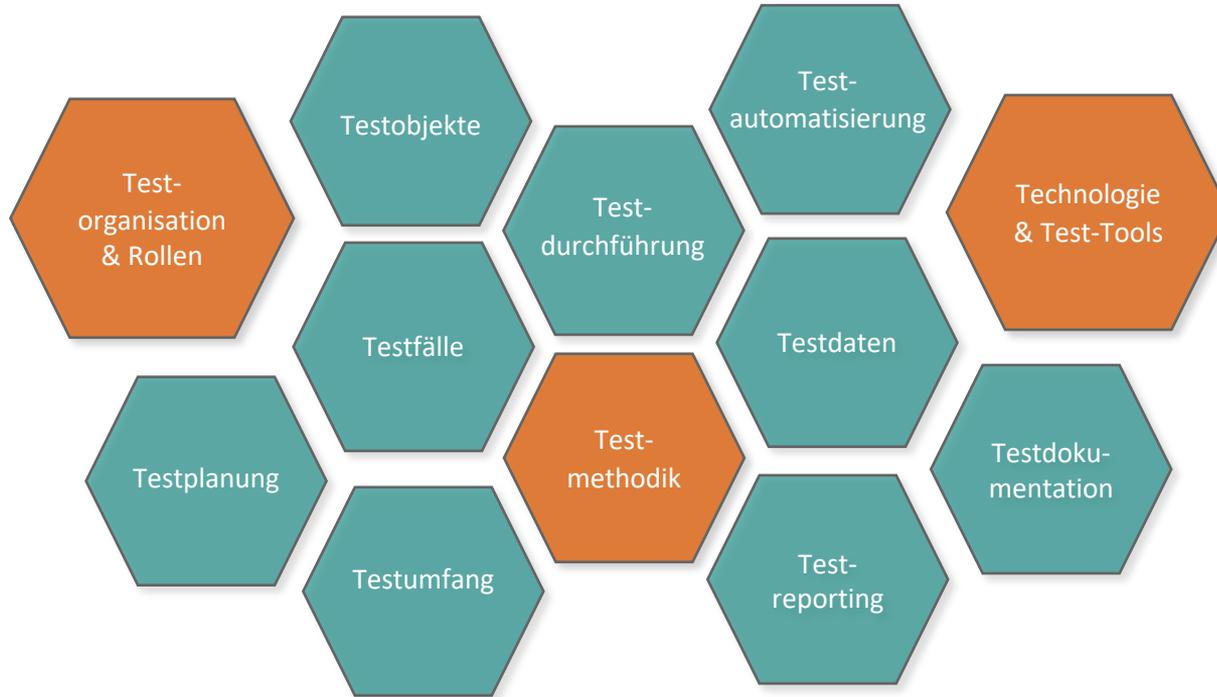
Diese Tools werden im Kapitel „Tools“ genauer vorgestellt

# Offene Fragen zum Thema Testautomatisierung?



# SAP Testmanagement Landkarte

## 12 Handlungsfelder



Mittagspause



# Testen im agilen Umfeld



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Testreporting

## Nachmittag

### Testen im agilen Umfeld

Agilität

Neue und alte Rollen

Agiles Testen

Hindernisse & Lösungen

## Nachmittag

Testen im agilen Umfeld

Agilität

Neue und alte Rollen

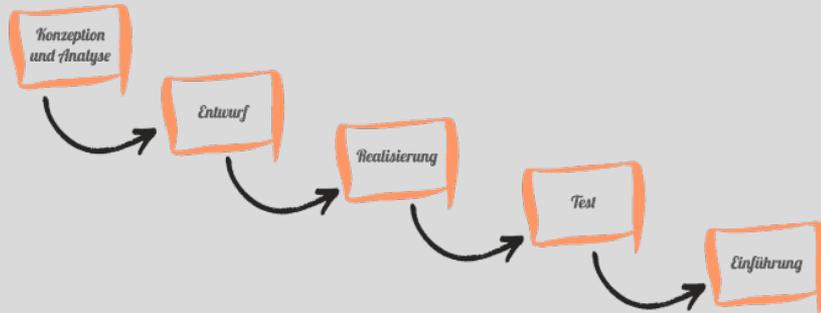
Agiles Testen

Hindernisse & Lösungen

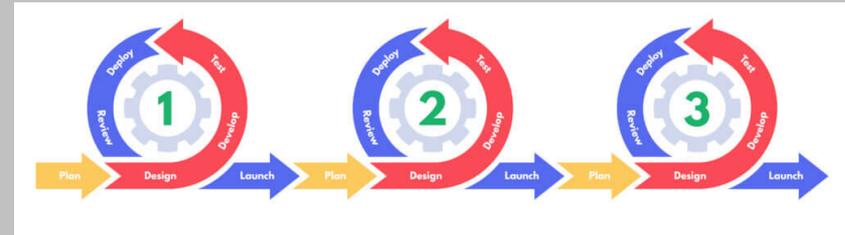
## Fragen, die dieses Kapitel beantwortet:

- Was bedeutet Agilität?
- Welche neuen Rollen gibt es beim agilen Vorgehen?
- Wie unterscheiden sich die Testphasen im agilen Umfeld?
- Welche Hindernisse und Lösungsansätze gibt es auf dem Weg zum agilen Testen?

## Wasserfall (klassischer Ansatz)

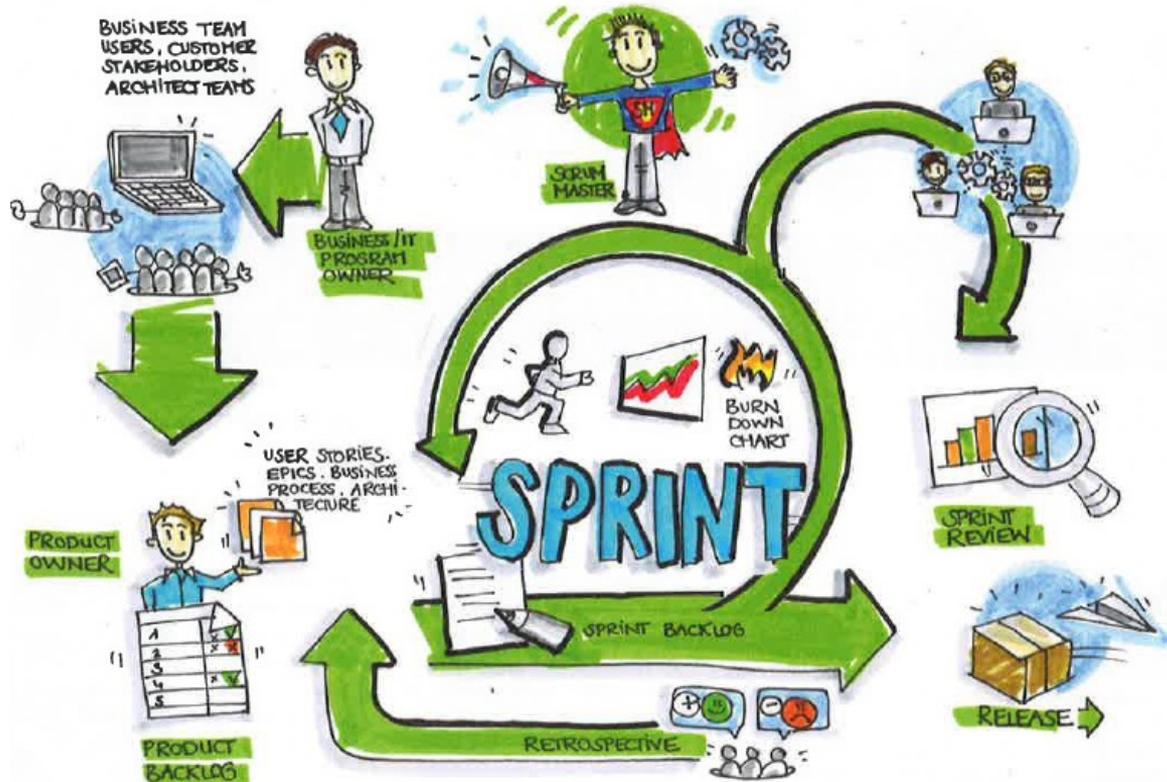


## Agile Projektmethodik



...unter sich ändernden Rahmenbedingungen  
**schnell und angemessen** zu agieren,  
ohne dabei **die Ziele** aus den  
Augen zu verlieren

# Qualitätssicherungsprozess in SCRUM



Wasserfall vs. Agil → ggf. auch Hybrid!



Quelle: <https://ifm-business.de/aktuelles/wp-content/uploads/2019/01/hybrides-projektmanagement-traditionell-agil-bruecke.png>

**SCRUM**

**XP**

**KANBAN**

**SCRUMBAN**

**Scrum@Scale (S@S)**

**SCRUM of SCRUMS**

**SPOTIFY Model**

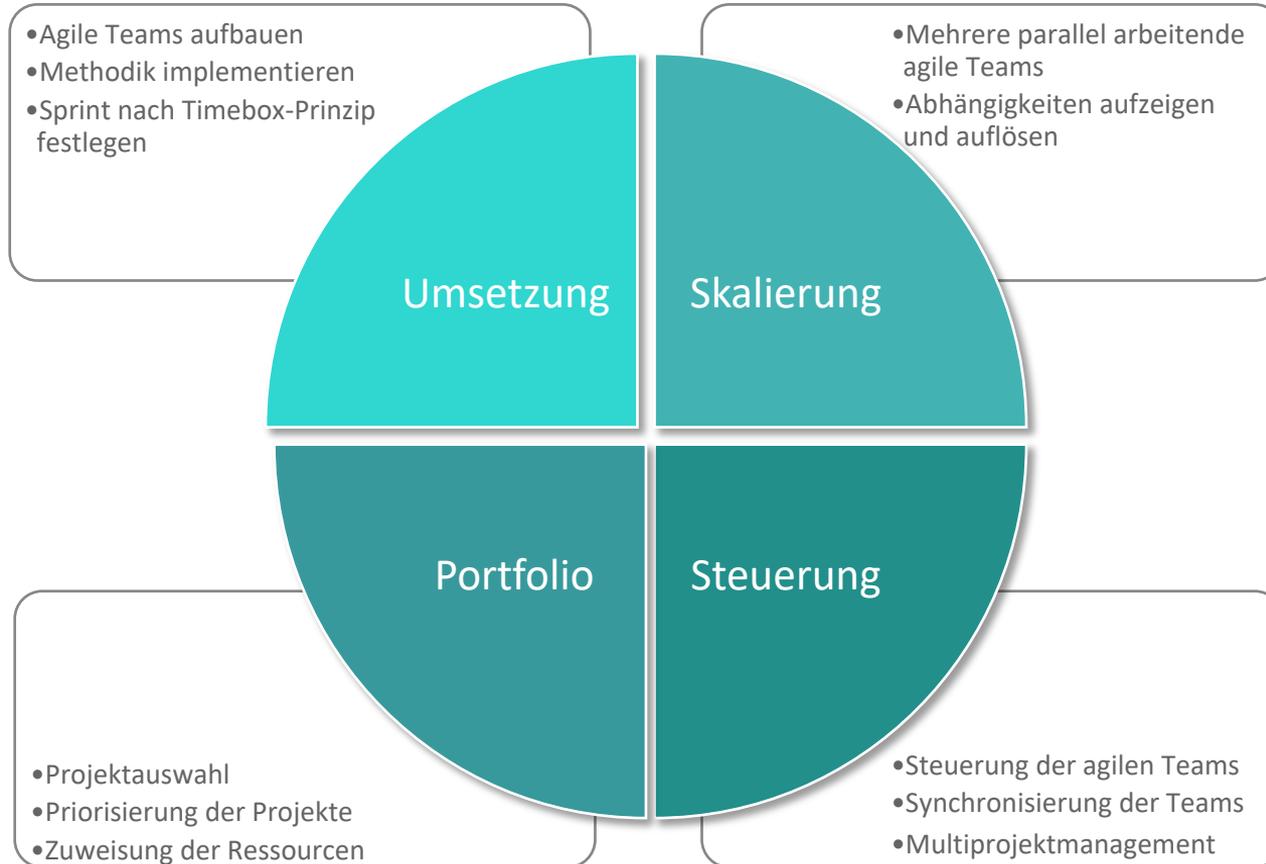
**DA (Disciplined Agile)**

**NEXUS**

**LeSS (Large Scale Scrum)**

**SAFe**

# Dimensionen der Agilität (angelehnt an SAFe)



## Nachmittag

### Testen im agilen Umfeld

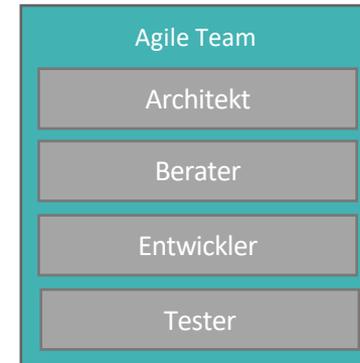
Agilität

Neue und alte Rollen

Agiles Testen

Hindernisse & Lösungen

- Bestehend aus 5 – 10 Mitgliedern
- Funktionsübergreifende Zusammensetzung
- Selbstorganisiert
- Eigenständig
- Arbeiten kurzen Iterationen (Sprints & Zyklen)
- Kundenorientierung



## Neue „agile“ Rollen

### **Agile Team Member**

Mitglieder sind z.B. Entwickler, Berater, Tester, Architekt  
Es gibt keine Hierarchie im Team  
Alle arbeiten daran die gemeinsamen Ziele zu erreichen

### **Product Owner**

Ansprechpartner für inhaltliche Vorgaben  
Verwaltet die Anforderungen in einem Backlog  
Überwacht den Fortschritt

### **Agile Coach**

Unterstützen die gesamte Organisation mit ihrem Methodenwissen und ihrer Erfahrung, um die agile Transformation zu begleiten

# Wird der Test Manager noch benötigt?



Testmanager

## Duale Rolle als Tester & Testmanager zugleich

### Rolle als Tester

#### Operative Testaufgaben

- Testfälle spezifizieren
- Testdaten generieren
- Testumgebung aufbauen
- Test ausführen
- Test verfolgen
- Test messen
- Testergebnisse auswerten
- Test protokollieren
- Fehler melden



### Rolle als Testmanager

#### Testmanagementaufgaben

- Test planen
- Test schätzen
- Test organisieren
- Test überwachen
- Test steuern
- Release freigeben

## Bestehende Rollen

### Testmanager

Nimmt eher eine Governance & Coach Rolle ein  
Gibt Vorgaben & erstellt das Testkonzept  
Analog des Agile Coach Ansprechpartner für Testmethodik

### Testkoordinator

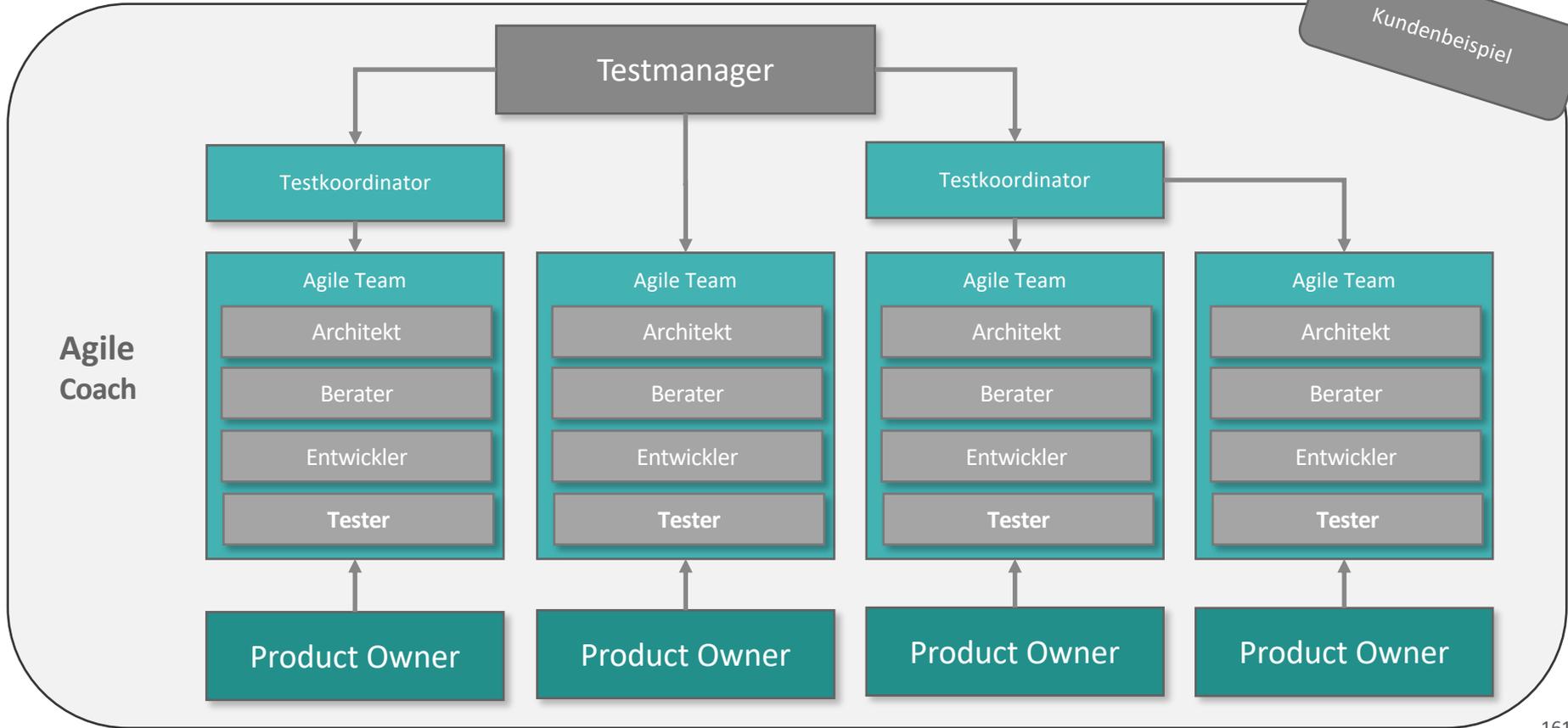
In größeren Projekten als Testmanagement Coach nötig

### Tester

Teil des agilen Teams  
Quality Coach im agilen Team  
Tester helfen dem Team die optimal mögliche und gewünschte Qualität zu liefern  
Definieren der Akzeptanzkriterien  
Schätzung der Testaufwände  
Prüfung & Anpassung bzw. Erstellung der Testfälle  
Durchführung der SFT's & FT's

# Testorganisation im agilen Umfeld aus der Praxis

Kundenbeispiel



## Nachmittag

### Testen im agilen Umfeld

Agilität

Neue und alte Rollen

Agiles Testen

Hindernisse & Lösungen

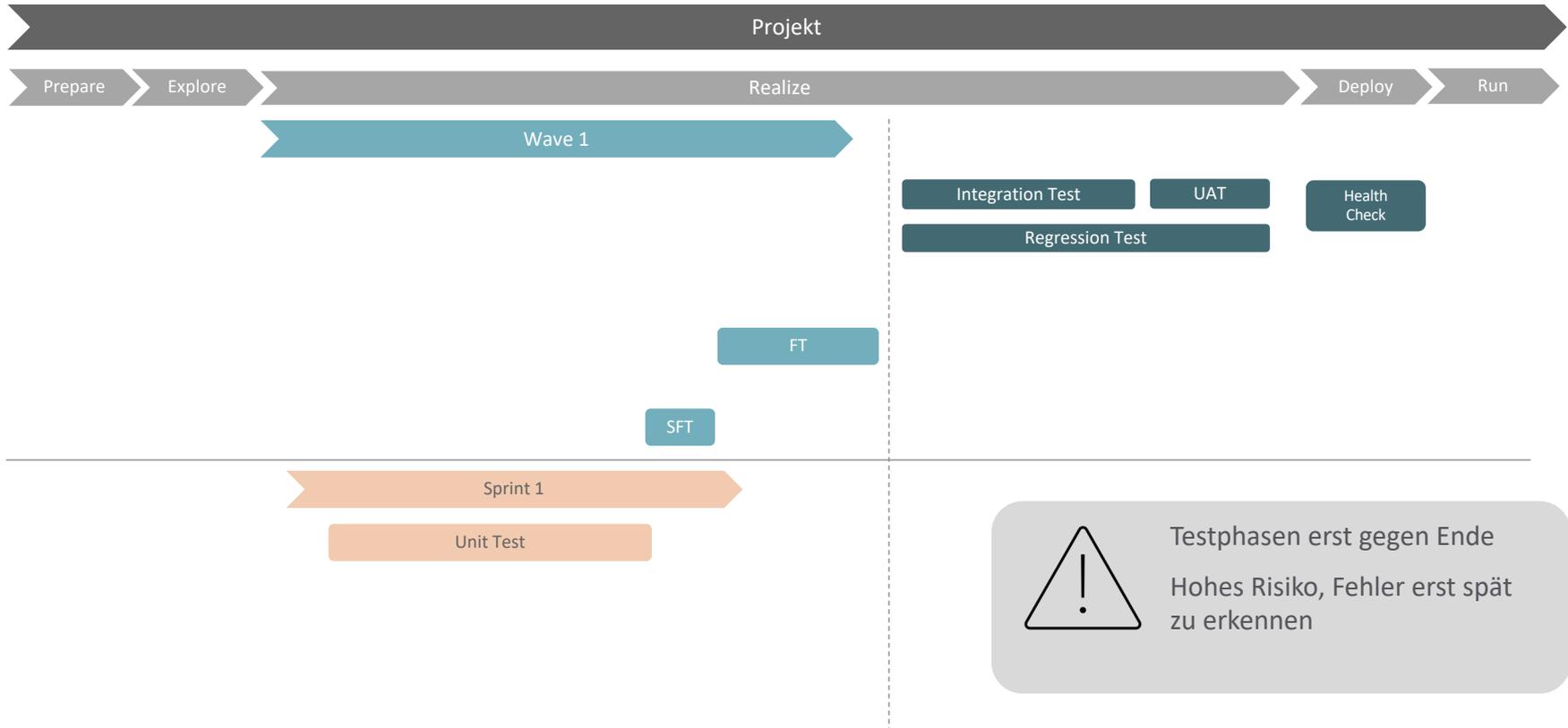
## ... für das Testen

- Anforderungen können sich häufig ändern oder entfallen
- Neue Anforderungen können hinzukommen
- Mehr Release-Zyklen ⇔ kleinere und häufigere Testphasen
- Qualität muss früher & häufiger sichergestellt werden
- Dezentrale Strukturen => mehr Ansprechpartner

## Testphasen im

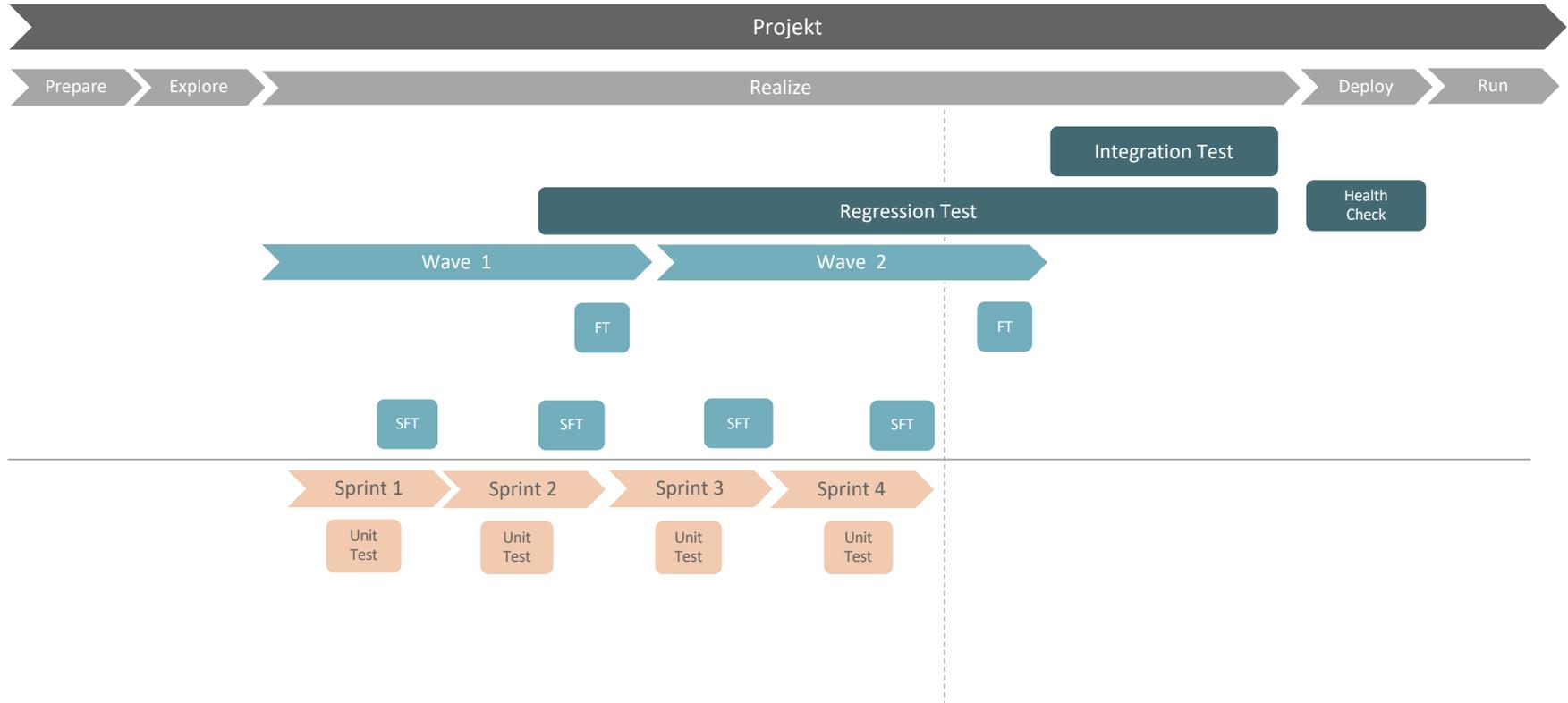
- Wasserfallmodell
- Hybriden Modell
- Agilen Modell

# Ablauf und Test eines Wasserfallprojektes

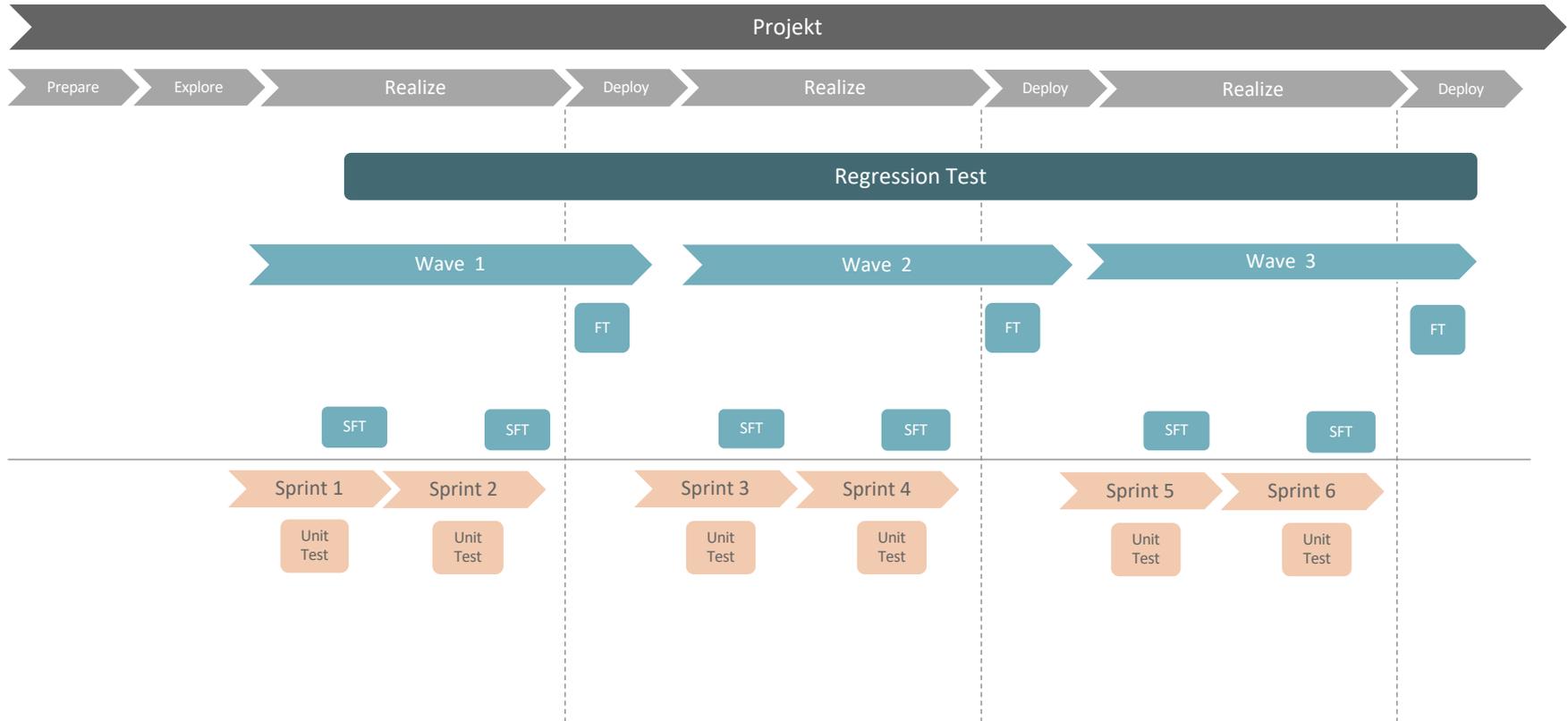


 Testphasen erst gegen Ende  
Hohes Risiko, Fehler erst spät zu erkennen

# Ablauf und Test eines hybriden Projektes



# Ablauf und Test eines agilen Projektes



# Teststufen Übersicht

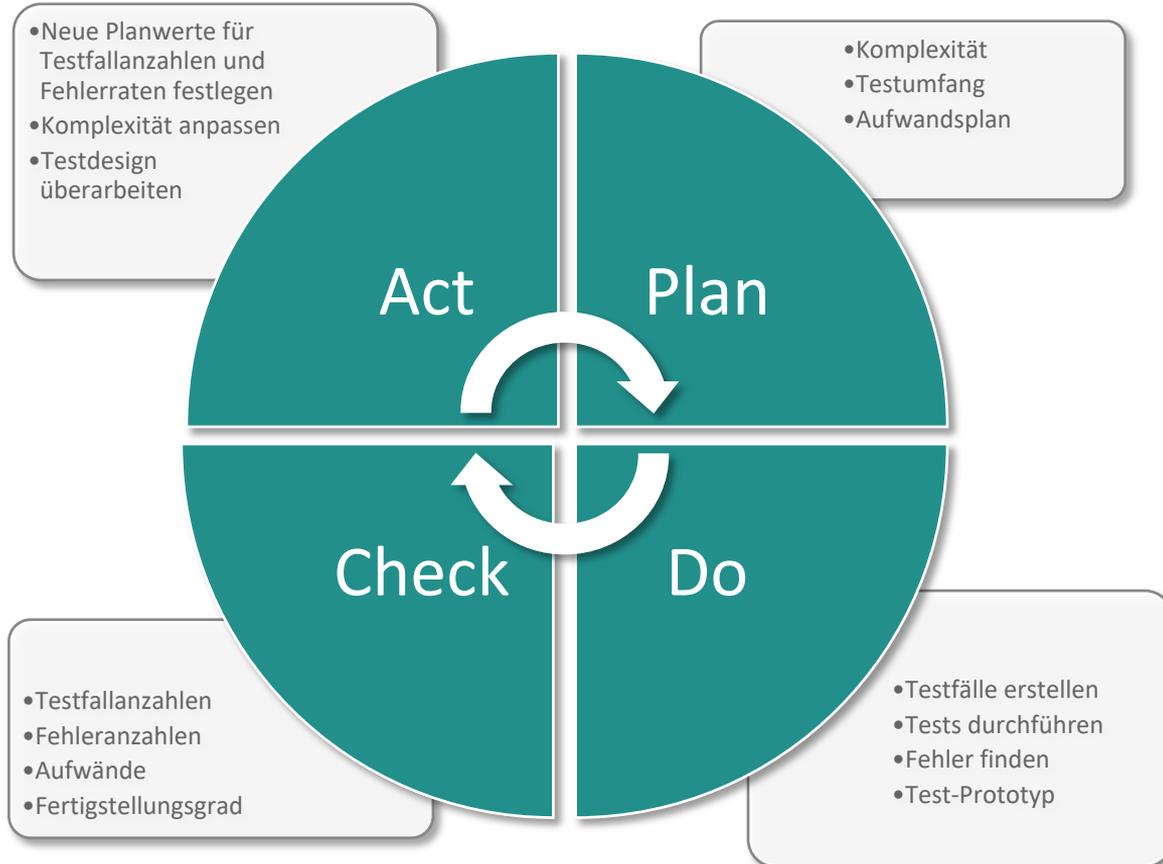
Kundenbeispiel

Teststufen	Wann	Wer testet	Testgegenstand	Funktion
Unit Test (UT)	Während Entwicklung	Entwickler	Durchgeführte Entwicklung / Konfiguration	Technische Korrektheit sicherstellen (nicht Formal) Bestätigung über Statuswechsel auf „zu Testen“
Single Functional Test (SFT)	Nach Umsetzung	Tester Projektorganisation	Anforderungen / Akzeptanzkriterien einer User Story	Abnahme der Funktionalität
Feature Test (FT)	Am Ende eines Entwicklungszyklus	Tester	Feature / Komponente / Modul	Softwarequalität des Features sicherstellen. Vorbereitung für weitere integrative Tests bzw. prozessuale Tests.
Feature Integration Test (FIT)	Nach dem Feature Test	Tester	Schnittstellen & Feature- übergreifende Prozesse	Integration zw. Features & Funktion der Einzelprozesse sicherstellen
System-Integrationstest (SIT)	Nach Ende der Umsetzung	Fachabteilung / Linienorganisation	End2End Prozesse inkl. Schnittstellen zu umgebenden Systemen	Integration zw. Systemen & Funktion der End2End Prozesse sicherstellen
Regressionstests (RT)	Start nach erstem Entwicklungszyklus sinnvoll	wenn möglich automatisiert	Fertiggestellte Prozesse / Applikationen	Sicherstellung, dass Anwendung/Prozess ordnungsgemäß funktioniert
User Acceptance Test (UAT)	zwischen SIT und Go-Live.	komplette Linienorganisation	Kernprozesse mit Vorgaben und exploratives Testen	Sicherstellen bzw. erhöhen der User Akzeptanz für das neue System



Wichtiger Indikator, ob Testeingangsparameter für weitere Teststufen erreicht wurden.

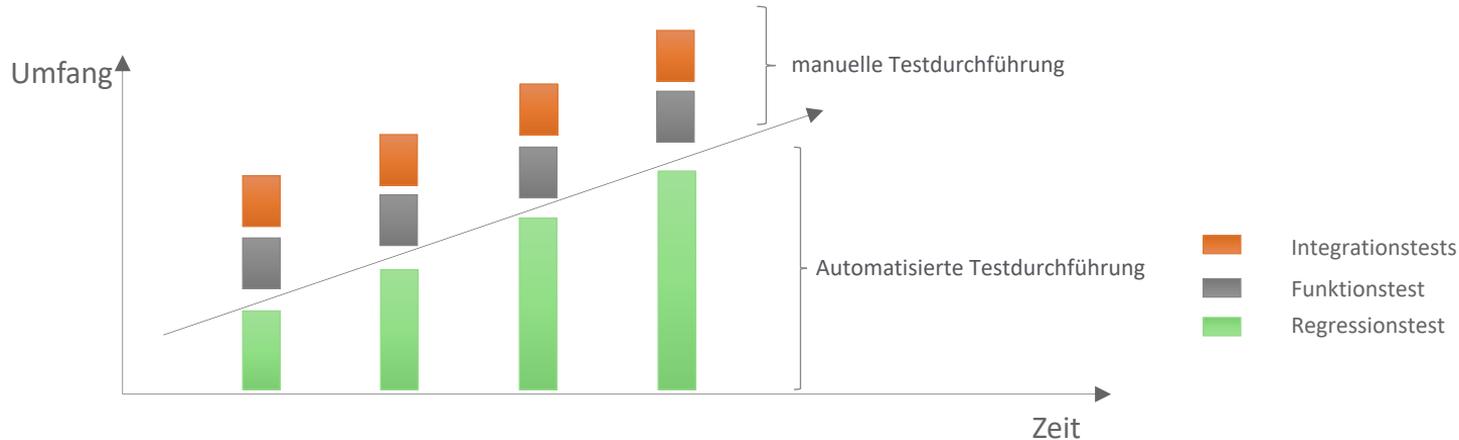
# Deming-Zyklus am Beispiel des Tests verdeutlicht



# Testautomatisierung im agilen Umfeld

- Mit steigender Anzahl der Iterationen/Sprints wächst das Produkt
- Der Umfang der Regressionstests steigt zusätzlich zum Test der neuen Funktionen
- Verfügbare Testzeit und Größe der Testorganisation bleibt gleich

**Bedeutet:** Ohne Testautomatisierung kann ein agiles Software-Projekt kaum funktionieren!



## Nachmittag

### Testen im agilen Umfeld

Agilität

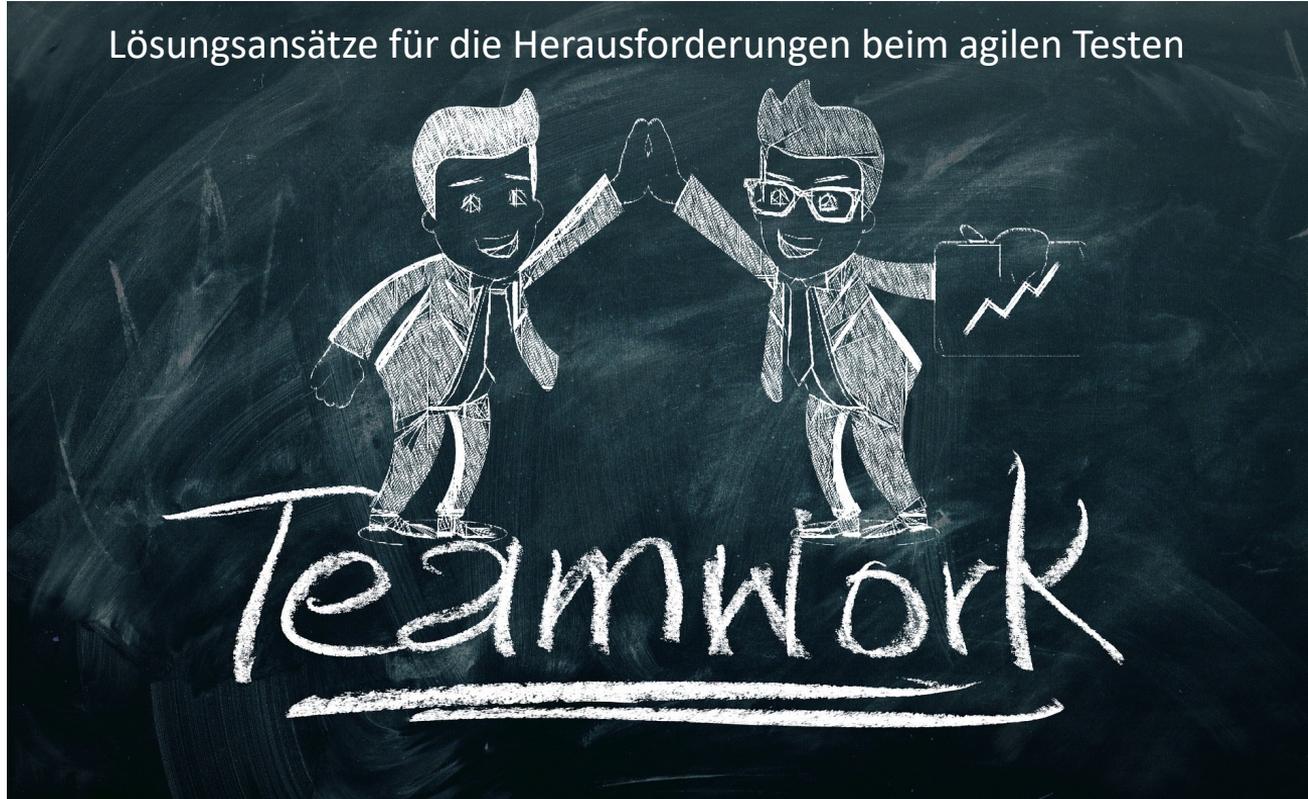
Neue und alte Rollen

Agiles Testen

Hindernisse & Lösungen

- Ausreichender Austausch mit Kunden bzgl. seiner Bedürfnisse
- Hohe Qualität bei der Definition der Akzeptanzkriterien bzw. Anforderungen
- Sinnvolles Schneiden von Features & User Stories für die Umsetzung & die Tests
- Früher und qualitativ hochwertiger Test, inkl. Schnittstellen, Berechtigungen, Nicht-funktionaler Anforderungen etc.)
- Entwicklung von eigenständig test- & deployfähigen Objekten (Features)
- Analyse & Auflösung von funktionalen, prozessualen & zeitlichen Abhängigkeiten

Lösungsansätze für die Herausforderungen beim agilen Testen

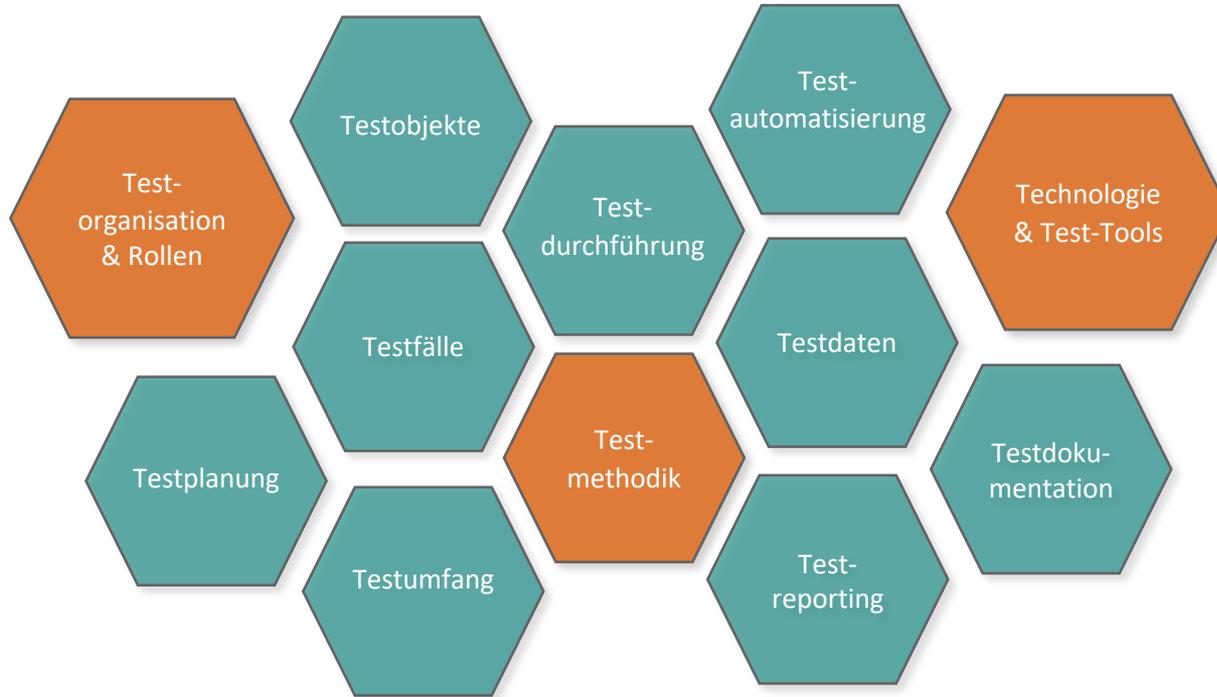


# Offene Fragen zum Testen im agilen Umfeld?



# SAP Testmanagement Landkarte

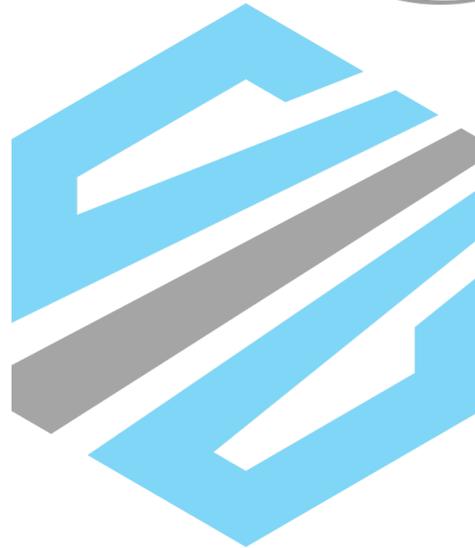
## 12 Handlungsfelder



Kaffeepause



# Erfahrungsbericht XXL-Projekt 2022-2023



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Testreporting

## Nachmittag

XXL-Großprojekt 2022-2023

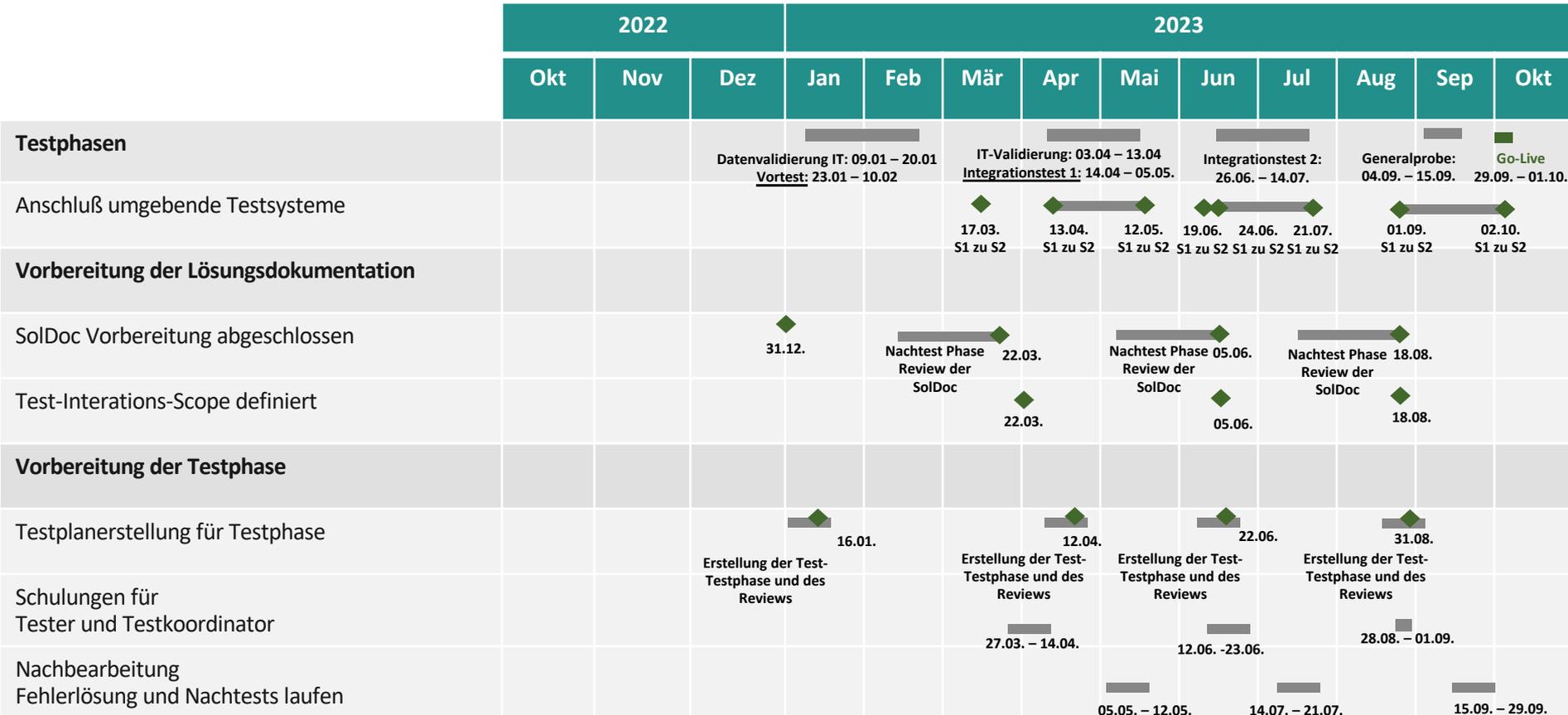
Projektgesamtübersicht

Ergebnisse

Aufwandsbetrachtung

Lessons Learned

# Projektgesamtübersicht



# Projektgesamtübersicht: Generalprobe

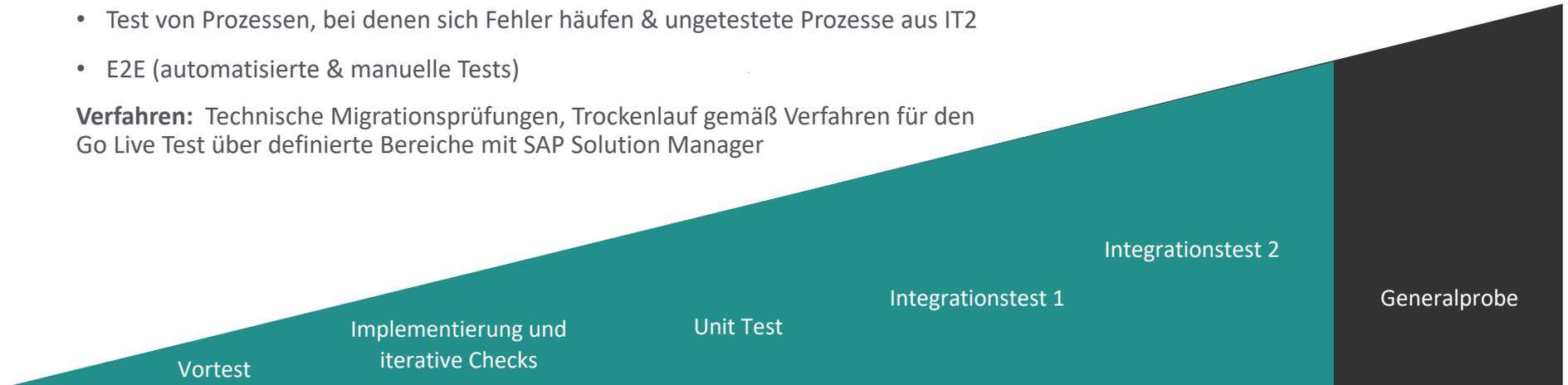


**Zeit:** Test: von Montag, 4. September bis Freitag, 15. September 2023

## Scope:

- Test der Hauptfunktionen inkl. Schnittstellen
- Validierung der Migration
- Trockenlauf von Stichprobentests  
(Vergleich von Berichten vorher & nachher sowie Stichprobenprüfungen)
- Test von Prozessen, bei denen sich Fehler häufen & ungetestete Prozesse aus IT2
- E2E (automatisierte & manuelle Tests)

**Verfahren:** Technische Migrationsprüfungen, Trockenlauf gemäß Verfahren für den Go Live Test über definierte Bereiche mit SAP Solution Manager



Um eine angemessene Prüfung mit einem definierten Umfang zu ermöglichen, wurde eine Risikobewertung mit Fokus auf zentrale, wertvolle Prozesse durchgeführt, basierend auf den folgenden Kriterien:

- Es wurden führende Unternehmen nach Umsatzvolumen ausgewählt.
- Vor allem wurden E2E-Prozesse mit dem Schwerpunkt der Prozessintegration berücksichtigt. Über diese E2E-Prozesse werden auch die Hauptfunktionen in Verkauf, Produktion, Lieferung und Zahlung abgesichert.
- Spezielle Prozesse wie Zoll und technische Beschwerden sollen in ausgewählten Niederlassungen validiert werden, die repräsentativ für solche Funktionalitäten sind.
- Zusätzlich werden Kernfunktionalitäten in Vertrieb und Betrieb mit direktem Einfluss auf Bilanzen überprüft, um die finanzielle Funktionalität im S/4HANA-System zu sichern.

## Grundsätze

Das vorgestellte Projekt ist eines der größten Projekte mit erheblichen Auswirkungen auf das zentrale SAP (S/4HANA) System für einen sehr großen deutschen Mittelständler:

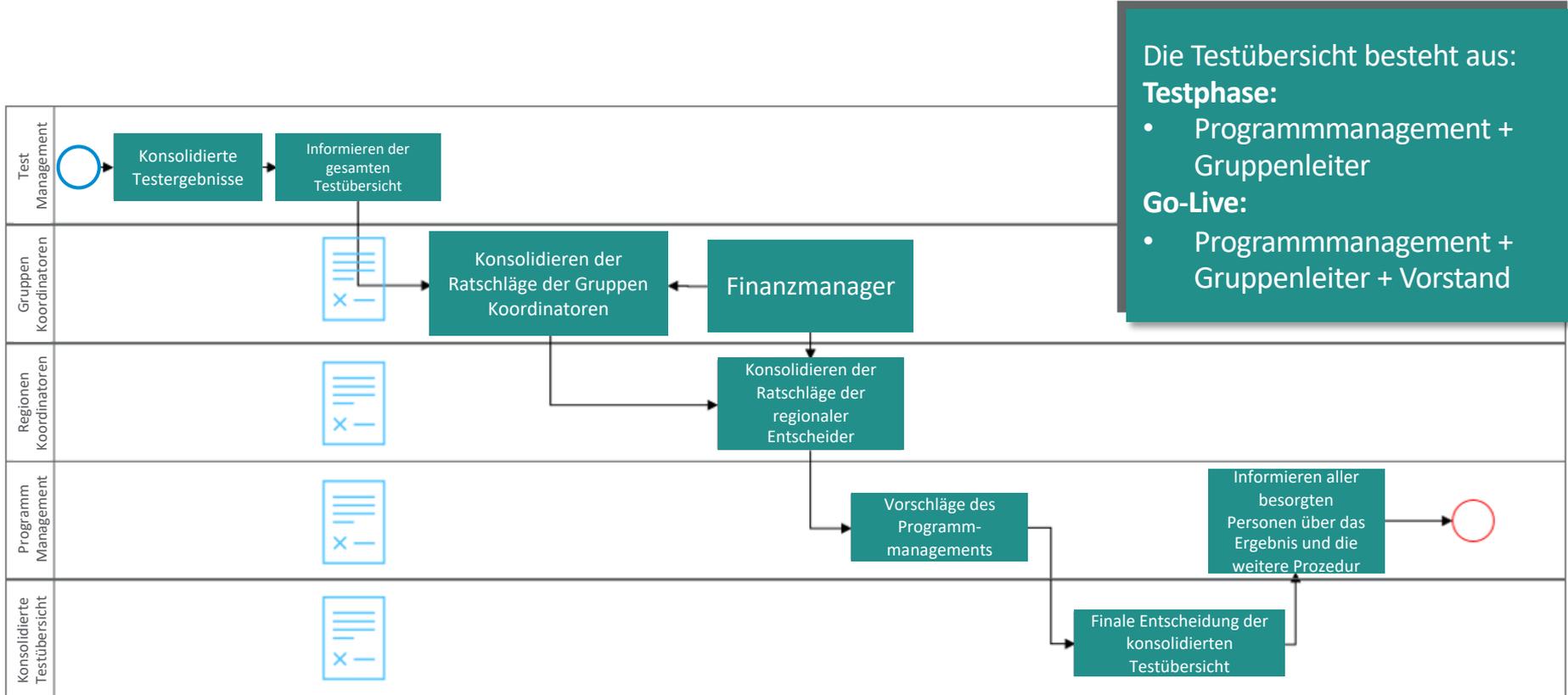
- Zentrale Prozessexperten aus verschiedenen Geschäftsbereichen müssen beteiligt sein.
- Alle Tochtergesellschaften sollten durch jemanden mit Geschäftsverantwortung vertreten sein.
- Dennoch muss die Teilnehmerzahl für effektive und sinnvolle Entscheidungen begrenzt werden.
- Deshalb können Koordinatoren für Gruppen/Regionen verschiedene Bereiche des Unternehmens vertreten.

## Organisation

Die Projekt-Testmanager und regionalen Testmanager koordinieren und konsolidieren:

- Teststatus und -fortschritt
- Regelmäßige Information der Entscheider
- Bereitstellung des Teststatusberichts 2 Tage vor dem Testübersichts-Meeting
- Einladung zum Testübersichts-Meeting
- Unterstützung der Entscheidungsfindung

# Projektgesamtübersicht: Testübersicht - Entscheidungspfad



Die Testübersicht besteht aus:

**Testphase:**

- Programmmanagement + Gruppenleiter

**Go-Live:**

- Programmmanagement + Gruppenleiter + Vorstand

## Nachmittag

XXL-Großprojekt 2022-2023

Projektgesamtübersicht

Ergebnisse

Aufwandsbetrachtung

Lessons Learned

# Ergebnisse: Test Ergebnisse pro Region



Beispiele sind aus dem Report des Integrationstest 2

Test Ergebnisse pro Region

Region	Gesamtzahl Testfalle Anzahl	OK Anzahl	OK %	Ausgeführt %
A	8.605	8.204	95.3%	97.3%
B	10.628	10.374	97.6%	99.2%
C	22.934	21.100	92.1%	96%
<b>GESAMT</b>	<b>42.167</b>	<b>39.678</b>	<b>94.1%</b>	<b>97.1%</b>

# Ergebnisse: Testergebnisse pro Priorität



## Testergebnisse pro Priorität

Region	Gesamtzahl Testfälle Anzahl	OK Anzahl	OK %	Ausgeführt %
Priorität 1	13.087	12.412	94.8%	97.7%
Priorität 2	14.959	14.207	95%	97.2%
Priorität 3 & 4	14.121	13.059	92.5%	96.2%
<b>GESAMT</b>	<b>42.167</b>	<b>39.678</b>	<b>94.1%</b>	<b>97.1%</b>

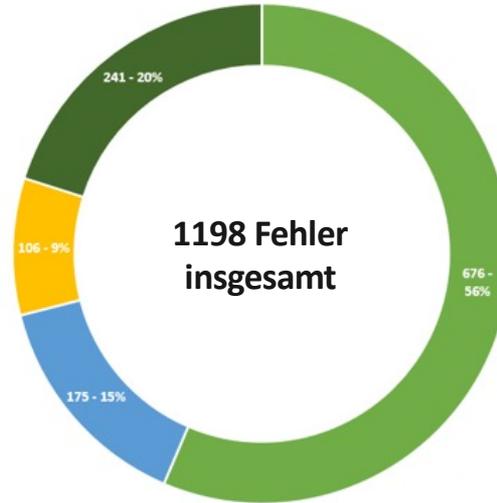
# Ergebnisse: Testergebnisse pro Gruppe

## Testergebnisse pro Modul / Gruppe

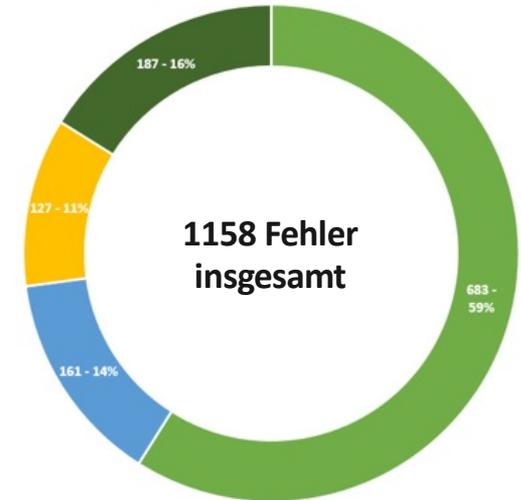
Region	Gesamtzahl Testfälle Anzahl	OK Anzahl	OK %	Ausgeführt %
Operations	14.370	13.525	94.1%	96.5%
Sales	10.365	9.947	96%	98.7%
Finance	5.515	4.943	89.8%	95.2%
Controlling	4.238	3.953	93.3%	97.1%
Service	2.976	2.841	95.5%	98%
Product Lifecycle Management	92	88	96%	98.9%
End-2-End	4.607	4.374	94.9%	96.5%
<b>GESAMT</b>	<b>42.167</b>	<b>39.678</b>	<b>94.1%</b>	<b>97.1%</b>

Status	Priorität 1 & 2
Neu	233
In Bearbeitung	336
Lösungsvorschlag und geschlossen	1787
<b>GESAMT</b>	<b>2.356</b>

### Fehler der Priorität 1: Sehr hoch



### Fehler der Priorität 2: Hoch



- Geschlossen
- In Bearbeitung
- Neu
- Lösungsvorschlag

## Nachmittag

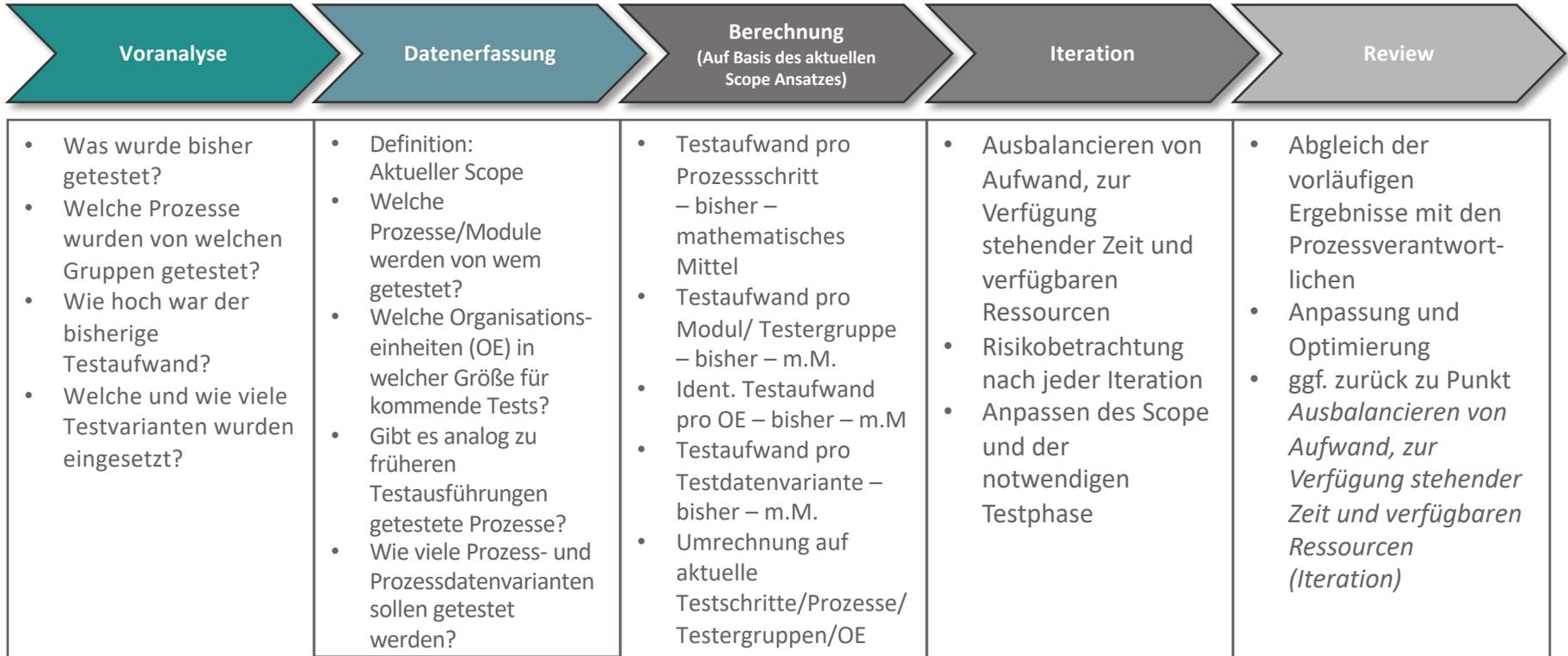
XXL-Großprojekt 2022-2023

Projektgesamtübersicht

Ergebnisse

Aufwandsbetrachtung

Lessons Learned



## Voranalyse:

1. Was wurde bisher getestet?  
(mindestens 3 Testphasen möglichst mit Aufwandsdaten heranziehen)
2. Was wurde in diesen Testphasen getestet, insbesondere welche und wie viele Prozesse von welchen Gruppen?
3. Analysieren Sie, welchen Testaufwand es bisher bei welchen Tests gab
4. Klären Sie, welche und wie viele Testdatenvarianten wo eingesetzt wurden  
(soweit sich dies nachvollziehen lässt)

## Datenerfassung:

5. Wie sieht der aktuelle Scope aus?
6. Wie viele Prozesse müssen von wem (Module/Gruppen) getestet werden?
7. Welche Organisationseinheiten mit welcher (ungefähren) Größe sollen in den kommenden Test einbezogen werden?
8. Gibt es Prozesse, die analog zu früheren Testausführungen getestet wurden (Normung)?
9. Wie viele Prozessvarianten sollen getestet werden?
10. Wie viele Testdatenvarianten sollen getestet werden?

## **Berechnung (auf Basis des aktuellen Scope-Ansatzes):**

11. Testaufwand pro Prozessschritt – bisher – mathematisches Mittel
12. Testaufwand pro Modul/Testergruppe – bisher – mathematisches Mittel
13. Identifikation Testaufwand pro Organisationseinheit – bisher – mathematisches Mittel
14. Testaufwand pro Testdatenvariante – bisher – mathematisches Mittel
15. Umrechnung auf aktuelle Testschritte – Prozesse – Testergruppen - Organisationseinheiten

## Iterationen:

16. Ausbalancieren von Aufwand, zur Verfügung stehender Zeit und verfügbaren Ressourcen
17. Risikobetrachtung nach jeder Iteration
18. Anpassen des Scope und der notwendigen Testphase

## Review:

19. Abgleich der vorläufigen Ergebnisse mit den Prozessverantwortlichen
20. Anpassung und Optimierung
21. Ggf. zurück zu Schritt 16

- Klären:
  - Was wurde bisher getestet?
    - Analyse der bisherigen Testphasen (2-3 falls möglich)
  - Welche Prozesse existieren und wurden schon getestet?

Anzahl von Process text	Spaltenbeschriftungen																							
Zeilenbeschriftungen	0001	0002	0004	0006	0007	0008	0010	0012	0013	0014	0015	0035	0040	0060	0070	0072	0075	0076	0080	0090	0091	0100	0110	0120
A	267	182	115	10	10	10	106	10	10	10	10	10	97	71	72	97	54	12	62	124	62	79	90	3
C	292	66	57				66						62	50	52	59	59	42	48	52	43	48	48	4
D			901										17	18							108			
M			286											185	47				47	184		47	185	4
O	1938	98	80				133				22			61	55	136	93	1	117	116	16	121	16	2
P		92																						
R		2																						
S	282	664												518	173		157		154	384		149	308	9
T																								
<b>Gesamtergebnis</b>	<b>2873</b>	<b>1296</b>	<b>1153</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>305</b>	<b>10</b>	<b>10</b>	<b>10</b>	<b>32</b>	<b>10</b>	<b>176</b>	<b>903</b>	<b>399</b>	<b>292</b>	<b>363</b>	<b>55</b>	<b>428</b>	<b>860</b>	<b>229</b>	<b>444</b>	<b>647</b>	<b>25</b>

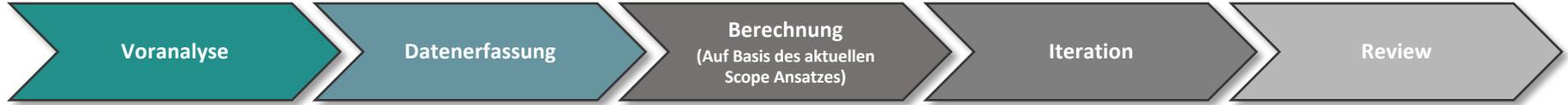
- Bisherigen Aufwand prüfen
  - Wie viel Aufwand gab es bis zu diesem Zeitpunkt?

Prozess	Aufwand (PT)
936 OQ.1.1 - Inspection methode	27
937 OQ.1.2 - Sampling Procedure	8
938 OQ.1.3 - Sampling scheme	12
939 OQ.1.4 - Dynamic Modification Rule	20
940 OQ.1.5 - Master inspection characteristic	29
941 OQ.1.6 - Material master data view Quality	8
942 OQ.1.7 - Inspection plan (Test plan)	31
943 OQ.10.1 - Concession	100
944 OQ.11.1 - First sample	1
945 OQ.2.1 - Goods receipt	34
946 OQ.2.2 - Concession	4
947 OQ.3.1 - Check adopt data	8
948 OQ.3.2 - inspection lot without documents	50
949 OQ.3.3 - inspection lot with documents	45
950 OQ.3.4 - inspection waives	5
951 OQ.4.1 - Good receipt	10
952 OQ.4.2 - Check basic data	37
953 OQ.4.3 - Check results	22
954 OQ.4.4 - inspection process BR	11
955 OQ.5.1 - Notification vendor	26
956 OQ.6.10 - rework order NB	8
957 OQ.6.11 - rework operation	45

# Aufwandsbetrachtung: Gesamtaufwand

- Ableiten der Anzahl der Prozesse zum Test
- Wie viele Personen/Organisationseinheiten stehen zur Verfügung
- Daraus Gesamtaufwand ableiten
  - Wie ist die Verteilung zwischen den Modulen?
  - Wie ist die Verteilung auf Organisationseinheiten?

Distribution	PUR	LO	FI/CO	SCM	PP	PM	QM	Sales	CRM	PLM	BW	actual	size			
Company Coc Name	150	150	275	225	175	425	75	75	25	2022,0				Person Days		
0001	31,5	31,5	7,0	47,3	30,0	50,0	8,8	15,8	5,3	227,1	very big			Person Days		originally planned
0002	2,0	2,0	7,0	3,0	5,0	9,0	1,6	1,0	0,3	30,8	big			Person Days		actual plan
0004	3,0	3,0	7,0	4,5	3,0	4,0	0,7	1,5	0,5	27,2	medium			Person Days		
0006	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0007	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0008	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0010	0,3	0,3	7,0	0,5	3,0	0,0	0,0	0,2	0,1	11,3	medium			Person Days		
0012	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0013	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0014	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0015	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0032	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0040	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0060	4,3	4,3	7,0	6,4	5,0	9,0	1,6	2,1	0,7	40,4	big			Person Days		
0061	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		
0062	0,3	0,3	7,0	0,5		0,0	0,0	0,2	0,1	8,3	small			Person Days		



Die dargestellte Vorgehensweise unter Betrachtung von manuellen Testerfahrungen und der Berücksichtigung von Prozess- und Testdatenvarianten führt in der Regel zu einer Aufwandsbetrachtung für die Fachbereiche im Rahmen  $\pm 10\%$  (bei besseren Daten sind noch wesentlich bessere Ergebnisse möglich).

→ Monate vor einer Testphase kann Ihr Fachbereich planen!

## Nachmittag

XXL-Großprojekt 2022-2023

Projektgesamtübersicht

Ergebnisse

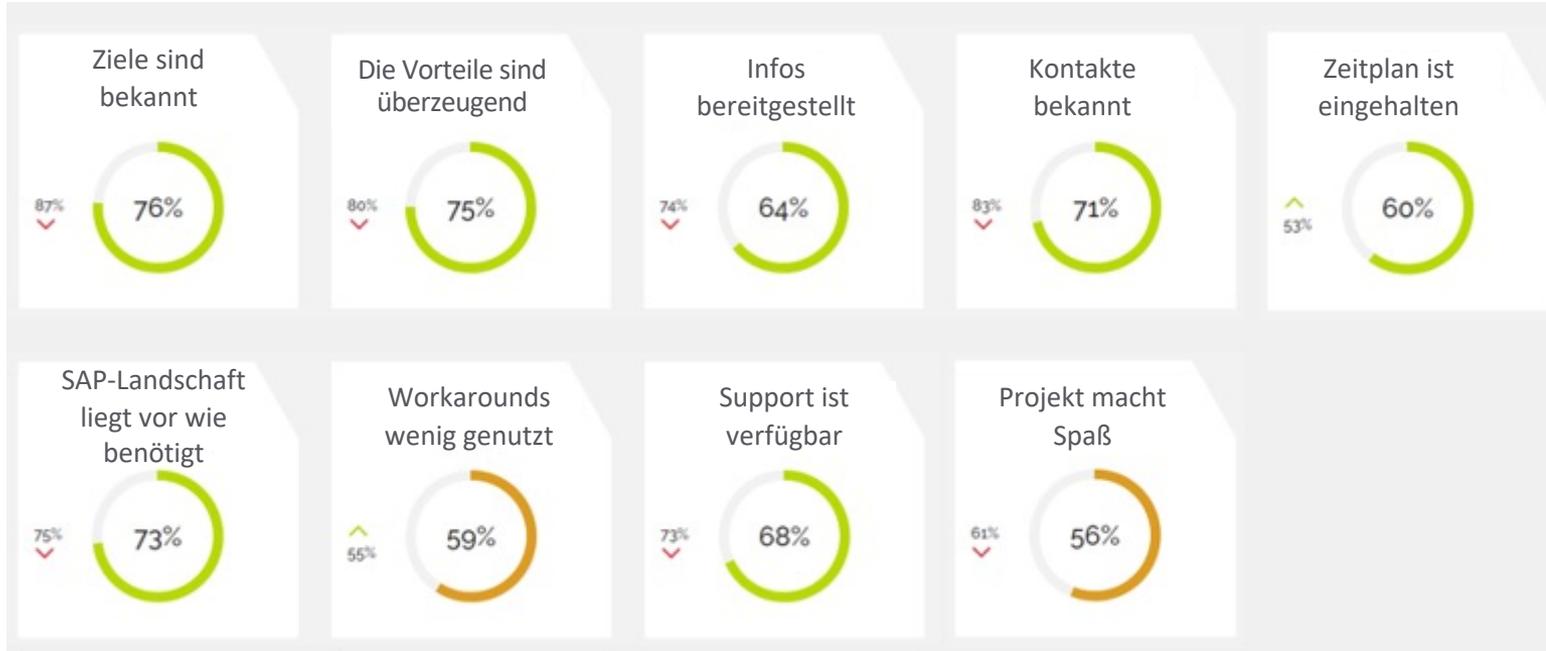
Aufwandsbetrachtung

Lessons Learned

## Lessons Learned:

- Trotz zahlreicher Schulungsmaßnahmen wurde der Aufwand der Testfallerstellung und präzisen Dokumentation seitens der Fachbereiche unterschätzt oder in Teilen ignoriert (Fachbereiche müssen sehr früh in die Erstellung von Testfällen involviert werden).
- Die Information, wer wann was testen soll, muss Monate im voraus passieren. Fachbereiche planen (weltweit) frühzeitig Ihren Jahresurlaub und können diesen nur schlecht wieder anpassen (5 Monate vor der Testphase ist zu spät).
- Die Ansprüche der Wirtschaftsprüfer hängen mehr als von gesetzlichen Regelungen, Prüfvorgaben oder internen Leitfäden vom Ausbildungsstand des jeweiligen Prüfers ab. Deshalb sind frühzeitig Abstimmungen zwischen Wirtschaftsprüfer, Projektleiter und Testmanager erforderlich.
- Testfälle für Tester müssen sehr transparent, einfach beschrieben und möglichst simple durchzuführen sein (weitere Erhöhung der Präzision erforderlich)
- Zu nutzende Testdaten müssen direkt in der Testausführung für den Tester sichtbar und eindeutig sein (eine Darstellung in Folien- und/oder Excelform und eine zugehörige Schulung sind nicht ausreichend)

# Lessons Learned: Umfrage an die Tester und Keyuser

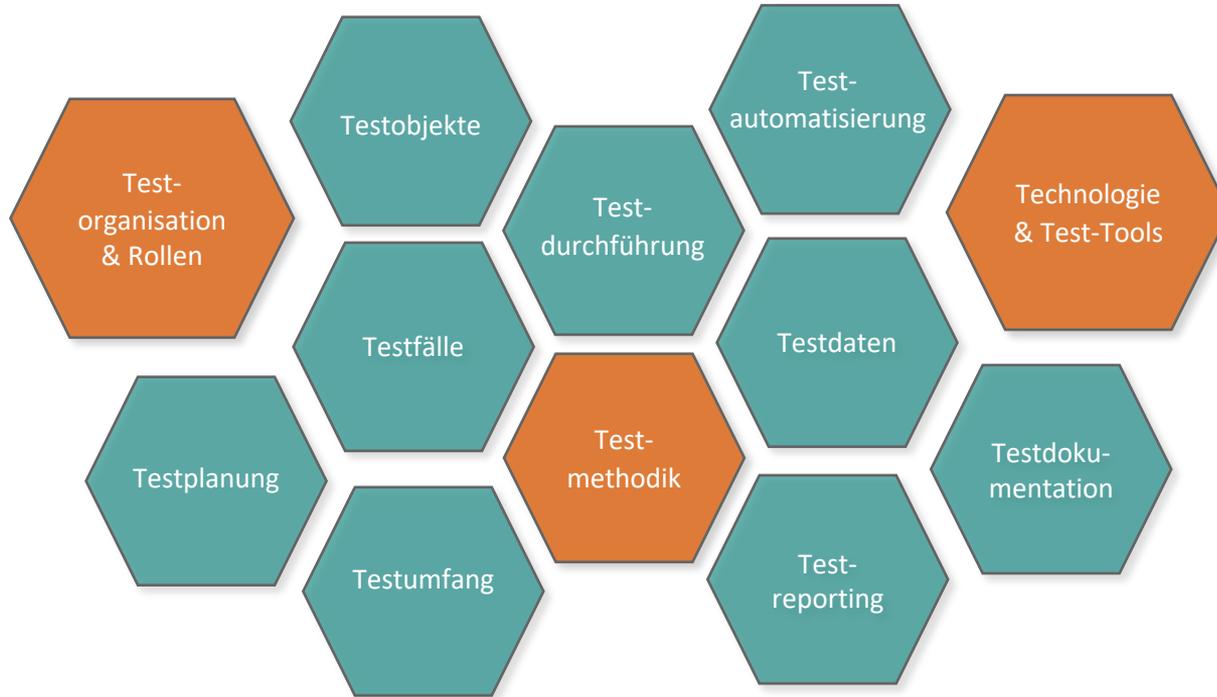


# Offene Fragen zum Thema XXL-Großprojekt?



# SAP Testmanagement Landkarte

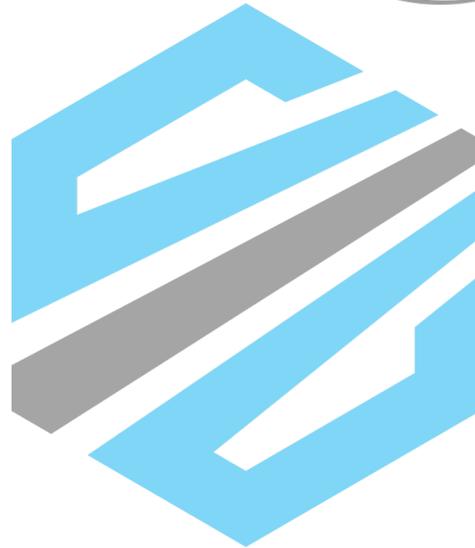
## 12 Handlungsfelder



Kaffeepause



# Test-Reporting



## Vormittag

Kennenlernen

Das 1x1 des Testens

Testautomatisierung

Mittagspause

## Nachmittag

Testen im agilen Umfeld

Erfahrungsberichte I.

Test-Reporting

## Nachmittag

### Test-Reporting

Einleitung

Reporting bei Test-Vorbereitung und -Durchführung

Testabschluss & Testdokumentation

Fazit & Tipps aus der Praxis

## Fragen, die dieses Kapitel beantwortet:

- Was sind die Grundlagen für ein effizientes Test-Reporting?
- Warum ist es wichtig das Reporting schon frühzeitig zu definieren?
- Was sind relevante Testmetriken?
- Was sind Testabnahme- und Testendekriterien?
- Was sind wichtige Tipps für die Praxis in Bezug auf das Test-Reporting?

## Zusammenstellung, Analyse und Präsentation von Informationen über den Testfortschritt, Testergebnisse und Testmetriken

### Ziel und Nutzen

- Das Hauptziel von Test-Reports ist es, die **Transparenz und Kommunikation im Projekt zu verbessern**, indem sie wichtige Informationen über den **Testfortschritt**, die **Qualität** und die **Risiken** liefern.
- Durch Test-Reporting können Projektmanager, Entwickler und andere Stakeholder fundierte **Entscheidungen treffen**, um den Projekterfolg sicherzustellen und Probleme frühzeitig zu erkennen und anzugehen.



## Wesentliche Aspekte zur Überwachung des Testfortschrittes?

- **Produktrisiken**  
Restrisiko, beherrschte Risiken, ...
- **Fehlerzustände**  
Anzahl der gefundenen und/oder behobenen Fehler
- **Testdurchführung**  
Konzeption und Durchführung von Testfällen, Vorgehen, ...
- **Überdeckung von Testbasis, Testcode oder anderen Testaspekten**
- **Vertrauen**  
Maß an Vertrauen in Bezug auf die Produktqualität

## Wem berichten wir?

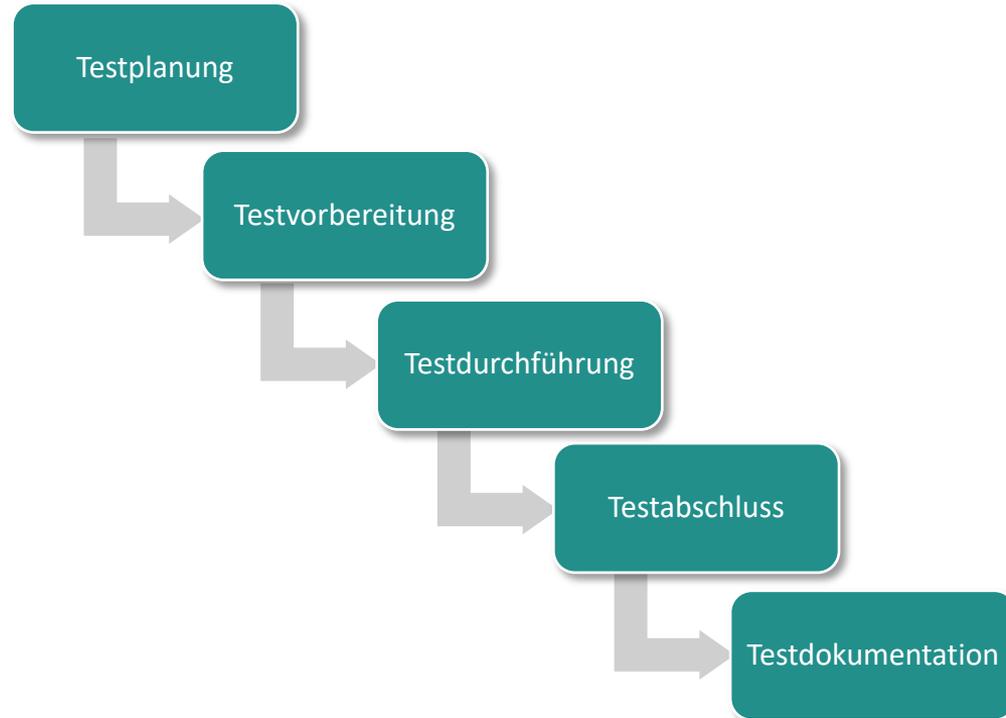
- Tester
- Entwickler / Defekt Bearbeiter
- Test-Manager & Testkoordinatoren
- Projekt Manager
- Programm-Leiter
- Unternehmensleitung



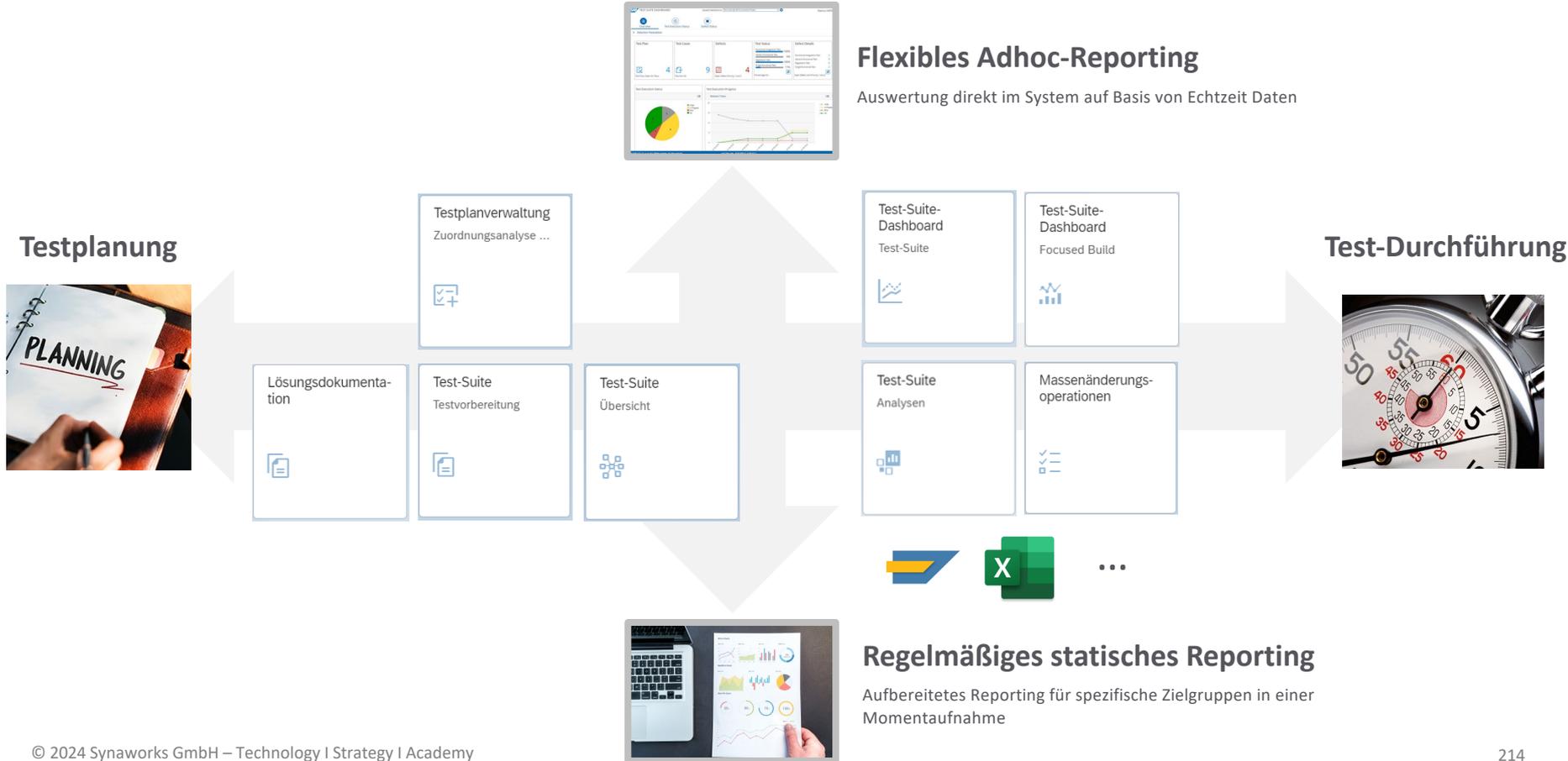
Für den jeweiligen Empfänger ist ein geeigneter Detaillierungsgrad zu verwenden



## Test-Reporting findet in mehreren Phasen statt



# Aus der Praxis: Adhoc – Statisches Reporting – Beispiel Solution Manager





Projektspezifische Analyse der Reporting-Anforderungen

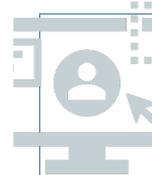


Entwicklung eines projektspezifischen Reporting-Konzepts

- Festlegung von Messgrößen und Metriken
- Hieraus können auch Anforderungen für Attribuierung von Testfällen, Fehlern etc. entstehen



Integration von Reporting-Anforderungen in das Testkonzept



Auswahl geeigneter Reporting-Tools und – Technologien

- Bzw. notwendige Integration schaffen



Kommunikation des Reporting-Prozesses



Regelmäßige Überprüfung und Anpassung des Reporting-Konzepts

## Nachmittag

### Test-Reporting

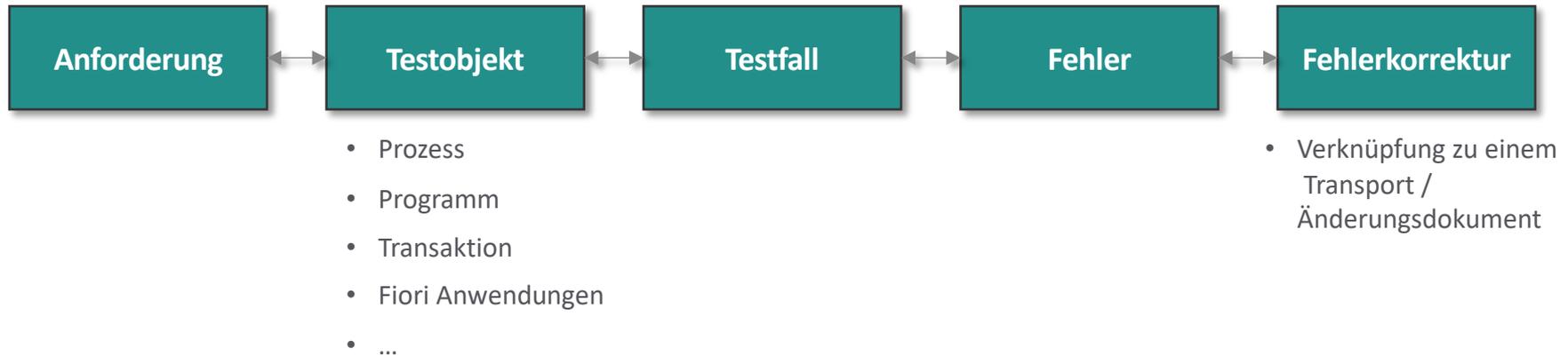
Einleitung

Reporting bei Test-Vorbereitung und -Durchführung

Testabschluss & Testdokumentation

Fazit & Tipps aus der Praxis

- Rückverfolgbarkeit (Traceability) wird zunehmend wichtiger bzw. durch regulatorische Anforderungen verpflichtend
- Sind für die verschiedenen Bereiche unterschiedliche Tools im Einsatz ist zu prüfen, wie diese integriert werden können (manuell / automatisiert)



- Integration des Testaktivitäten in den übergreifenden **Projektplan**
- **Dokumentation** / Schulungsunterlagen der **Testobjekte** (Prozesse, Schnittstellen, etc.)?
  - z.B. Vollständigkeit Prozessdokumentation je Stream / Bereich

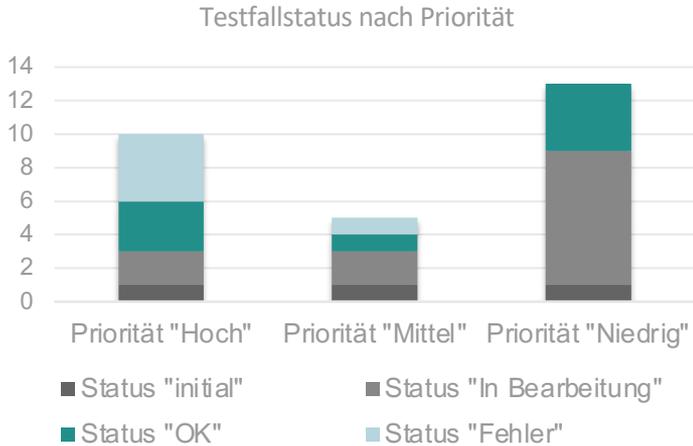


Die Dokumentation ist nicht Aufgabe des Testteams. Aber Voraussetzung, damit die Testphasen vorbereitet & durchgeführt werden können.

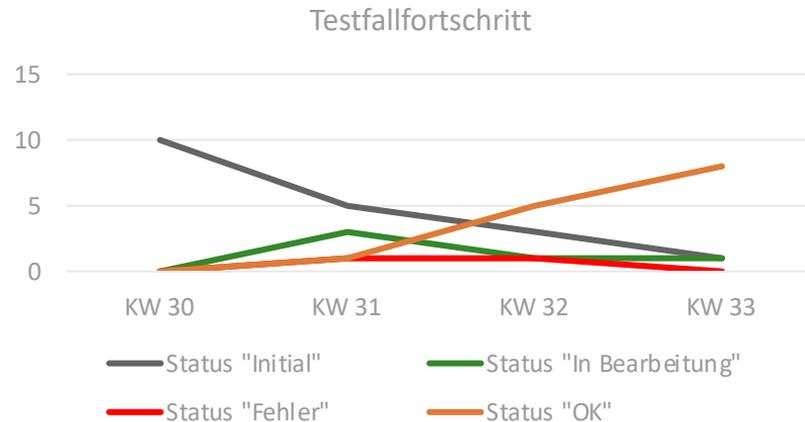
- **Testfall-Abdeckung:** Erzeugte vs. benötigte Testfälle (automatisierte und/oder manuelle) je Bereich
- **Testdatenverfügbarkeit:** Erzeugte vs. benötigte Testdaten je Bereich
- **Testaufwandschätzung:** Aufwand für Test-Durchführung
- **Bereitschaft Testumgebung:** ist Testsystem vorbereitet?
- **Testressourcenallokation:** Personal, Werkzeuge und anderen Ressourcen bereit?

Testfall	Fehler	
Testfallstatus nach Priorität	Fehlerstatus nach Kritikalität	Fehlerfortschritt
Testfortschritt (Liniendiagramm)	Offene Fehler nach Kritikalität in Bezug zur Testfall Priorität	Fehler nach Offen/Geschlossen und Kritikalität
Testfortschritt (Quotient)	Fehlerstatus nach Kategorie (Fehlerverteilung)	Gefundene Fehler nach Kritikalität im Zeitverlauf

Reporting ist status- bzw. fortschrittsbasiert und lässt sich pro Testpaket, Testplan und Teststufe aggregieren.



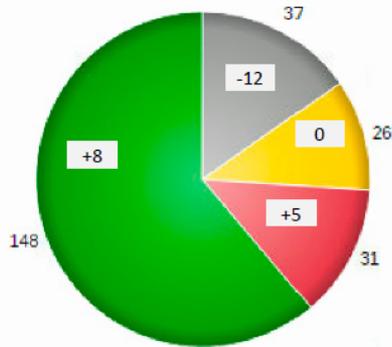
		Offene Fehler nach Kritikalität		
		Hoch	Mittel	Niedrig
Testfall Priorität	Hoch	5	1	3
	Mittel	2	2	1
	Niedrig	3	6	1



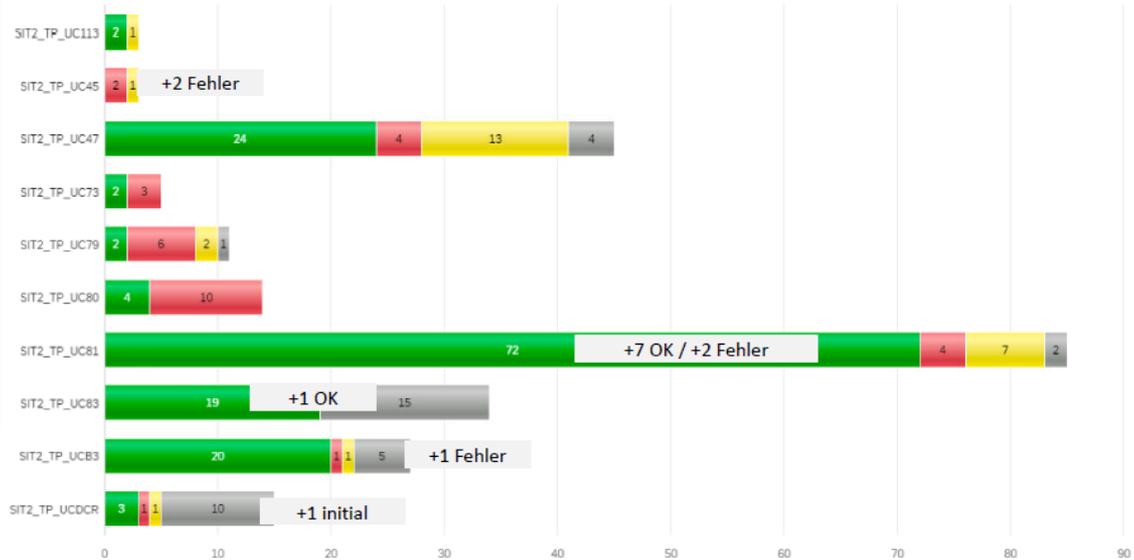
## #2: AUSFÜHRUNG BESTEHENDE TESTPAKETE

Info: 1 Testpaket enthält n Testfälle

Vs.  
Vortag



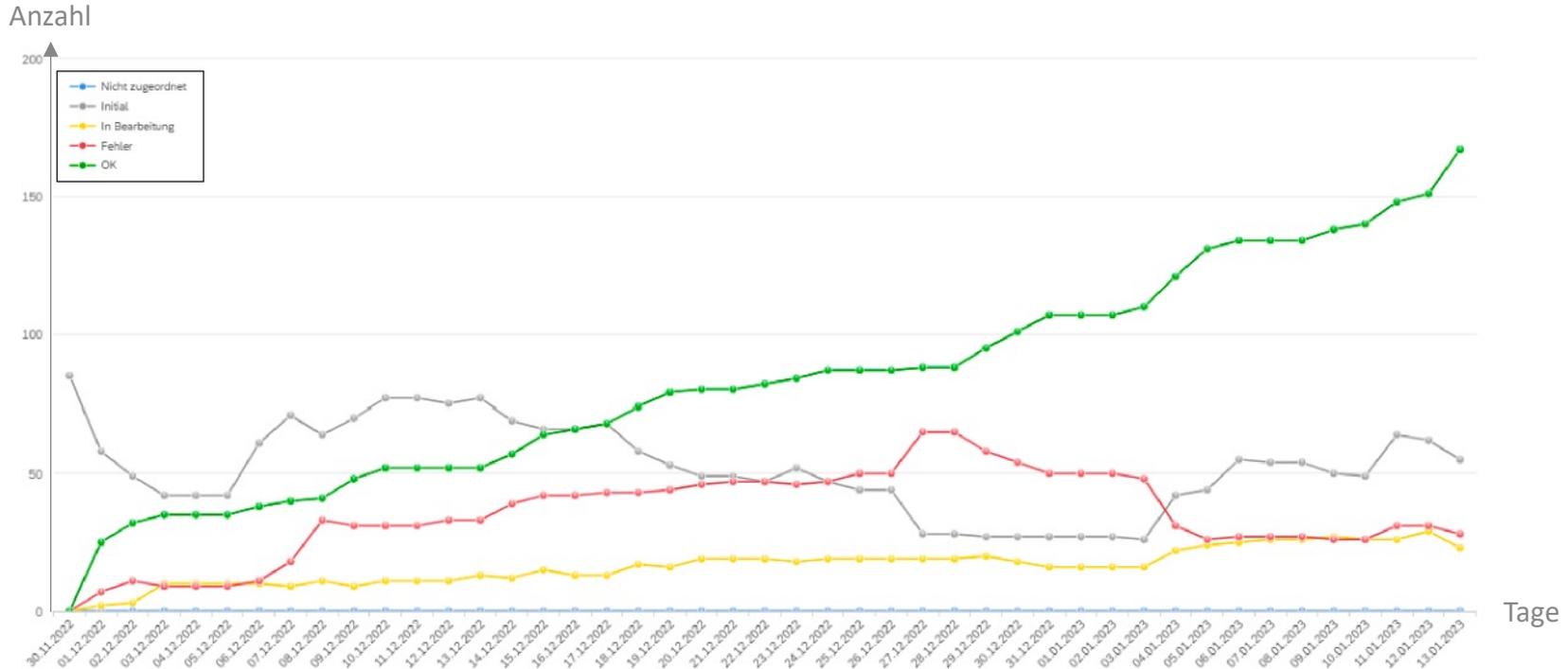
	23.12	10.01	delta
Initial	51	37	-14
In Process	18	26	+8
Errors	46	31	-15
OK	85	148	+63
<b>Total</b>	<b>200</b>	<b>242</b>	<b>+42</b>



Testfälle

# Reporting - Testdurchführung

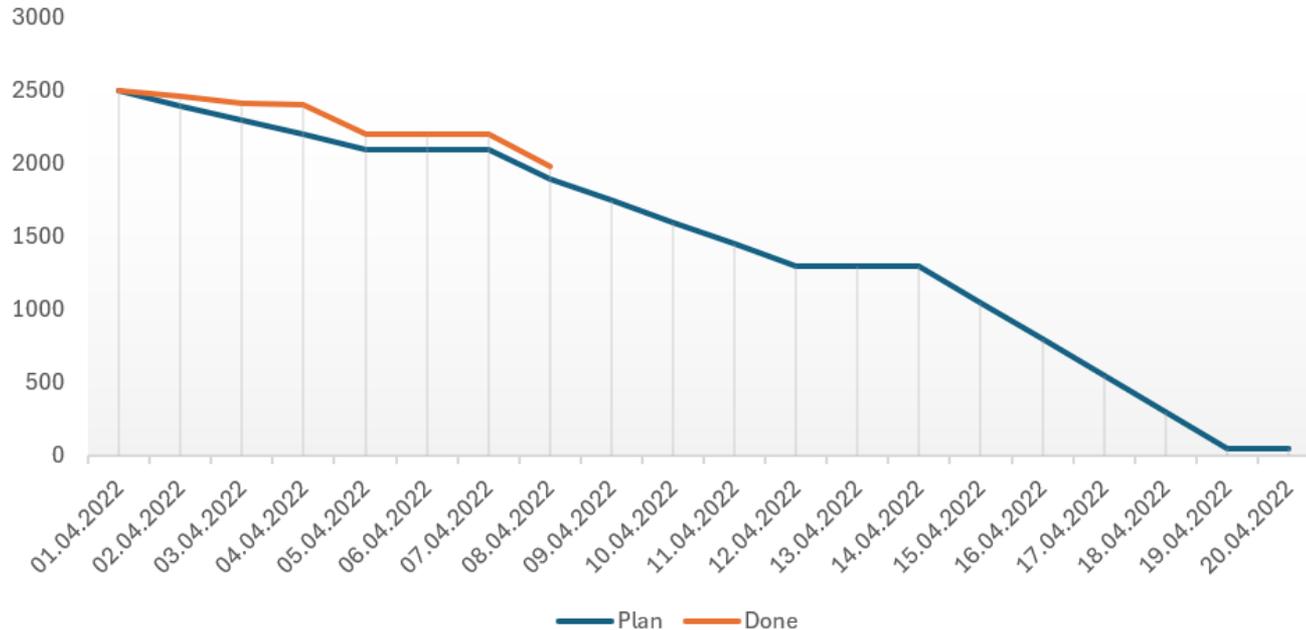
## Testfortschritt im zeitlichen Verlauf



# Reporting - Testdurchführung

## Testfortschritt – Burndown-Chart

### Durchgeführte Testfälle Plan/IST



# Reporting - Demo

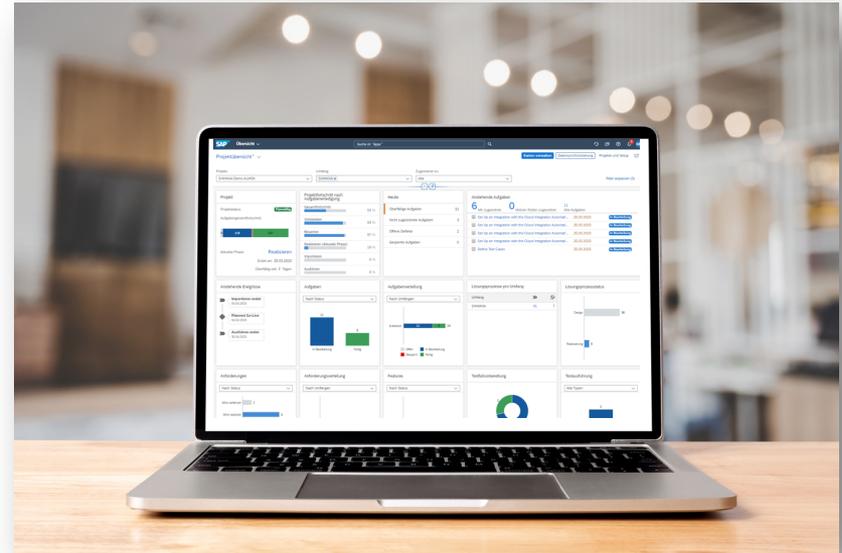


## Demo: Reporting in CloudALM



### [Cloud ALM Demo Tenant \(sap.com\)](https://support.sap.com/en/alm/demo-systems/cloud-alm-demo-system.html)

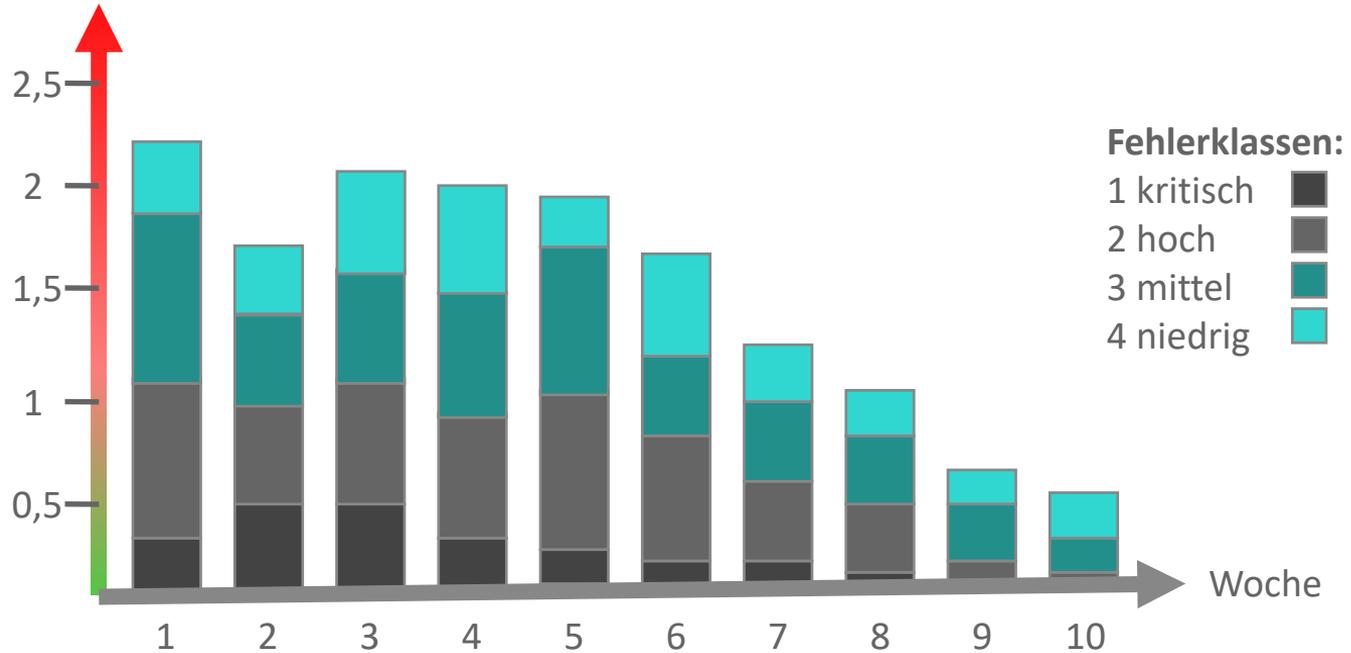
<https://support.sap.com/en/alm/demo-systems/cloud-alm-demo-system.html>



# Reporting - Testdurchführung

## Fehlerfindungsrate als Indikator für das Testende

Fehler / Teststunde



## Nachmittag

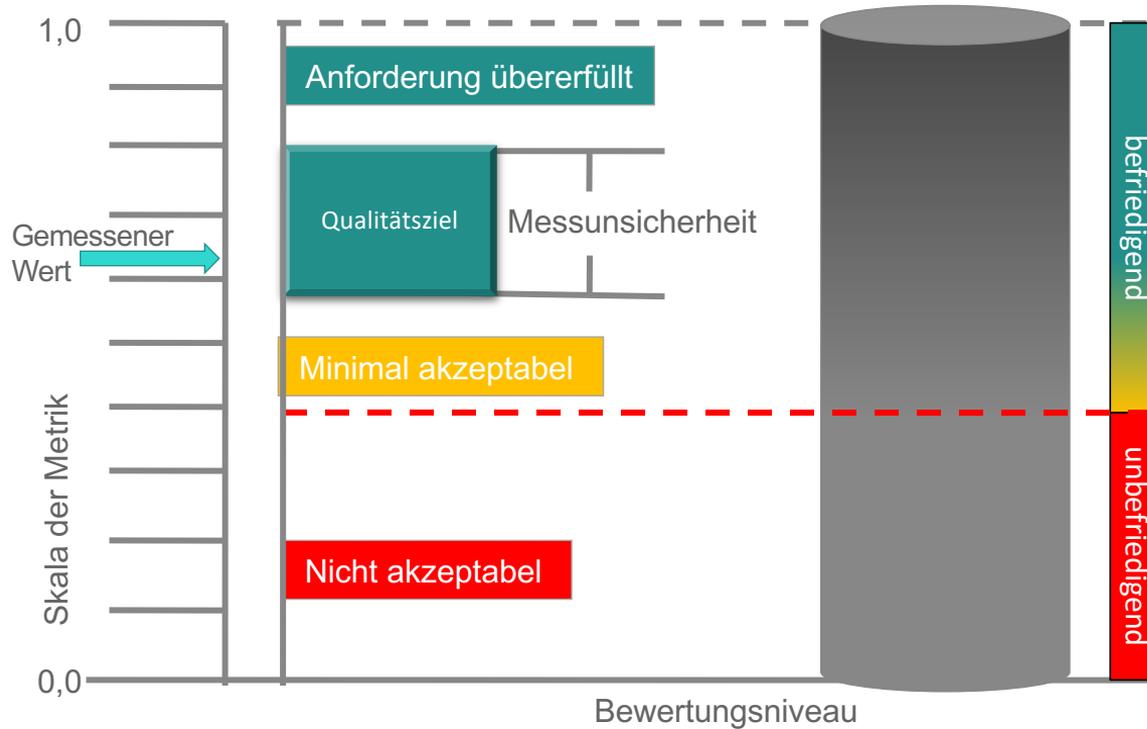
### Test-Reporting

Einleitung

Reporting bei Test-Vorbereitung und -Durchführung

Testabschluss & Testdokumentation

Fazit & Tipps aus der Praxis



# Praxisbeispiel:

## Testmonitoring (Prioritäten und Abnahmekriterien)



Kriterium	Zielwerte
Testabdeckung (bearbeitete Testfälle)	100%: kein Testfall befindet sich mehr im Status „initial“
Positive Testabdeckung Testfälle im Status „OK“ oder „OK mit Vorbehalt“	Sind 100 % der mit „sehr hoher“ und „hoher“ Priorität eingestuften Testfälle positiv getestet? Sind 80 % der mit „mittlerer“ Priorität eingestuften Testfälle positiv getestet? Sind 60 % der mit „geringer“ Priorität eingestuften Testfälle positiv getestet?
Offene Defects nach Priorität „sehr hoch“ und „hoch“	keine offene Defects mit Priorität „hoch“ oder „sehr hoch“
Offene Defects nach Priorität „mittel“ und „niedrig“	Defects mit Priorität „mittel“ und niedrig sind analysiert und Handlungsempfehlung liegt vor

## Warum?

- Dokumentation des Testergebnisses
  - Test-Ergebnisse und Schlussfolgerungen
- Entscheidungsunterstützung
  - Einschätzung Test-Status und Risiken
- Rückblick und Verbesserung
  - Reflexion Test-Prozess

## Inhalte?

- Zusammenfassung
- Testergebnisse
- Fehlerbericht
- Test-Metriken
- Erkenntnisse und Empfehlungen
- Zusätzliche Anhänge

## Revisions sichere Dokumentation für die Wirtschaftsprüfer

### Gründe

- Compliance und Gesetzeskonformität
- Risikomanagement und Haftungsbegrenzung

### Anforderungen

- Unveränderlichkeit und Nachvollziehbarkeit
- Langfristige Archivierung und Zugriffssteuerung
- Compliance mit internationalen Standards und Best Practices

1	Dokumentenhistorie .....	2
2	Testplandetails .....	4
2.1	Allgemeine Daten .....	4
2.1.1	Kopf .....	4
2.1.2	Prozessdokumentation .....	4
2.1.3	Systemlandschaft .....	4
2.1.4	Erstellung/Änderung .....	6
2.2	Einstellungen .....	7
2.2.1	Freigabestatus .....	7
2.3	Termine .....	7
2.3.1	Ebene der Datenerfassung für Plan .....	7
2.3.2	Plandaten und -aufwand .....	7
2.3.3	Istdaten und -aufwand .....	8
2.4	Testpaketdetails .....	8
2.5	Status der Testplanausführung .....	8
3	Testpaketdetails .....	9
3.1	Testpaket IT2_X02A_02CA .....	9
3.1.1	Allgemeine Daten .....	9
3.1.2	Erstellung/Änderung .....	9
3.1.3	Istdaten und -aufwand .....	9
3.1.4	Erstellung/Änderung .....	9
3.2	Einstellungen .....	9
3.2.1	Freigabestatus .....	9
3.3	Termine .....	9
3.3.1	Ebene der Datenerfassung für Plan .....	9
3.3.2	Plandaten und -aufwand .....	9
3.3.3	Istdaten und -aufwand .....	9



Eine frühzeitige Integration der Wirtschaftsprüfer sollte erfolgen, damit Anforderungen aufgenommen und berücksichtigt werden können.

## Nachmittag

### Test-Reporting

Einleitung

Reporting bei Test-Vorbereitung und -Durchführung

Testabschluss & Testdokumentation

Fazit & Tipps aus der Praxis



## Kommunikation

- Relevante **Stakeholder** und Entscheidungsträger frühzeitig identifizieren
- **Anforderungen** und Erwartungen abstimmen
- **Meeting-Strukturen** für Status-Updates planen



## Übersicht

- Klare und präzise Kommunikation: **Verständlich** und **aussagekräftig**
- Anpassung der Reports an **Bedürfnisse der Zielgruppe**
- Planmäßige **Aktualisierung**, während Testphase höhere Frequenz



## Integration

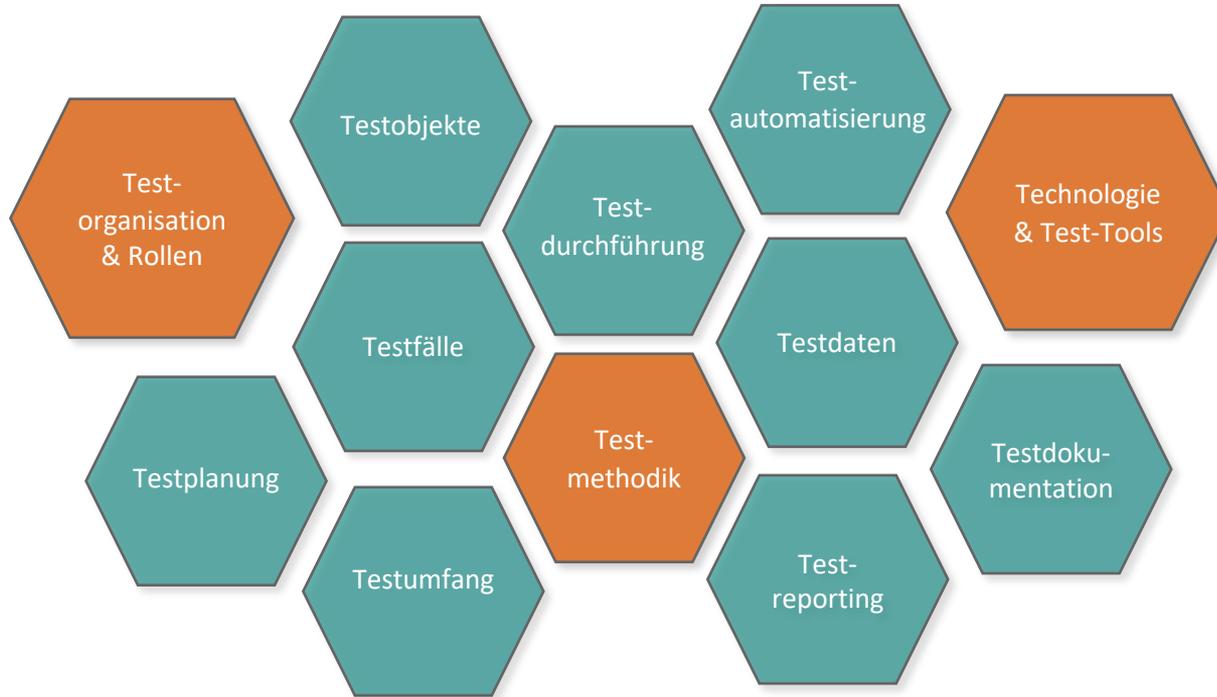
- Reporting in **Software-Entwicklungsprozess** integrieren
- Möglichst **automatisierter** Aufbau des Reportings (Geschwindigkeit & Fehler reduzieren)
- **Erkenntnisse** aus vorherigen Testberichten für zukünftige Projekte **nutzen**

# Offene Fragen zum Test-Reporting?



# SAP Testmanagement Landkarte

## 12 Handlungsfelder





Herzlichen Dank für Ihre  
Aufmerksamkeit!



**Kontakt**

SKYWAY GmbH  
Freundchenstraße 33  
67269 Grünstadt

&

Synaworks GmbH  
Hauptstraße 25  
69117 Heidelberg

