



**NOSERENGINEERING**  
WE KNOW HOW




The Noser Way of Testing



# The Noser Way of Testing

Immer wieder wird nach einem «beispielhaften Testprozess» mit zugehörigen Artefakten gefragt. Um diese Nachfragen zu beantworten, entstand dieses Booklet.

Dieses Booklet soll Ihnen vermitteln, wie Sie eine Teststrategie, eine Testidee, ein Testkonzept und eine Vision entwerfen und definieren können. Hierbei sind die langjährigen Erfahrungen von Noser Engineering eingeflossen. Diese sind konform mit der ISO-Norm 29119.

Zunehmend werden Projekte nach agilen Projektmethoden (z.B. SCRUM oder Kanban) durchgeführt, wo das Testen erfahrungsgemäss oft vernachlässigt wird. Mit dem Piktogramm  werden Abschnitte hervorgehoben, welche aufzeigen, wie das Testen auch agile Projekte erfolgreich unterstützen kann.

© 1. Januar 2016, Autor Dominique Daniel Portmann.  
7<sup>te</sup> Auflage, neu überarbeitet und an ISO 29119 angepasst.

**Approved by Noser™**  
Test Validation

**Wir wissen wie:**

The Noser Way of Testing

**Wir beherrschen die Tools:**

MS TFS\*, HP ALM\*, LabView\*, TestStand\*, JIRA, Tosca, Ranorex\*, Selenium (\*Wir sind Partner)

**Wir haben über 150 erfahrene Mitarbeitende:**

Noser Engineering AG arbeitet seit 1984 am Erfolg ihrer Kunden mit

**Wir halten Testen einfach**

**Ihr Ansprechpartner:**

Dominique Daniel Portmann  
Inf. Ing. HTL MAS MSG  
Leiter Business Unit Test Engineering

dominique.portmann@noser.com  
+41 41 455 66 11



# Warum unterstützt Testen das Management?

Das systematische Testmanagement hat sich von der reinen Prüfdisziplin zum Steuerungsinstrument für nachhaltige Systemqualität weiterentwickelt.

Mit Hilfe des Testmanagements lassen sich Probleme und unnötige Kosten vermeiden. Fehler können zu einem frühen Zeitpunkt des Software-Lebenszyklus erkannt und rechtzeitig behoben werden. Mit dem Testmanager oder der Testmanagerin hat das Management und die Projektleitung eine kompetente Ansprechperson, die faktenbasierte Entscheidungsgrundlagen erarbeitet.

Dies schafft Transparenz, beschleunigt den Entscheidungsprozess und sichert die Entscheidungen ab.



*Es bildet speziell in agilen Projekten mit kurzen Zyklen einen entscheidenden Erfolgsfaktor.*

*Professionelles Testmanagement verspricht Wettbewerbsvorteile und Kostenersparnis.*

*Getestet wird immer:  
Entweder geplant  
und beauftragt oder  
ungeplant durch  
die Kunden und  
Benutzerinnen und  
Benutzer.*

*Entscheiden Sie,  
wer testet und wo  
die Testresultate  
veröffentlicht  
werden:  
intern oder in der  
Tageszeitung.*

## Ihr Nutzen

Das Testbudget kann **eingeplant** werden, die Kosten für Nachbesserungen sind nicht abschätzbar.

Fehler kosten Geld. Testen stellt sicher, dass Fehler früher und systematisch erkannt werden. Je früher ein Fehler erkannt wird, desto geringer sind dessen Behebungskosten.

### Mit Testen sparen Sie Geld

Ein konsequent gelebter Testprozess hat Einfluss auf die genaue und systematische Erfassung und Beschreibung der Anforderungen — so kann **schneller entwickelt** werden.

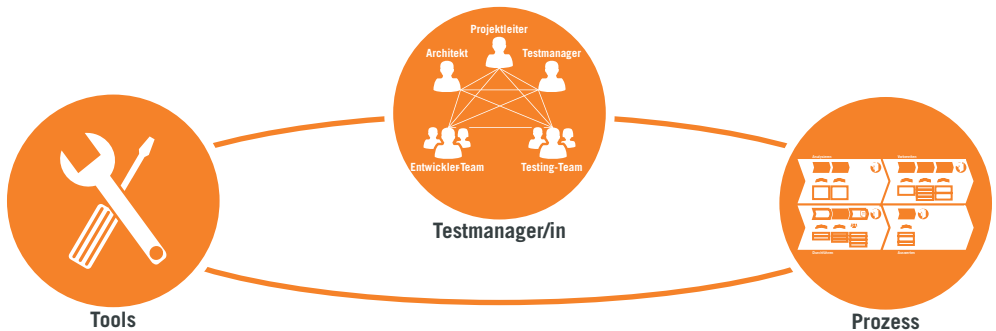
Gelebtes Testmanagement stellt sicher, dass **Fehler, Änderungen und Neuanforderungen** klar unterschieden werden. Dies schafft Klarheit, auch bei den Kosten.

Testmanagement ist auch Führungsarbeit. Gerade in agilen Prozessen bilden Testresultate oft die einzige verlässliche Entscheidungsgrundlage.



*Moderne (agile) Prozesse setzen auf das Testen als Mittel zur Fortschrittskontrolle.*

# Die drei Erfolgsfaktoren



## Schlüssel zum Erfolg

### Tools

- Einfach zu bedienen, gut gepflegt
- Zugewiesene Verantwortlichkeit (für Pflege)
- Müssen von allen Projektbeteiligten akzeptiert sein
- Müssen pro Projekt individuell anpassbar sein
- Benötigen offene und einfache Schnittstellen
- Können einfach dem Prozess angepasst werden

### Prozess

- Muss einfach, verständlich und übersichtlich sein
- Muss gelebt und gepflegt werden
- Braucht eine verantwortliche Person (Pflege, Schulung, Durchsetzung)
- Muss breit akzeptiert sein
- Muss die nötigen Freiräume geben
- Muss wichtige Details definieren
- Soll das Projekt unterstützen und nicht behindern

### Testmanagerin, Testmanager oder Testmaster

- Gute, solide Ausbildung in Anwendung (fachspezifisch) als auch in der Entwicklung (adäquat zu den Entwicklern)
- Erfahrung in der Projektleitung
- Fundierte Testausbildung (z.B. ISTQB) und Erfahrung
- Softskills: kommunikativ mit Menschenkenntnis und Flair für Leute und Situationen
- Leidenschaft und Motivation

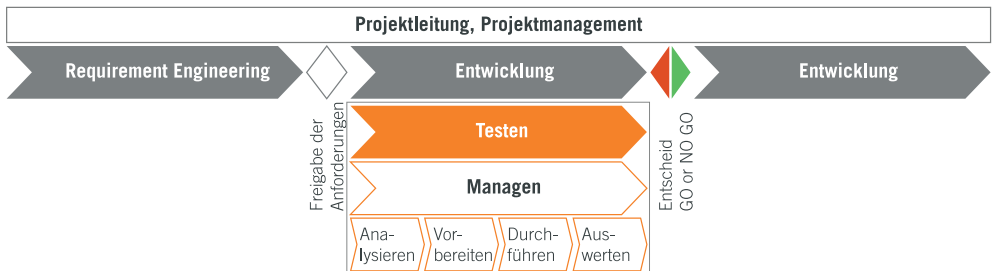


*In agilen Projekten wird den Projektbeteiligten keine fixe Rolle zugeteilt. Die Teams organisieren sich selbst und handeln interdisziplinär.*

*Für Testaspekte ist das gesamte Team verantwortlich. So ist beispielsweise auch eine Entwicklerin oder ein Entwickler dafür besorgt, dass die Klassen/Komponenten mit UnitTests überprüft werden.*

*Unterstützend sollen Personen mit speziellem Testwissen (z.B. Testautomation, Testarchitektur, Testdaten) in SCRUM-Teams «eingebettet» werden.*

## Testen im Projekt



Das Testen beginnt bereits mit der Abnahme (resp. Freigabe) der Anforderungen. Dabei überprüft das Testteam, ob der gewünschte «Endzustand» klar, vollständig und widerspruchsfrei beschrieben, messbar und somit testbar ist.

Am besten erfolgt diese Abnahme durch ein Team, welches aus Entwicklung, Architektur, Testengineering, Sachverständigen sowie evtl. einer Vertretung aus Support und Anwendung besteht.

---

Das Testteam stellt die «Testbarkeit» der Anforderungen sicher.

---



*In agilen Projekten wird das Produktinkrement nach jeder Iteration (vgl. Sprint bei SCRUM) abgenommen. Ziel ist es, nach jeder Iteration ein potenziell auslieferbares Produkt zu erstellen.*

*Die Anforderungen sind bei Projektbeginn nicht in Stein gemeißelt, sie werden nach und nach verfeinert. Gerade hier stellt das Testen sicher, dass frühzeitig klare Akzeptanzkriterien definiert werden.*

## Der Einfluss von Projektmethoden

	Wasserfall	RUP	SCRUM
Testplanung	Viel Zeit Testen als Phase	Eigene Disziplin	Rollend
Entwicklung versus Testteam	Zwei Welten	Getrennt	Integriert
Testinfrastruktur	Viel Zeit für Vorbereitung	Kann vorbereitet werden	Gleichzeitiges Testen und Infrastruktur bauen
Change Request	Grosser Impact CR kann verloren gehen	Iterativ	Muss sehr gut verwaltet (Backlog) und kommuniziert werden Testtasks werden laufend eingebracht
Testkonzept	Hat Raum und Zeit	Vorgesehenes Artefakt	Kommt öfters zu kurz

## Aus der Strategie ergibt sich das Testkonzept

---

«Die Praxis soll das Ergebnis des Nachdenkens sein, nicht umgekehrt.» (Hermann Hesse)

---

Grundsätzlich bestimmt die Strategie, wie weit das Risiko durch das Testen minimiert werden soll. So ergibt sich das «WAS» soll getestet werden und das «WIE» und «Wie intensiv» muss getestet werden. In Anlehnung an das oben stehende Zitat dient die Teststrategie dazu, sich vorgängig Gedanken zum Testvorhaben zu machen und diese verbindlich festzuhalten.

Das Testkonzept ermöglicht es, das Testen risikobasiert zu Priorisieren und zu Gruppieren. Es bildet die Basis für ein erfolgreiches Testen.

# Ein Testkonzept in 15 Minuten

## Teststrategie in 15 Minuten

<b>Was?</b> Welche Testaktivitäten sind notwendig?	<b>Wie?</b> Wie intensiv muss getestet werden?	<b>Wann?</b> Wie viel Zeit/Budget steht zur Verfügung?
<b>Qualitätsanforderungen</b> Qualitätsmerkmale/Qualitätsziele, welche messbar sein sollen	<b>Kritikalität</b> Projekt, Teilprojekt, Testobjekt, Risikoanalyse	<b>Termin/Budget</b> Dringlichkeit, Gesamtbudget, Ressourcen
Ableitung relevanter Testaufgaben	Bewertung und Ableitung angemessener Testmethoden	Planung der Tests

### In einem ersten Schritt gilt es:

- Das akzeptierbare Risiko zu bestimmen
- Die Prioritäten festzulegen
- Die einzelnen Testobjekte zu definieren und abzugrenzen
- Das notwendige Testvorgehen vorzugeben
- Pro Testobjekt die erwarteten Testtypen und Testarten zu definieren
- Die erforderliche Testtiefe festzulegen
- Führungskonzepte zu etablieren
- Eine angepasste Testorganisation mit verbindlichen Verantwortlichkeiten festzulegen

---

Ein beispielhaftes Testkonzept ist im Anhang beschrieben.



*Auch in SCRUM-Projekten ist ein Testkonzept nötig. Damit es vor Entwicklungs- und Testbeginn vorliegt, müssen wichtige Testaspekte vorgängig mit dem Product Owner geklärt werden. Hier ist das Testmanagement gefordert, denn oft fehlt es den Product Ownern und SCRUM-Mastern an spezifischem Test- und Testmanagementwissen.*

*SCRUM berücksichtigt die Testaspekte zu wenig. Daher ist es wichtig, vor den eigentlichen Sprints zu prüfen, ob ein – an SCRUM angepasstes – Testkonzept besteht.*



## Welche Testziele haben wir?

---

«Testen unterscheidet sich von Experimentieren dadurch, dass es beim Testen eine Erwartung gibt, die belegt werden soll, während das Ergebnis beim Experimentieren offen ist oder nur vermutet werden kann.» (Dominique Portmann)

---

Aufgrund einer Risikobeurteilung gibt ein Testkonzept Auskunft, wie umfassend und mit welchen Schwerpunkten getestet werden soll. Im Verlauf der Entwicklung verschiebt sich der Testfokus. Das Erstellen eines Testkonzeptes gibt die Gelegenheit – und auch die Chance – sich systematisch mit den verschiedenen Aspekten auseinanderzusetzen. Dazu müssen viele Fragen gestellt und beantwortet werden. Nachfolgend eine mögliche Auswahl:

- Soll zugunsten einer kurzen Testdurchlaufzeit auf ein tiefes Testen verzichtet werden?
- Wie relevant ist die Sicherheit, benötigt dieser Aspekt zusätzliche Testfälle?
- Wie wichtig sind Bedienbarkeit, Ergonomie sowie Aufbau und Farbgebung der Menüstruktur?
- Benötigt es zusätzliche «sehr kreative Spezialfälle»?
- Wie wichtig sind Performance, Geschwindigkeit, Wartezeiten und Verfügbarkeit etc.?
- Was passiert im Fehlerfall, was ist der «WORST CASE»? (Menschenleben, Gesundheit, Reputation)
- Wie ist die Haftung und Garantie geregelt, was passiert im Fehlerfall?
- Wie lange «lebt» das Produkt?
- In welchem Umfeld wird das Produkt eingesetzt?
- Wie «komplex» ist das Produkt?
- Was kostet das Produkt?
- Gibt es Gesetze, Vorschriften, Normen?
- Welchen «Standard» haben vergleichbare Produkte?
- In welcher «Liga» wird das Produkt lanciert?

---

«Jedes Konzept ist besser als kein Konzept!»

---

Das Testkonzept beschreibt, wie sich mit zunehmendem Entwicklungsfortschritt der Testfokus verändert. Am Anfang ist es wichtig, mit einfachen Tests zu belegen, dass neu entwickelte Funktionen die richtigen Resultate liefern und der gewählte Architektur- oder Entwicklungsansatz erfolgreich ist. Mit zunehmendem Projektfortschritt nimmt der Funktionsumfang und damit auch der Testumfang zu.

Ohne klares Testkonzept, welches risikobasiert den Testaufwand optimal vorgibt, kann ein Testvorhaben nicht erfolgreich geführt werden.



*Gerade im agilen Vorgehen liegt hier eine der grössten Herausforderungen, denn für jede Iteration ist grundsätzlich mehr Testaufwand einzuplanen, der Testaufwand wächst exponentiell! Ein möglicher Ausweg bildet die Testautomatisierung. So empfiehlt es sich, frühzeitig das Automatisieren einzuplanen.*

## Qualitätsmerkmale nach ISO 25000

### Funktionale Eignung

- Angemessenheit
- Richtigkeit
- Interoperatilität
- Ordnungsmässigkeit

Ist die geforderte Funktionalität in der Software gegeben?

### Zuverlässigkeit

- Reife
- Fehlertoleranz
- Wiederherstellbarkeit

Wie zuverlässig arbeitet die Software?

### Benutzbarkeit

- Verständlichkeit
- Erlernbarkeit
- Bedienbarkeit

Ist die Software einfach bedienbar?

### Leistungseffizienz

- Zeitverhalten
- Verbrauchsverhalten

Wie effizient arbeitet die Software?

### Wartbarkeit

- Analysierbarkeit
- Modifizierbarkeit
- Stabilität
- Prüfbarkeit
- Anpassbarkeit

Wie leicht lässt sich die Software modifizieren?

### Übertragbarkeit

- Anpassbarkeit
- Installierbarkeit
- Konformität
- Austauschbarkeit

Wie leicht lässt sich die Software auf ein anderes System portieren?

### Sicherheit

- Zugriffssicherheit
- Datenverschlüsselung

Wie sicher sind unsere Daten und Programme vor nicht autorisiertem Zugriff?

### Kompatibilität

- Austauschbarkeit
- Erweiterbarkeit
- Abwärtskompatibilität

Wie kompatibel ist die Software beim Austausch und der Verarbeitung von Daten mit und von anderen Systemen?

*Norm ISO 25000 ersetzt ISO 9126*

# Was soll getestet werden?

---

«Wie wird ein Elefant aufgegessen? Biss für Biss, Stück für Stück» Quelle unbekannt

---

Bei grösseren Testvorhaben macht es Sinn, einzelne überschaubare und somit test- und fñhrbare Stñcke resp. Testobjekte zu bilden. Testobjekte können aufgrund verschiedenster Kriterien gruppiert werden:

- Funktionale Aspekte
- Architektur-Layer
- Aspekte der Zuständigkeit von Schlüsselpersonen
- Abhängigkeiten von Umsystemen
- Gruppen von unterschiedlichen nichtfunktionalen Tests
- Manuelle und automatische Tests
- Grenzwert- und Negativtests
- Quick- oder Smoketests

Die Testobjekte sind dann gut gewählt, wenn sie untereinander aus-  
gewogen sind und unabhängig voneinander getestet werden können.

## Testvielfalt

Tests müssen nicht nur gut priorisiert sein, sondern sie müssen auch in die «Breite» gehen. Daher muss ein Testkonzept verschiedene Testaspekte berücksichtigen und in die Testplanung einfließen lassen.

### Um was geht es?

### Nutzen für das Projekt

#### Formale Eintrittskriterien

Sind alle Vorbereitungen getroffen und belegt?  
Liegen die Daten der Unit-Tests (resp. der vorgängigen Teststufen) vor?

Die nötigen Voraussetzungen zum Starten der Tests sind gegeben, unnötige Tests und Reibungsverluste können so deutlich verringert werden.

#### Vitalität

Mit einfachen und schnellen Tests wird überprüft, ob das System lebt und grundsätzlich testbar ist.

Dank einer schnellen Übersicht kann entschieden werden, ob ein frühzeitiger Testabbruch notwendig ist und ob sich der «grosse Testaufwand» lohnt.

## Um was geht es?

## Nutzen für das Projekt

### Neue Funktionen

Neue Funktionen werden schnell und zeitnah getestet.

Neue Funktionen und/oder unbekannte Technologien werden schnell nachweisbar überprüft.

### Gefixte Bugs

Fehler aus früheren Tests haben eine hohe Managementattention und binden dadurch viele Ressourcen. RETESTS von reparierten BUGs erhalten daher eine hohe Priorität.

Offene Fehler behindern den Projektfortschritt und binden Ressourcen.  
Eine schnelle Rückmeldung unterstützt den Projektfortschritt.

### Retest/Regression

Funktioniert es noch?

Regelmässige Regressionstests erhöhen die Systemstabilität.

### «Nur leicht modifiziert»

Verhindern von ungewollten Code-änderungen (Kollateralschäden)

Es wird sichergestellt, dass bereits getestete Funktionen stabil bleiben. Es kann sichergestellt werden, dass die Software-Architektur nicht verletzt wird.

### Monkey-, Negativ- und Grenzwerttests

Gewohnte Pfade werden verlassen, die Testabdeckung wird erweitert.

Die Erweiterung der Schönwetter-Testfälle um weitere Tests erhöht die Qualität entscheidend.

### Unvorhergesehenes

Reserve für unerwartete Fragen aus dem Management oder nicht vorgesehene Entwicklungen einplanen

Oft binden unerwartete Ereignisse oder Fragen aus dem Management Ressourcen, welche anderweitig eingeplant sind. Daher soll das Testmanagement hierfür in jedem Testdurchgang genügend Zeit einplanen. Dies ermöglicht – gerade in hektischen Phasen – ein souveränes Auftreten.

---

Eine gut ausgewählte Testvielfalt reduziert das Risiko.

## Der Testaufwand wächst mit jeder Iteration

Mit jeder Iteration stehen neue Funktionen zur Verfügung. Mit jeder weiteren Iteration steigt die Zahl der gefundenen Fehler, die erneut getestet werden müssen. Zusätzlich steigt in jeder Iteration der Anteil an bestehenden Funktionen, die wiederholt (Regression) getestet werden müssen.

	1 Iteration	2 Iteration	3 Iteration	4 Iteration	5 Iteration
Automatisiertes Testen					
					Test A
				Test A	Test B
			Test A	Test B	Test C
Zusätzlich Negativ- und Grenzwertest		Test A	Test B	Test C	Test D
Neue Features werden getestet (Schönwetter)	Test A	Test B	Test C	Test D	Test E

Ungetestet
  Retest
  Neue Tests, neue Funktionen
  Getestet, automatisiertes Testing



Um den manuellen Testaufwand zu verringern, müssen die Entwickler in jeder Iteration auch Tests automatisieren und das Testframework erweitern.

*In der «Definition of Done» bildet die Testbarkeit der Funktionalität einen zwingenden Bestandteil. Ebenso muss sichergestellt werden, dass in jeder Teststufe das MAXIMUM an Tests durchgeführt wird. Speziell muss das Potenzial von Unit-Tests gezielt eingeplant und ausgeschöpft werden. Die Resultate der Unit-Tests bilden das formale Eintrittskriterium in die nächste Teststufe.*

*Eine durchdachte Testarchitektur ermöglicht einen hohen Grad an automatisierten Tests. Der Bau sowie die Pflege der Testarchitektur muss Teil der Projektplanung sein und in den Produkt- und Sprintbacklogs als Aufgaben enthalten sein.*

*Das Anwenden von strukturiertem, explorativen Testen (SET) hat sich im SCRUM-Umfeld sehr bewährt.*

## RPI: Risiko Prioritäts Index – Most Critical first

Der Testaufwand wächst exponentiell. Dank risikobasiertem Testen kann sichergestellt werden, dass sich der Testaufwand in Grenzen hält und dennoch genügend tief und breit getestet wird. Gerade in hektischen Phasen bleibt nie genügend Zeit für das Testen. Die systematische Priorisierung bildet die Grundlage zur massvollen Reduktion der notwendigen Tests. Für die Priorisierung müssen die Anforderungen vorgängig systematisch nach drei Kriterien bewertet werden:

- **Business Relevanz:** Auswirkung im Fehlerfall, wie schlimm ist ein Fehler? (3 = am schlimmsten)
- **Auffindbarkeit:** Wie offensichtlich und schnell kann ein Fehler entdeckt werden (3 = schwer und aufwendig erkennbar)
- **Komplexität:** Wie komplex ist die Anforderung und die dazu notwendige Umsetzung und Realisierung? (3 = sehr komplex)

Das Produkt dieser drei Kriterien führt zum «Risiko Prioritäts Index (RPI)». Der so ermittelte RPI ist ein Führungsinstrument und bildet die Grundlage für die konsequente Bestimmung der Testtiefe, des Testumfangs sowie der Testreihenfolge.

Produkt	Priorität	RPI	Testumfang, Testtiefe
27	very high	1	alles in jeder Iteration, sehr gründlich und sehr tief
18			alles pro zwei Iterationen, gründlich und tief
12	high	2	alles pro zwei bis drei Iterationen, wechselnd gründlich/tief
9			wesentliche Abdeckung, in mehreren Iterationen, gründlich und tief
8			wesentliche Abdeckung über mehrere Iterationen, Wechsel Tiefe/Breite
6	medium	3	wesentliche Abdeckung verteilt auf mehrere Iterationen
4			Abdeckung wesentlicher Pfade, alternative Pfade wechselnd
3			Abdeckung wesentlicher Pfade, alternative Pfade stichprobenartig, Wechsel
2	low	4	nur Abdeckung, wesentliche Pfade
1	very low	5	stichprobenartige Abdeckung wesentlicher Pfade nicht testen

---

Risikobasiertes Testen: Das Risiko bestimmt die Testtiefe und -breite.

---

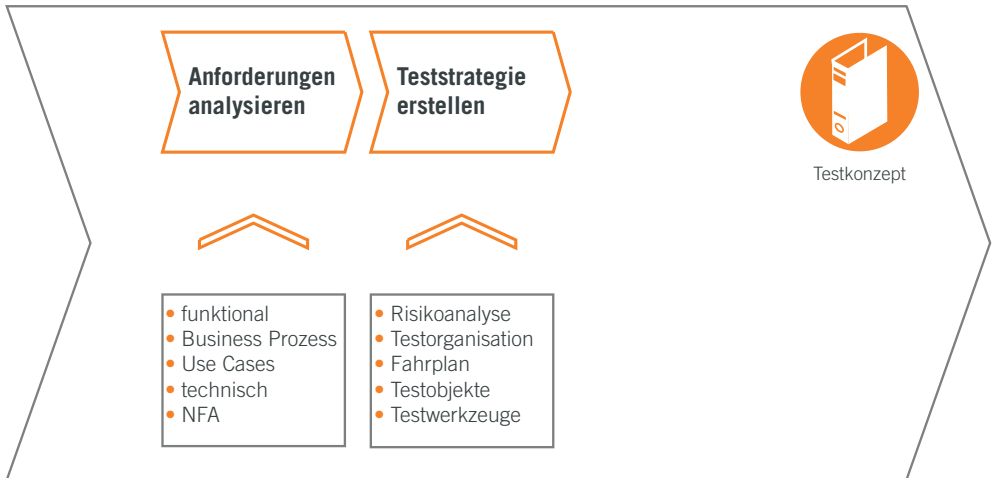
## Kennzahlen

Führen bedingt Kennzahlen. Das Testmanagement soll der Projektleitung jederzeit wichtige und verlässliche Kennzahlen liefern.

Für jede Testphase sind unterschiedliche Aussagen notwendig.

	Analysieren	Vorbereiten	Durchführen	Auswerten
<b>Worum geht es?</b>	<ul style="list-style-type: none"> <li>Was, warum?</li> <li>Risiko?</li> <li>Wie viel, wie tief?</li> </ul>	<ul style="list-style-type: none"> <li>Testanalyse</li> <li>Testdesign</li> <li>Testplanung</li> </ul>	Dokumentiert und nachvollziehbar Resultate festhalten	<ul style="list-style-type: none"> <li>Resultate darstellen</li> <li>Resultate zusammenfassen</li> <li>Ergebnis beurteilen</li> </ul>
<b>Was entsteht?</b>	Testkonzept Risikoanalyse Testobjekte Testmethode Testorganisation	<ul style="list-style-type: none"> <li>Testszenarios</li> <li>Testdaten</li> <li>Testumgebung</li> <li>Testtools</li> <li>Testdrehbuch</li> <li>Tests</li> <li>Testautomation</li> </ul>	<ul style="list-style-type: none"> <li>Fehler</li> <li>Fortschritt</li> <li>Reports</li> <li>Resultate</li> </ul>	<ul style="list-style-type: none"> <li>Testreport</li> <li>Testprotokoll</li> <li>Abnahmebericht</li> <li>Testbericht</li> <li>Management-summary</li> </ul>
<b>Was ist zu beachten?</b>	Ist das Testen risikoadäquat?	<ul style="list-style-type: none"> <li>Fortschritt der Vorbereitung</li> <li>Testabdeckung</li> </ul>	<ul style="list-style-type: none"> <li>Testfortschritt</li> <li>Testabdeckung</li> <li>Fehlerentwicklung</li> <li>Stabilität</li> </ul>	Einfache ansprechende Darstellung, managementtauglich und aussagekräftig
<b>Kennzahlen</b>	Anzahl erwarteter Testszenarien  Erwartete Testabdeckung	<ul style="list-style-type: none"> <li>Anzahl               <ul style="list-style-type: none"> <li>– erwartete TS</li> <li>– erledigte TS</li> </ul> </li> <li>Erreichte Testabdeckung</li> <li>Erwarteter Testaufwand</li> <li>Erwarteter Testdurchlauf</li> <li>Anzahl Testdaten</li> </ul>	Summe der Fehler: <ul style="list-style-type: none"> <li>Anzahl               <ul style="list-style-type: none"> <li>– offener Fehler</li> <li>– durchgeführter Tests</li> <li>– erfüllter Requirements</li> <li>– nicht getesteter Requirements</li> </ul> </li> </ul>	Wie beim Testen, aber mit Verlauf, Entwicklung sowie kurzer Beurteilung
<b>Schlüsselfrage</b>	<b>Was darf es kosten?</b>	<b>Sind wir bereit?</b>	<b>Ist es reif?</b>	<b>Wagen wir es?</b>

# Analysieren



Hier müssen zwei wichtige Aussagen getroffen werden.

## Risiko- und Kostenfrage

Wie viele Testszenarien sind zu erstellen?

Der Testumfang wird durch das Risiko bestimmt und ist im Testkonzept vorgegeben.

## Abdeckung

Welche Testabdeckung ist notwendig?

Auf Basis der bestehenden Anforderungen kann eine erste Abschätzung der zu erwartenden Anzahl an Testfällen gemacht werden.

Als Faustregel kann folgende Annahme getroffen werden:

### Schnell und grob

Pro Anforderung gibt es mindestens einen Testfall

### Schnell und optimistisch

Anzahl Anforderungen \* 3 = Anzahl erwartete Testszenarien

### Vorsichtig

Anzahl Anforderungen \* 20/RPI = Anzahl erwartete Testszenarien



Schlüsselfragen für die Führung

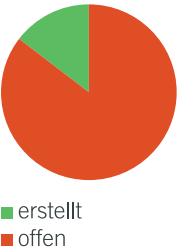
Einfach

Sind alle Anforderungen durch mindestens einen Testfall abgedeckt?

Genau

Ist die Abdeckung risikogerecht?

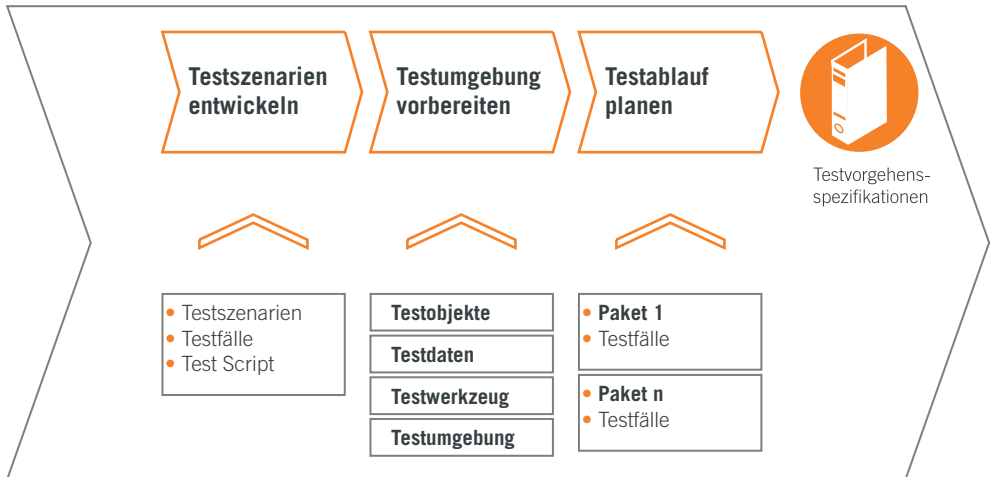
- Strategie
- Risikoanalyse
- Testorganisation
- Testinfrastruktur
- PLACEHOLDER



\*Wie den Faktor bestimmen?

Annahme	Faktor	Bemerkung
Schnell und grob	1	Pro UC ein TC
Einfach und optimistisch	3	TCs für Normalfall, Variante und Ausnahme
Schnell und grob	20/RPI	Abhängig vom RPI

# Vorbereiten



«Tester und Q-Menschen sind nie bereit, sie können die Testvorbereitung sowie die Testinfrastruktur problemlos vergolden.»

Hier wird die Testbereitschaft untersucht und dokumentiert. Es stellt sich die Frage: Sind wir bereit zum Testen?

## Fortschritt

- Wie viele Testscenarien und Testfälle sind zu erwarten?
- Wie viele Testscenarien und Testfälle sind bereits erstellt?
- Wie lange dauert die Vorbereitung noch?
- Ist die Testumgebung bereit?

## Abdeckung

- Für welche Anforderungen wurden bereits Tests mit Testdaten erstellt?
- Ist die Ableitung der Tests aus den Anforderungen sichergestellt (Traceability)?

## Schlüsselfragen für die Führung

### Einfach Genau

Entspricht der Fortschritt dem Zeitverlauf, der Restzeit?  
Entspricht die Testtiefe und Testreihenfolge dem RPI?



In agilen Projekten ist es wichtig, dass eine teamübergreifende, klare «Definition of Done» vereinbart wird. Sämtliche Anforderungen (User-Stories) müssen die «Definition of Done» erfüllen.

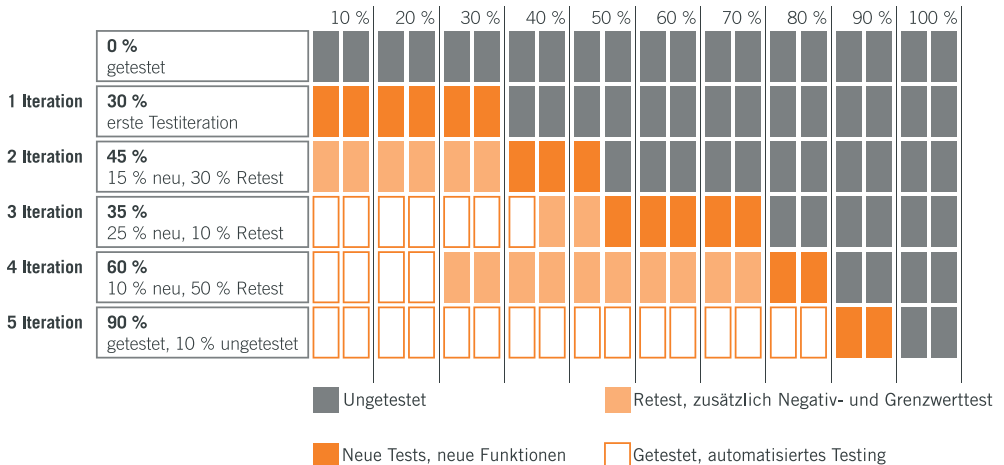
Aus Sicht des Testings sind folgende Kriterien zu beachten:

- Messbarkeit
- Überprüfbarkeit
- Widerspruchsfreiheit
- Verständlichkeit

Pro Anforderung, pro User-Story müssen die spezifischen Akzeptanzkriterien definiert und überprüft werden.

## Vorbereiten: eine gute Planung

Die Testdurchführung muss sorgfältig geplant werden. Wichtige Kriterien für die Planung sind: Risiko (RPI), Testabdeckung, Anzahl an notwendigen Regressionen.

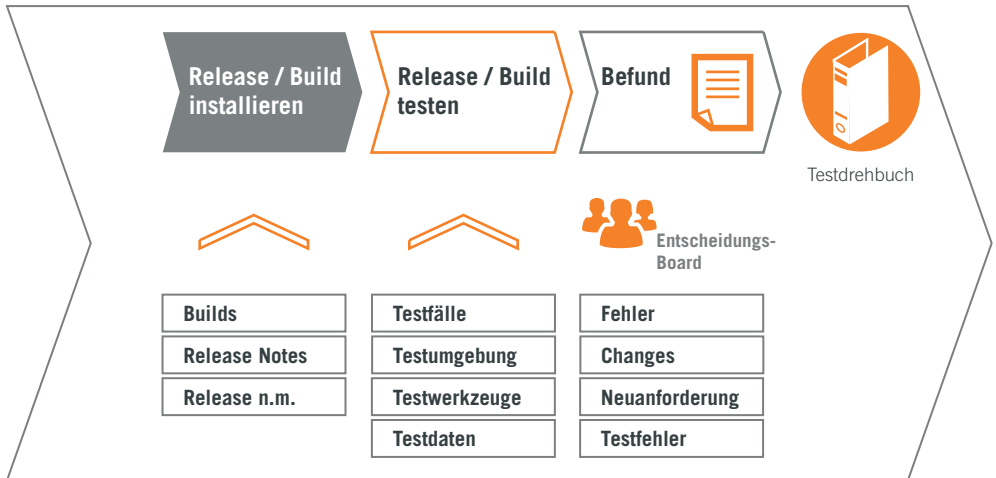


Die Planung kann auch an einem Board erfolgen. Wichtig ist, dass auch Test- und Review Tasks eingepflegt werden.

Story	Backlog		In Process	To Verify	Done
As a user, I... 8 points	Code the... 9	Test the... 8	Code the... DC 8	Test the... SC 6	Code the... Test the... DC 6
	Code the... 2	Test the... 8	Test the... SC 8		Test the... Code the... Test the... DC 6
	Code the... 8	Code the... DC 8			
As a user, I... 5 points	Code the... 8	Test the... 8	Code the... DC 8		Test the... Test the... Code the... DC 6
	Code the... 4	Test the... 6			

☐ Entwickler Task
 ☐ Tester Task
 ☐ Review Task

# Durchführen



«Die Praxis zeigt immer wieder, dass «von Hand gepflegte» Tabellen und Listen in kritischen Momenten nicht mehr gepflegt werden, denn operative Testarbeiten werden den administrativen Arbeiten immer vorgezogen. Doch genau in hektischen Phasen sind Kennzahlen als Entscheidungsgrundlagen besonders wichtig!»

Während der eigentlichen Testphase fallen sehr viele Informationen an. Geeignete Workflows und Prozesse stellen sicher, dass die Informationen jederzeit aktuell und verfügbar sind.

In der oft entstehenden Hektik ist es wichtig, dass die Abläufe, die gewünschten Kennzahlen sowie die Informationsempfänger vorgängig im Testkonzept definiert wurden.

## Fortschritt

- Wie viele Tests/Testsszenarien stehen zur Verfügung?
- Wie viele Tests/Testsszenarien konnten im Verhältnis zu allen Test/Testsszenarien getestet werden?
- Wieviel konnte erfolgreich getestet werden?
- Wie viele Testfälle sind noch zu testen?
- Wie lange dauern die Tests noch?

*Wichtig:  
Die jeweiligen Kennzahlen müssen immer beurteilt und in Bezug zur Gesamtheit oder in einem Verlauf oder Verhältnis beurteilt werden.*

## Reifegrad

- Summe aller Fehler (pro Fehlerklasse)
- Summe der offenen Fehler (pro Fehlerklasse)
- Anzahl der erfolgreich getesteten Anforderungen

## Fehlerklassifizierung

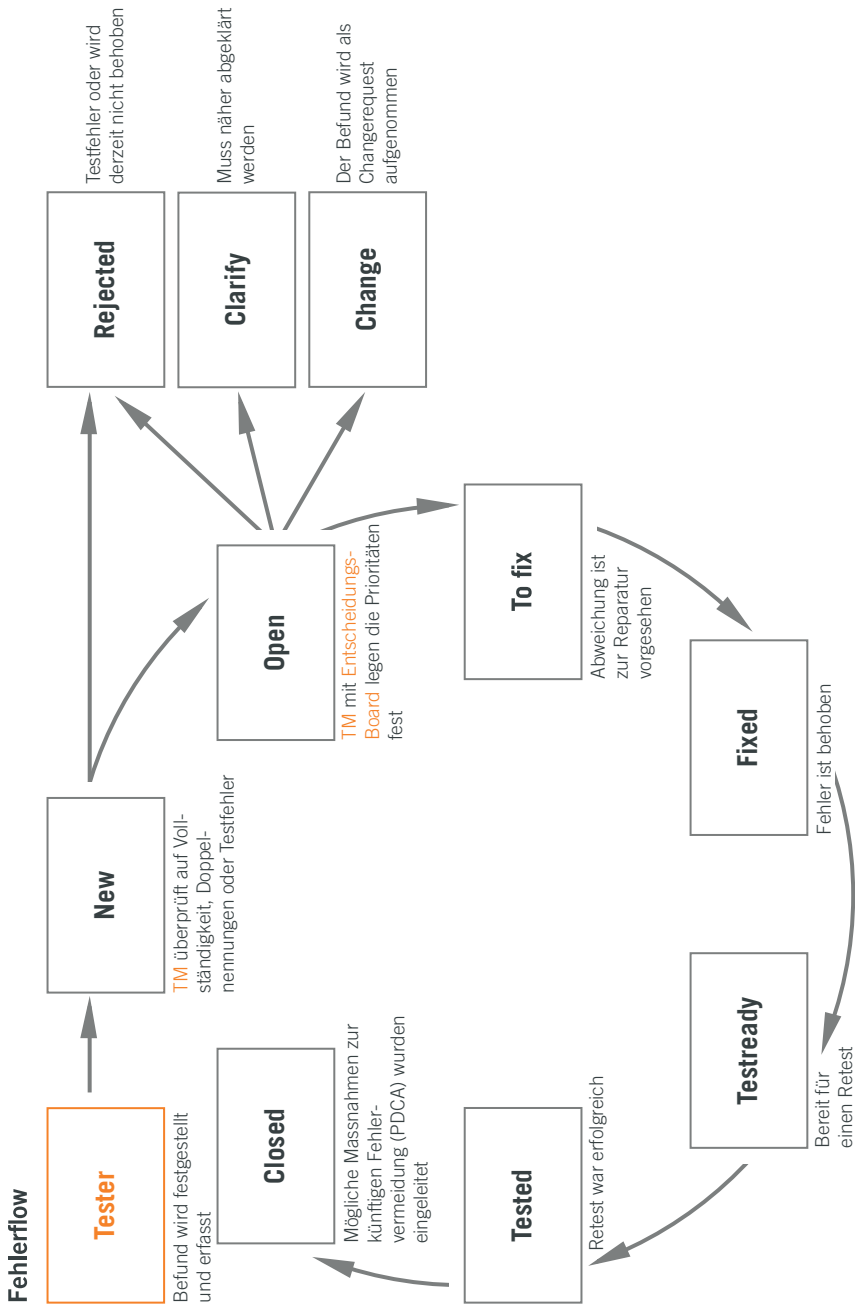
### Fehlerauswirkung, Schweregrad (Severity)

<b>1 Low</b>	Leichter Fehler, betrifft einzelnen Testschritt, Funktion bleibt im Wesentlichen gewährleistet
<b>2 Medium</b>	Betriebsstörender Fehler, Systemfunktion nicht beeinträchtigt. Wesentliches funktioniert, wenn auch eingeschränkt
<b>3 High</b>	Schwerer Fehler, Auswirkungen auf Funktion, keine Auswirkung auf andere Funktionen/Systeme
<b>4 Urgent</b>	Fataler Fehler, Auswirkung auf ganzes System, Testabbruch

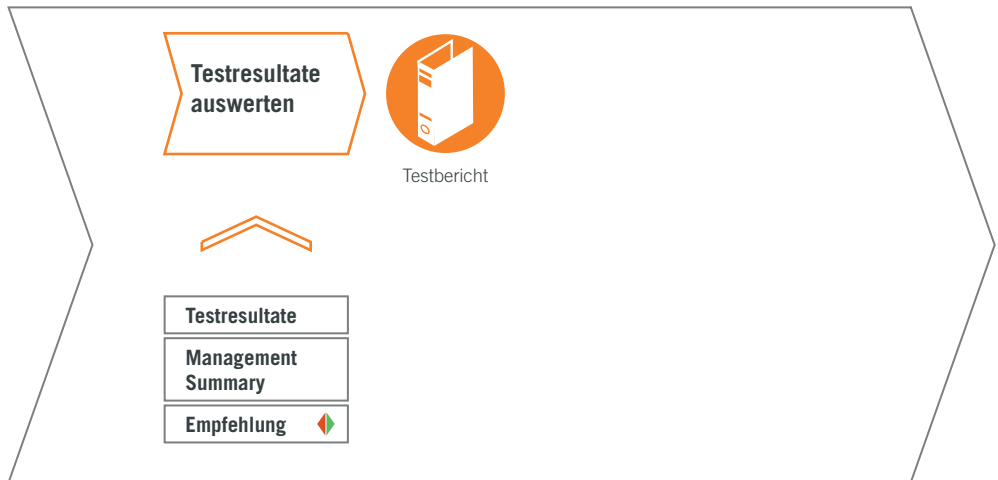
### Beobachtungsgüte/Reproduzierbarkeit

«A»	Eindeutig festgestellter, belegbarer und reproduzierbarer Fehler
«B»	Nicht ohne weiteres reproduzierbar, aber wiederholt aufgetretener Fehler
«C»	Nicht reproduzierbar

# Prozess: Incident-/Defectmanagement



## Auswerten



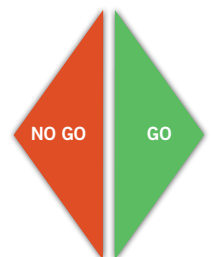
In der Auswertungsphase werden die vorhandenen Testresultate in «management-tauglichen» Reports zusammengefasst und grafisch dargestellt.

Der Testreport soll dem Management ermöglichen, basierend auf dem zu erwartenden Restrisiko eine Entscheidung betreffend dem Go/NoGo zu treffen.

## Wagen wir es?

RPI	Abdeckung	offene Fehler
1	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	0
2	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	0
3	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	Maximal 1 low
4	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	Maximal 3 low plus 1 medium
5	<div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div><div></div></div>	Maximal 10 low plus 3 medium

■ Getestet  
■ Ungetestet





## Argumente fürs Testen

Oft fehlen gute und überzeugende Argumente um die entscheidenden Personen, für Investitionen ins Testen, zu überzeugen. Daher hier eine Auswahl an Argumentationshilfen.

### Stichwort

Katastrophe

### Nutzenargument

Abwehren von unerwarteter Katastrophe

### Beispiel

Personenschaden, Betriebsausfall, Reputation, unermessliche Kostenfolgen

---

Regulatorien

Keine Haftung, da Haftung nachweisbar erfüllt

### Beispiel

Risikoanalyse, dokumentierte Testabläufe, Sorgfaltspflicht und Resultate FEMA, MTBF

---

Garantiekosten

Entlastung nach Übergabe

### Beispiel

Abnahmetests, Abnahmedokument

---

Interdisziplinarität

Gut qualifiziertes (teures) Personal mehrfach nutzen

### Beispiel

Bessere Requirements, schnellere Entwicklung, früh eine aktuelle Dokumentation

---

Klare Verantwortlichkeit

Entlastung durch Delegieren

### Beispiel

Aktive Testmanagerinnen und Testmanager gestalten, führen und übernehmen so auch Verantwortung für das Gelingen des Projekts.

---

Qualität, Review

Projekterfolg

### Beispiel

4-Augen-Prinzip  
Zweitmeinung, nicht nur vom «Entwicklungsleiter»

---

## Stichwort

KPI

### Beispiel

Abdeckung, Durchlaufzeit, Anzahl Fehler, Fehlerklassen, Restrisiko

(Test-)Strategie

## Nutzenargument

Kennzahlen zum Führen

Gezielt und geplant

Agieren statt Reagieren

Nur notwendiges Testen

### Beispiel

Personenschaden, Betriebsausfall, Reputation, unermessliche Kostenfolgen

---

Automatisierung

Einsparungen (Schlagwort)

### Beispiel

Der Testmanager kann aufzeigen, wann und wo es sich rechnet.

---

Vertrauen ist gut, Kontrolle ist besser

Klares Einhalten von Regeln und Vorgaben

### Beispiel

Versionierung, Codereviews, Dokumentation, Statistiken, Kennzahlen

---

Geteilte Verantwortung

Bessere und objektivere Beurteilung, weniger Abhängigkeit von Einzelpersonen

### Beispiel

Testmanager leitet Prioboard. Hier werden Befunde objektiv beurteilt und priorisiert. Der Projektleiter wird nur als «letzte Instanz» benötigt und kann dann ohne Vorgeschichte entscheiden.

---

Es gibt viele gute Gründe, um in ein professionelles Testen zu investieren. Je nach Strategie und Entscheidungsträger braucht es zum Überzeugen angepasste Argumente.

---

Wann wird getestet, Sie können wählen:

- Entweder testen Sie, geplant und systematisch, dann ist es budgetierbar.
- Oder Sie überlassen das Testen den Kunden, mit nicht budgetierbaren Folgekosten.

Egal, wie Sie entscheiden: Getestet wird Ihr Produkt immer!

---

# Testautomatisierung

Die Testautomatisierung ist ein sehr breites Feld und sollte immer Teil des Testkonzepts sein. Dabei sollten im Minimum folgende Fragen geklärt werden:

## Aspekte des Testmanagements und des Testkonzepts

- Was ist das Ziel der Automatisierung?
- Wie kann eine definierte Ausgangslage für reproduzierbare Tests sichergestellt werden?
- Auf welchen Teststufen soll automatisiert werden?
- Welcher Aspekt spielt die Reduktion von Manipulationsfehlern?
- Wie wichtig ist die Zeitersparnis und die Testdurchlaufzeit?
- Muss das Testen parallelisiert werden?
- Müssen Vollzeitstellen (FTE) reduziert werden?
- Wie werden (grosse Mengen von) Testresultate verwaltet?
- Wie komplex sind einzelne Handlungsabläufe?
- Wie wird die Pflege und Wartung der Tools, der Tests sowie der Testdaten organisiert?
- Wie wird die Wiederverwendbarkeit der automatisierten Tests sichergestellt?
- Welches Spezialwissen (Skills) ist für die Automatisierung sowie für die Testdurchführung notwendig?
- Wie ist die Lebensdauer der Testautomatisierung/Amortisation?
- Lohnt sich die Automatisierung? (ROI)

## Aspekte der Technik

- Eigenes Tool, eigenes Framework
- Passt das Tool zur eigenen Technologie
- Homogenität der Toolandschaft
- Design der Testarchitektur
- Design des Testframeworks, Modularität
- Versionsverwaltung
- Continuous Integration/Nightly Build
- Automatische Schnittstellen zum Testmanagement
- Toolwahl (Microsoft TFS, HP ALM, NI, RANOREX, SELENIUM, TOSCA, ...)



*In agilen Projekten sind die Release-Zyklen sehr kurz. Gerade deshalb bildet die Testautomatisierung einen entscheidenden Erfolgsfaktor. Manuelle Tests können in zu grosser Anzahl nicht seriös bearbeitet werden!*

# Das Testkonzept – der Grundstein eines erfolgreichen Testprojekts

Wie erwähnt, bildet das Testkonzept das entscheidende Fundament für ein erfolgreiches Testprojekt. Diese Struktur entspricht der in der ISO-Norm 29119 definierten Struktur.

## Einleitung

Einleitung

Ziel und Zweck

Geltungsbereich

Geltende Dokumente/Referenzen

## Testobjekt

Definition, welche Softwarekomponenten/Umsysteme getestet werden sollen (Scope) und was explizit nicht Bestandteil des Testprojekts ist (Constraint).



Komponentendiagramm als Basis, um die Testobjekte zu definieren. Softwarekomponenten, die Bestandteil des Testobjekts sind, können z.B. entsprechend eingefärbt werden.

## Testumfang

Definition, welche Qualitätsmerkmale und Qualitätskriterien getestet werden sollen.

Ein Qualitätsmerkmal hat als Ausprägung mehrere Qualitätskriterien.



### Qualitätsmerkmale (Beispiele)

- Funktionalität
- Zuverlässigkeit
- Effizienz
- Benutzbarkeit
- Wartbarkeit
- Übertragbarkeit
- Sicherheit

### Qualitätskriterien (Beispiele)

- Richtigkeit
- Angemessenheit
- Interoperabilität
- Stabilität
- Performance
- Verständlichkeit

## Kommunikationswege

Beschreibt die Kommunikationswege der Mitglieder des Testprojekts je nach Rolle.



### Mögliche Rollen

- Projektleiter
- Product Owner
- Testmanager
- Testengineer
- Tester

## Risikobetrachtung

Einstufung der Risiken nach Auffindbarkeit, Schweregrad und Umsetzungskomplexität. Falls diese schon im Rahmen eines Risikomanagements geführt werden, kann hier auf das entsprechende Dokument verwiesen werden. Zu den Risiken sollen auch immer adäquate Massnahmen beschrieben werden.



Die Risikobeurteilung führt zu einer Priorisierung der Tests und liefert Hinweise auf die nötige Testtiefe und Breite sowie die Anzahl der nötigen Regressionstests.

## Teststufen

Definition, ob folgende Teststufen durchgeführt werden:

- Komponententests
- Integrationstests
- Systemtests
- Abnahmetests

An dieser Stelle können auch abhängige Lieferobjekte aufgelistet werden. Für jede Teststufe wird die Verantwortung und das Testvorgehen definiert.



Je nach Projekt kann auf einzelne Teststufen verzichtet werden. So kann es vorkommen, dass es keine Unterscheidung zwischen Integrations- und Systemtests gibt.

## Lieferobjekte

Welche Lieferobjekte entstehen aus dem Testprojekt?

Wer ist verantwortlich für das Erstellen der notwendigen Artefakte, wer ist der Empfänger der Artefakte?

Hier sind Lieferobjekte zu nennen, die sich nicht auf eine bestimmte Teststufe beziehen.



### Beispiele

- Testkonzept
- Dokumentation der Testumgebung
- Testdaten
- Status-Reports (pro Iteration)
- Final-Report (nach Akzeptanztest)
- Testscripts
- Automatisierungscode

## Komponententests/Unit-Tests

Werden Unit-Tests durchgeführt? Wer führt diese durch? Wo werden diese definiert/beschrieben?



Die Entwicklung erstellt adäquate Unit-Tests direkt in der Entwicklungsumgebung.

## Planung

Falls nicht bereits genügend detailliert in der Projektplanung ersichtlich, wird hier die Terminplanung mit Testaktivitäten auf der Zeitachse ausgewiesen.



GANT-Chart der einzelnen Testiterationen

## Abnahmetest

Folgende Punkte müssen hier beschrieben werden:

- Wer die Abnahmetests vorgängig frei gibt (genehmigt)
- Wer die Abnahmetests durchführt
- An wen die Testresultate rapportiert werden (Abnahmeprotokoll)

## Testabbruch

Beschreibt Kriterien, unter welchen die Tests teilweise oder ganz abgebrochen werden, resp. wiederaufgenommen werden. Es muss identifiziert werden, wer dies veranlasst. Hier können auch Tätigkeiten für die Wiederaufnahme definiert werden.



Werden schwerwiegende Fehler während eines Testverlaufs gefunden, welche weitere Tests verunmöglichen, so entscheidet der Testmanager und der Projektleiter über den Testabbruch.

## Testaktivitäten

Beschreibt, welche Aktivitäten vor, während und nach dem Testprojekt durchgeführt werden. Auch Aktivitäten, die explizit nicht durchgeführt werden, sollten vollständigshalber beschrieben werden.



Analyse/Vorbereitung

Studium der Testbasis, Entwurf der Testfälle, Aufbau Testinfrastruktur

Durchführung/Auswertung

Funktionale Tests, Bugtracking, Zwischenbericht, nicht Funktionale Tests in Bezug auf Effizienz, Stabilität, etc.

Abschluss

Verfassen Abschlussbericht, Datenarchivierung, Abschlusspräsentation

## Personal

Beschreibt, wer welche Rolle innerhalb des Testprojekts hat, ob zusätzliches Personal rekrutiert werden muss oder ob Ausbildungsbedarf beim aktuell definierten Projektteam besteht.



Beispiele für Rollen

- Testmanager
- Testengineer
- Testautomatisierer
- Tester
- Testdatenverantwortlicher
- BUG-Team

Folgende Kapitel müssen pro Teststufe (z.B. System- und Integrationstest) separat definiert werden. Sprich, diese Kapitel wiederholen sich pro eingesetzte Teststufen.

## Testdesigntechniken

Beschreibung, wie die Testfälle hergeleitet werden.



Review:  
Informell, Walkthrough, Inspektion

Statische Analyse:  
Datenflussanalyse, Kontrollflussanalyse

Dynamische Tests (Blackbox):  
Äquivalenzklassen, Grenzwertanalyse, Entscheidungstabelle,  
Anwendungsfallbasiert

## Testendkriterien

Beschreibt, nach welchen Kriterien das Testende der jeweiligen Teststufe erreicht ist und die Tests als abgeschlossen betrachtet werden. Hierbei sollen qualitative Aspekte und nicht monetäre oder zeitliche Aspekte genannt werden.



Alle Testfälle mit Priorität «Critical» verlaufen erfolgreich und maximal 10% der nicht kritischen Testfälle schlagen fehl.

## Testmetriken

Beschreibt, welche Metriken auf den Ebenen: Anforderungen, Testfälle, Testsets, Incidents/Defects während des Tests auf der jeweiligen Teststufe erhoben werden. Dabei kann auf vorhandene Metriken des entsprechenden Testmanagementtools zurückgegriffen werden.



Beispiel (Anforderungen)

- Anzahl Anforderungen, bei denen die Testfälle erfolgreich durchgeführt werden konnten, resp. ein oder mehrere Testfälle fehlschlagen.

Beispiele (Testfälle)

- Anzahl Testfälle mit den Status «Passed» und «Failed»



## Anforderungen an Testdaten

Spezifiziert, welche Testdaten erstellt werden und wo diese abgelegt/verwaltet werden.



### Beispiele

- Stammdaten des Kunden
- Daten von Vorgängerversionen
- Von Testing erstellte Skripte

## Anforderungen an die Testumgebung

Beschreibt die Testumgebung (Software und Hardware), sowie die abhängigen Umsysteme, die Einfluss auf die Testumgebung haben. Dabei soll auch die Version der entsprechenden Systeme angegeben werden.



Der Versionsstand darf während einer Testiteration nicht verändert werden. Dies betrifft ebenfalls Änderungen, die die Konfiguration betreffen und nicht direkt im Programmcode definiert sind.

## Regressionstests und Fehlernachtests

Beschreibt, wie Testfälle identifiziert werden, die in der nächsten Iteration wiederum getestet werden. Ferner wird definiert, wie mit Incidents/Defects umgegangen wird, resp. wie der Prozess hierfür definiert ist.

# Übersicht «Dienstleistungen im Testengineering»

## Assessment

Sie möchten wissen, wo Sie stehen?

**Wir zeigen Ihnen ihr Potential.**

Auf Basis einer umfassenden Checkliste erheben wir entlang des «Noser Way of Testing» den aktuellen Reifegrad (Maturität) Ihrer Prozesse. Das Ergebnis wird in einer übersichtlichen Grafik sowie einem kurzen Bericht festgehalten. Auf Wunsch bieten wir zusätzlich ein Assessment an.

## Teststrategie

Sie haben viel vor, wissen aber noch nicht wie anpacken?

**Wir zeigen wie und packen es an.**

Auf Basis des Assessments sowie unserer langjährigen Erfahrung definieren wir die auf Ihre Bedürfnisse passende Teststrategie und leiten daraus konkrete Ziele ab. Zielführende Massnahmen werden risikogerecht priorisiert und in konkrete Arbeitspakete aufgeteilt. Eine Roadmap zeigt überprüfbar auf, wann Sie welche Ziele erreichen.

## Coaching

Sie haben Respekt vor der Herausforderung?

**Wir coachen Ihre Schlüsselpersonen.**

Wir unterstützen und begleiten Sie auf dem Weg und lassen Sie von unserer Erfahrung profitieren. Mit regelmässigen Meetings begleiten wir Ihre Schlüsselpersonen. Agil und massgeschneidert definieren wir Schritt für Schritt die Ziele und unterstützen Sie bei der Umsetzung. Je nach Bedürfnis, Methode, Technologie oder Phase ergänzen wir das Coaching durch passende Experten oder Expertinnen.

## SCRUM und Testmanagement

Sie möchten SCRUM einsetzen?

**Testmanagement macht SCRUM erfolgreich.**

Unsere Erfahrung zeigt, dass nach einer anfänglichen Euphorie SCRUM allein die Softwarequalität nicht nachhaltig verbessert. Konsequentes, auf SCRUM angepasstes Testmanagement kann jedoch Product Owner und SCRUM-Master entlasten und die Softwarequalität nachhaltig und messbar verbessern.

## Schulung

Sie haben Bedarf an Ausbildung?

**Wir teilen unser Wissen und unsere Erfahrung mit Ihnen.**

«The Noser Way of Testing» bedeutet über 30 Jahre Erfahrung. Mit angepassten Schulungen und Kursen teilen wir unser Wissen und unsere Erfahrungen mit Ihnen. Gezielt schulen wir Einzelpersonen oder ganze Teams. Nach bestandener Lernkontrolle erteilen wir das beliebte «Approved by Noser»-Kurszertifikat. Auf Wunsch organisieren wir auch Inhouse-ISTQB-Kurse.

## Change Management

Steht eine Veränderung an?

**Wir begleiten Sie mit Change-Management Spezialisten.**

Das Einführen eines effektiven Testmanagements hat Einfluss auf bestehende Organisationen und Abläufe. Nebst den technischen Aspekten spielen so auch organisatorische Aspekte, sogenannte «weiche Faktoren» eine entscheidende Rolle. Wird dieser Change Aspekt zu wenig beachtet, so verlaufen die Anstrengungen oft im Sand oder erzeugen unnötige Reibungen. Unsere erfahrenen Projektleiter und Testmanager unterstützen und begleiten Sie durch das «Tal der Veränderung».

## Impuls

Sie möchten das Thema Qualität und Testen einbringen?

**Wir zünden den Funken.**

Ein auf Ihre Situation angepasstes Impulsreferat sensibilisiert und begeistert Ihre Entscheidungsträger und Schlüsselpersonen. Wir zünden den Funken und bringen aktuelle, brennende Themen «Rund ums Testen» auf den Punkt. Profitieren Sie von einem kurzen Impulsreferat: Planen Sie für das nächste Meeting etwas zusätzliche Zeit ein. Nach dem Referat folgt oft eine spannende Frage- und Antwort-Runde. Oft bildet dieser Impuls den zündenden Funken für ein erfolgreiches Testen.

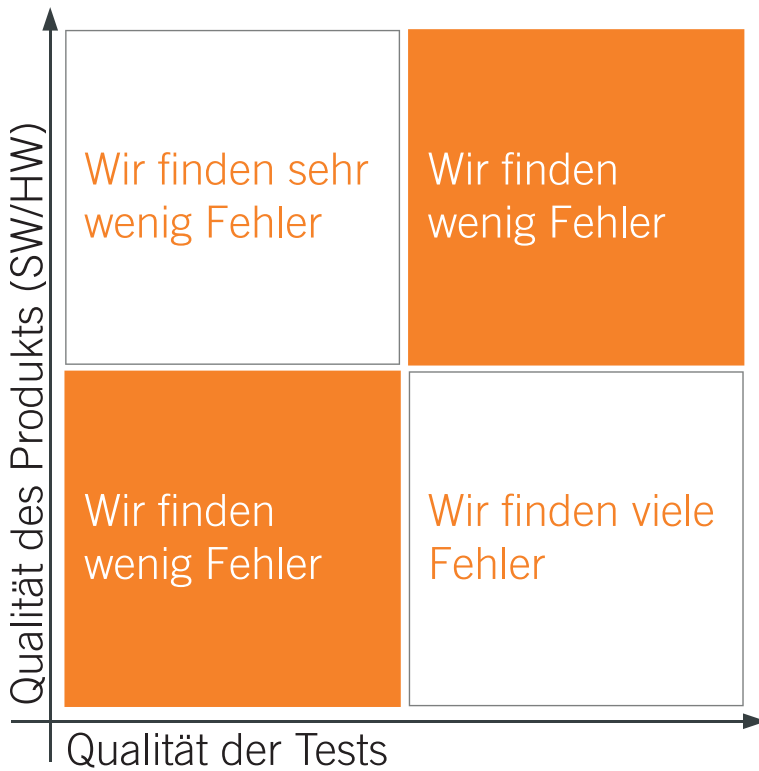
## Ihr Führungscockpit

Führen Sie im Blindflug, vermissen Sie Kennzahlen?

**Wir etablieren das bewährte Cockpit.**

Das Führen von Projekten bedingt verlässliche und objektive Kennzahlen. Unterstützt von geeigneten Tools (z.B. Microsoft TFS, HP-ALM, JIRA etc.) etablieren wir bei Ihnen das bewährte Führungscockpit. So haben Sie die relevanten Kennzahlen jederzeit aktuell und online verfügbar.

Warum ist die Testqualität so wichtig:



Gleicher Befund bei total anderer Ausgangslage

Bestellen Sie sich Ihr Testing Poster kostenlos und profitieren Sie von unserem Know-how. E-Mail mit Betreff «Testing Poster» an [testing@noser.com](mailto:testing@noser.com).



## Noser Engineering AG in Ihrer Nähe

### **Winterthur**

Rudolf-Diesel-Strasse 3  
CH-8404 Winterthur  
Tel. +41 52 234 56 11  
[info@noser.com](mailto:info@noser.com)  
[www.noser.com](http://www.noser.com)

### **Luzern**

Platz 4  
CH-6039 Root D4  
Tel. +41 41 455 66 11  
[luzern@noser.com](mailto:luzern@noser.com)  
[www.noser.com](http://www.noser.com)

### **Bern**

Galgenfeldweg 18  
CH-3006 Bern  
Tel. +41 31 917 45 11  
[bern@noser.com](mailto:bern@noser.com)  
[www.noser.com](http://www.noser.com)

### **München**

Konrad-Zuse-Platz 1  
DE-81829 München  
Tel. +49 89 990 14 88-0  
[muenchen@noser.com](mailto:muenchen@noser.com)  
[www.noser.com](http://www.noser.com)

### **Heppenheim**

Donnersbergstrasse 1  
DE-64646 Heppenheim  
Tel. +49 62 526 74 45-0  
[heppenheim@noser.com](mailto:heppenheim@noser.com)  
[www.noser.com](http://www.noser.com)





