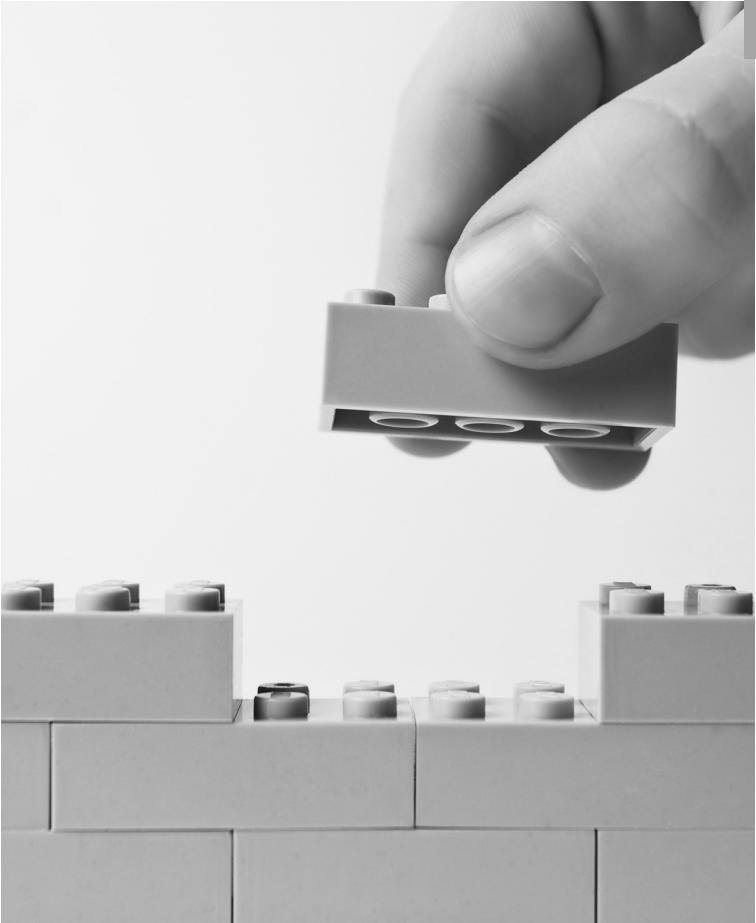


1 – Grundlagen des Testens

1. Was ist Testen?
2. Warum ist Testen notwendig?
3. Testaktivitäten, Testmittel und Rollen des Testens
4. Wesentliche Kompetenzen und bewährte Praktiken beim Testen



1.1 Was ist Testen?

- **FL-1.1.1 (K1)** Sie können typische Testziele identifizieren
- **FL-1.1.2 (K2)** Sie können Testen von Debugging unterscheiden



Was ist Testen?

Was ist Testen?



Softwaresysteme sind ein integraler Bestandteil unseres täglichen Lebens.

Software, die nicht ordnungsgemäß funktioniert, kann zu vielen Problemen führen.

Z. B. Geld-, Zeit- oder Ansehensverlust und in Extremfällen sogar zu Verletzung oder Tod.

Softwaretests **bewerten die Qualität der Software** und helfen das **Risiko einer Fehlerwirkung** im Betrieb **zu verringern**.

Was ist Qualität?

Der Grad, zu dem eine Komponente oder ein System die expliziten und impliziten Bedürfnisse seiner verschiedenen Stakeholder erfüllt.

Umfrage



- Kennen Sie Beispiele für Softwarefehler?
- Was waren die Folgen?

Missverständnisse über das Testen I



Testaktivitäten

Testaktivitäten

Das Testen von Software ist eine Reihe von Aktivitäten zur Entdeckung von Fehlerzuständen und zur Bewertung der Qualität von Arbeitsergebnissen der Softwareentwicklung. Werden diese Arbeitsergebnisse getestet, werden sie als Testobjekte bezeichnet.

Ein weit verbreitetes Missverständnis:

„Testen besteht nur aus dem Ausführen von Tests“
(d. h. dem Ausführen der Software und der Prüfung der Testergebnisse).

Das Testen von Software umfasst jedoch auch **andere Aktivitäten** und **muss auf den Softwareentwicklungslebenszyklus abgestimmt sein**.

Missverständnisse über das Testen II



Verifizierung und Validierung

Verifizierung und Validierung

Ein weiteres Missverständnis:

„Testen konzentriert sich ausschließlich auf das Prüfen, ob das System die spezifizierten Anforderungen erfüllt.“

Testen beinhaltet ebenfalls das **Prüfen, ob das System die Bedürfnisse** der Benutzer und anderer Stakeholder in seiner Betriebsumgebung **erfüllt**.



Schlüsselbegriffe – Testziele

Verifizierung

Bestätigung durch Bereitstellung eines objektiven Nachweises, dass festgelegte Anforderungen erfüllt worden sind.

Validierung

Bestätigung durch Überprüfung, dass ein Arbeitsergebnis den Bedürfnissen eines Stakeholders entspricht.

Merkfrage

"Wurde das Produkt richtig realisiert?"

Merkfrage

"Wurde das richtige Produkt realisiert?"

Testen – dynamisch oder statisch?



Dynamischer Test

Beim dynamischen Test wird die Software ausgeführt.

Beim dynamischen Test werden verschiedene Arten von Testverfahren und Testansätzen verwendet, um Testfälle abzuleiten.



Statischer Test

Beim statischen Test wird die Software nicht ausgeführt.

Zum statischen Test gehören Reviews und statische Analysen.

Management der Testaktivitäten und Testwerkzeuge

Testen muss geplant, verwaltet, geschätzt, überwacht und gesteuert werden.

Tester verwenden Werkzeuge.

Testen ist nicht nur eine technische Aktivität, es ist eine weitgehend intellektuelle Aktivität.

Das erfordert von Testern:

- Fachwissen
- die Anwendung analytischer Fähigkeiten
- den Einsatz kritischen Denkens
- Systemdenken

Die Norm ISO/IEC/IEEE 29119-1 liefert weitere Informationen zu Konzepten für das Testen von Software.



Testziele



Typische Testziele I

- Evaluieren von Arbeitsergebnissen, z. B. Anforderungen, User-Storys, Entwürfe und Code
- Auslösen von Fehlerwirkungen und Finden von Fehlerzuständen
- Sicherstellen der erforderlichen Überdeckung eines Testobjekts
- Verringern des Risikos einer unzureichenden Softwarequalität
- Verifizieren, ob spezifizierte Anforderungen erfüllt wurden
- Verifizieren, ob ein Testobjekt den vertraglichen, rechtlichen und regulatorischen Anforderungen entspricht

Typische Testziele II

- Informationen für die Stakeholder bereitstellen, um fundierte Entscheidungen treffen zu können
- Vertrauen in die Qualität des Testobjekts aufzubauen
- Validieren, ob das Testobjekt vollständig ist und wie erwartet funktioniert (aus Sicht der Stakeholder)

Testziele können je nach Kontext variieren. Dazu gehören ...

- das zu testende Arbeitsergebnis
- die Teststufe
- Risiken
- der Softwareentwicklungslebenszyklus
- Faktoren im Zusammenhang mit dem geschäftlichen Kontext
 - z. B. Unternehmensstruktur, Wettbewerbserwägungen oder die Zeit bis zur Markteinführung

Unterschiedliche Testziele je nach Teststufe



Testziele im Komponententest

- Fehlerwirkungen provozieren und Fehlerzustände finden
- Die erforderliche Überdeckung des Testobjekts sicherstellen
- Verifizieren, ob die spezifizierten Anforderungen erfüllt wurden

Testziele im Abnahmetest

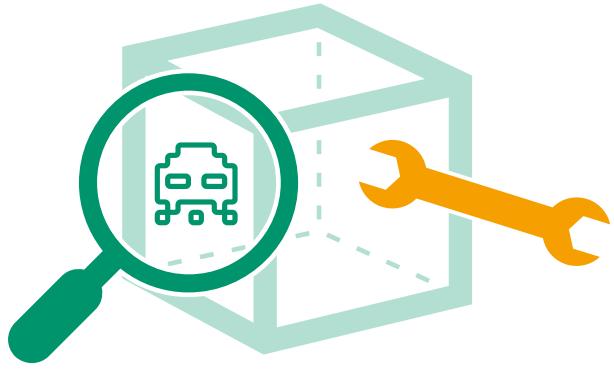
- Vertrauen in die Qualität des Testobjekts aufbauen
- Validieren, ob sich das Testobjekt aus Sicht der Stakeholder wie erwartet verhält

Testen und Debugging

Testen und Debugging sind getrennte Aktivitäten

Testen

- löst Fehlerwirkungen aus, die durch Fehlerzustände in der Software verursacht werden (**dynamischer Test**)
- findet direkt Fehlerzustände im Testobjekt (**statischer Test**)



Debugging

- findet, analysiert und beseitigt die Fehlerzustände

Debugging Prozess

1. Reproduzieren einer Fehlerwirkung
2. Diagnose (Finden des Fehlerzustands)
3. Fehlerbehebung

Je nachdem, mit welchem **Test** (dynamisch oder statisch) Fehler gefunden werden,
ändert sich der **Debugging Prozess**.

Debugging nach dynamischem Test

Dynamischer Test

- löst **Fehlerwirkungen** aus, die durch Fehlerzustände in der Software verursacht werden

Nach dem Debugging:

- **Fehlernachtests** prüfen, ob das Debugging erfolgreich war.
- **Regressionstests** prüfen, ob die Korrekturen in anderen Teilen des Testobjekts Fehlerwirkungen verursachen.

Debugging

- findet, analysiert und beseitigt die **Fehlerzustände**



Debugging Prozess

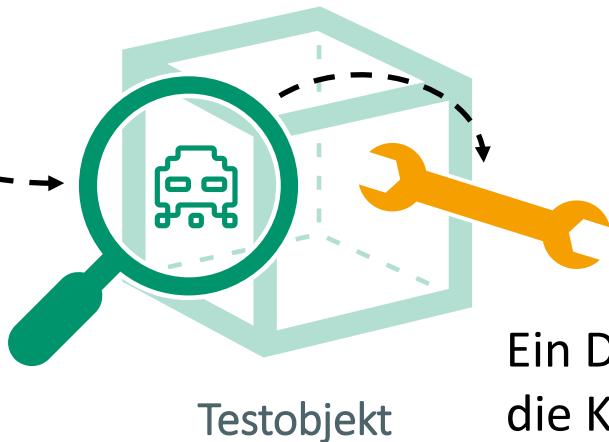
1. Reproduzieren einer Fehlerwirkung
2. Diagnose (Finden des Fehlerzustands)
3. Fehlerbehebung

Idealerweise wird der **Test** und der **Fehlernachtest** von **derselben Person** durchgeführt.

Debugging nach statischem Test

Statischer Test

- findet Fehlerzustände direkt im Testobjekt



Debugging

- beseitigt den Fehlerzustand direkt

Debugging Prozess

1. Reproduzieren einer Fehlerwirkung
2. Diagnose (Finden des Fehlerzustands)
3. Fehlerbehebung

Ein Debugging Prozess ist oft notwendig, damit die Korrektur keine weiteren Probleme aufwirft.

Wird beim statischen Test ein **Fehlerzustand festgestellt**, sind beim Debugging **Reproduktion oder Diagnose nicht erforderlich**.



Testen – ein mögliches Vorgehen

Ah, da ist ja die neue Softwareversion.

Wurde auch Zeit! Mal probieren ...

Install,
Datei laden ...

Scheint zu funktionieren!



Moment, #DivNull!?

Da muss ich gleich mal anrufen!



Ich habe keine Ahnung, was ich eingegeben habe, aber es ist zu einem Fehler gekommen.



Ist besser als nichts zu tun, aber ineffizient, weil unsystematisch, unvollständig und nicht dokumentiert.

Debugging – ein mögliches Vorgehen



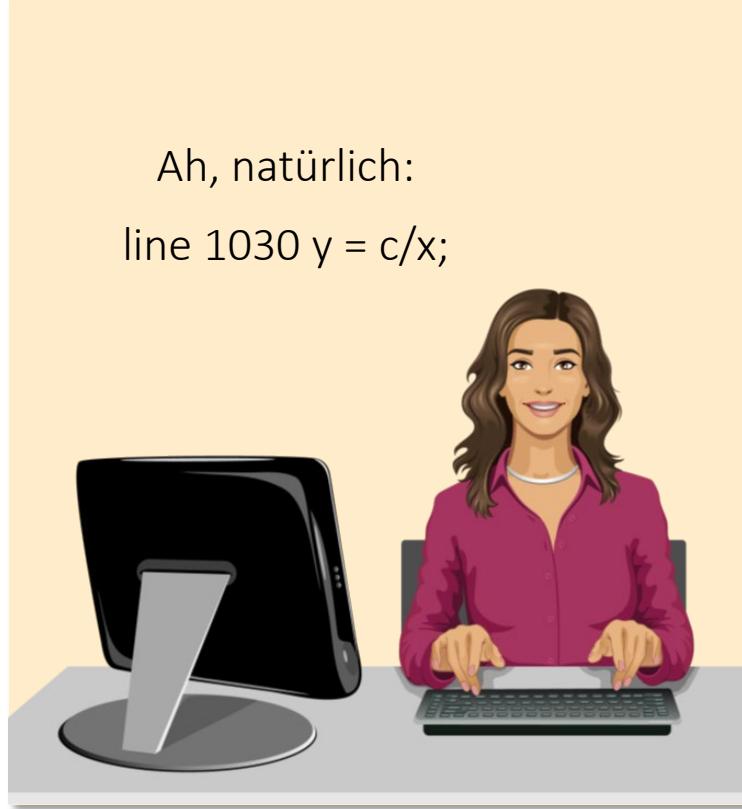
#DivNull ?!?

Hm, wo wird denn
im Code dividiert?



Ah, natürlich:

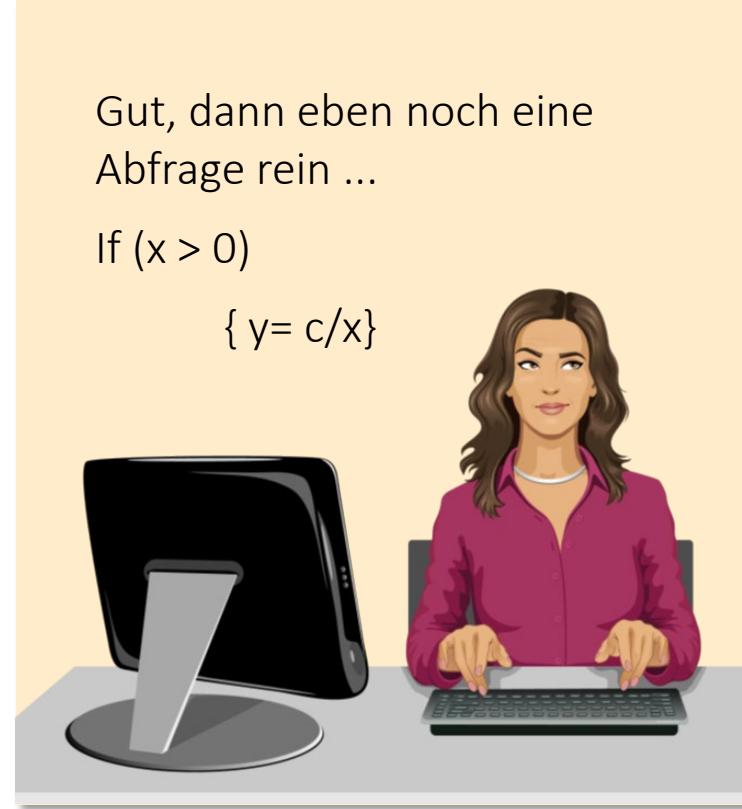
line 1030 y = c/x;



Gut, dann eben noch eine
Abfrage rein ...

If ($x > 0$)

{ $y = c/x$ }



Debugging zur Lokalisierung und Behebung von Fehlerzuständen,
während der Softwareentwicklung, oder auch während der Fehlerkorrektur.



Schlüsselbegriffe – Grundlagen

Debugging

Der Prozess der Aufdeckung, Analyse und Entfernung der Ursachen von Fehlerwirkungen in einer Komponente oder einem System.

Testen

Der Prozess innerhalb des Softwareentwicklungslebenszyklus, der die Qualität einer Komponente oder eines Systems und der zugehörigen Arbeitsergebnisse bewertet.

Testobjekt

Das zu testende Arbeitsergebnis.

Validierung

Bestätigung durch Überprüfung, dass ein Arbeitsergebnis den Bedürfnissen eines Stakeholders entspricht.

Testziel

Der Zweck des Testens.

Verifizierung

Bestätigung durch Bereitstellung eines objektiven Nachweises, dass festgelegte Anforderungen erfüllt worden sind.



Typische Testziele

Typische Ziele des Testens sind

- Evaluieren von Arbeitsergebnissen
(Anforderungen, User-Storys, Entwürfe und Code)
- Auslösen von Fehlerwirkungen und Finden von Fehlerzuständen
- Sicherstellen der erforderlichen Überdeckung eines Testobjekts
- Verringern des Risikos einer unzureichenden Softwarequalität
- Validieren, ob das Testobjekt vollständig ist und sich aus Sicht der Stakeholder wie erwartet verhält.
- Verifizieren
 - ob spezifizierte Anforderungen erfüllt wurden
 - ob ein Testobjekt den vertraglichen, rechtlichen und regulatorischen Anforderungen entspricht
- Bereitstellen von Informationen für die Stakeholder, damit diese fundierte Entscheidungen treffen können
- Aufbauen von Vertrauen in die Qualität des Testobjekts



Testen und Debugging

Testen

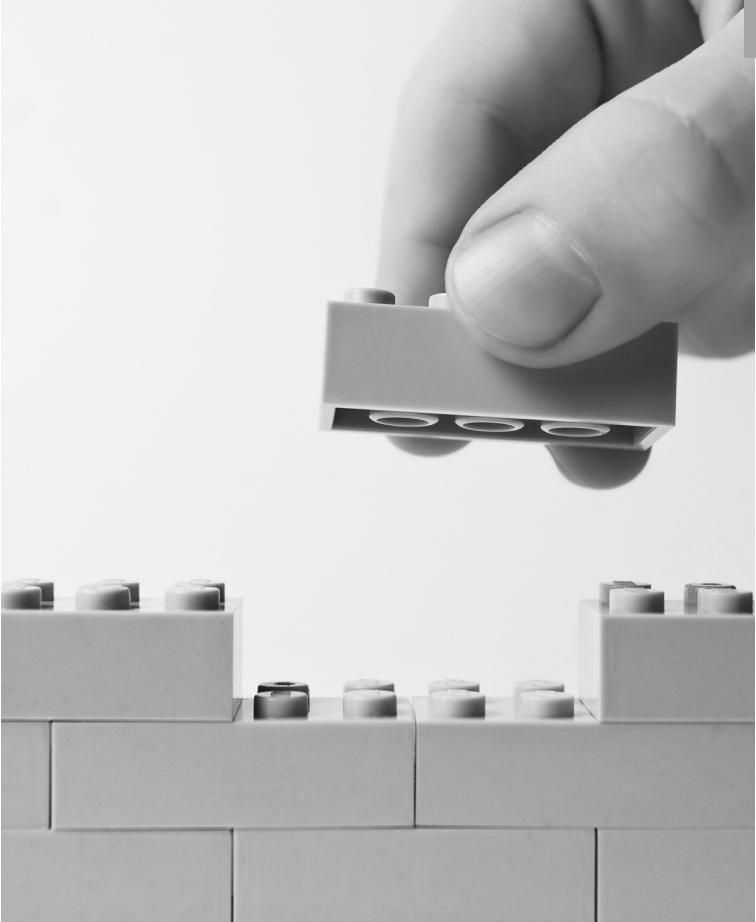
Testen kann Fehlerwirkungen auslösen, die durch Fehlerzustände in der Software verursacht werden, oder kann direkt Fehlerzustände im Testobjekt finden.

Debugging

Die beim Test gefundenen Fehlerwirkungen (die Fehlerzustände) finden, analysieren und beseitigen.

1 – Grundlagen des Testens

1. Was ist Testen?
2. Warum ist Testen notwendig?
3. Testaktivitäten, Testmittel und Rollen des Testens
4. Wesentliche Kompetenzen und bewährte Praktiken beim Testen



1.2 Warum ist Testen notwendig?

- FL-1.2.1 (K2) Sie können Beispiele geben, warum Testen notwendig ist
- FL-1.2.2 (K1) Sie können die Beziehung zwischen Testen und Qualitätssicherung wiedergeben
- FL-1.2.3 (K2) Sie können zwischen Grundursache, Fehlhandlung, Fehlerzustand und Fehlerwirkung unterscheiden

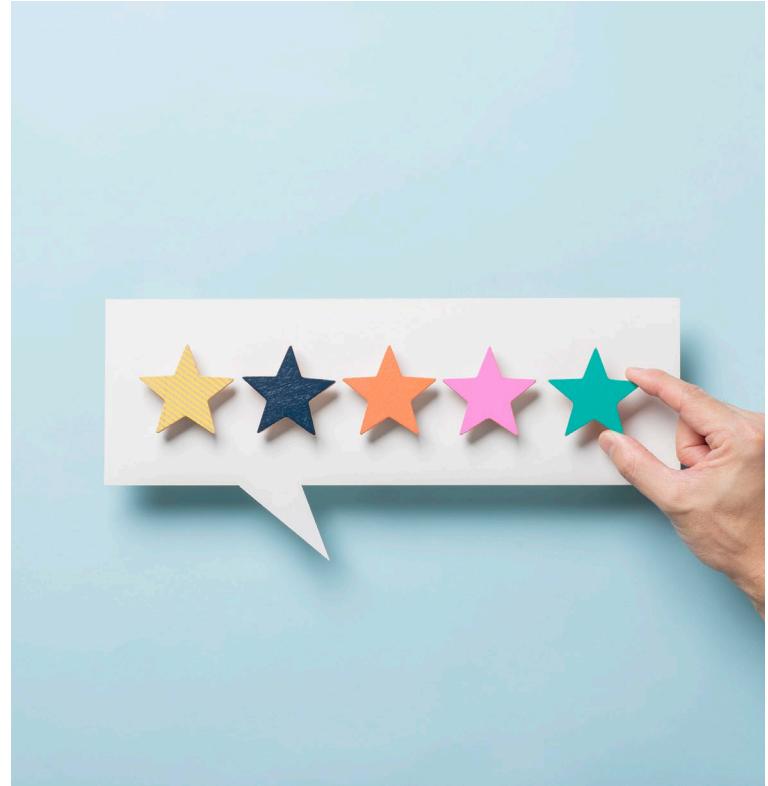
Warum ist Testen notwendig?

Testen, als eine Form der Qualitätssteuerung, trägt dazu bei, die vereinbarten Testziele innerhalb des festgelegten Umfangs sowie der Zeit-, Qualitäts- und Budgetvorgaben zu erreichen.

Der Beitrag des Testens zum Erfolg sollte nicht auf die Aktivitäten des Testteams beschränkt sein.

→ **Jeder Stakeholder kann seine Testkompetenzen einsetzen, um das Projekt dem Erfolg näher zu bringen.**

Das Testen von Komponenten, Systemen und den zugehörigen Arbeitsergebnisse (z. B. Dokumentation) hilft bei der Ermittlung von Fehlerzuständen in der Software.



Der Beitrag des Testens zum Erfolg

Der Beitrag des Testens zum Erfolg I

Testen findet Fehler

Testen ist ein kosteneffizientes Mittel zur Erkennung von Fehlerzuständen.

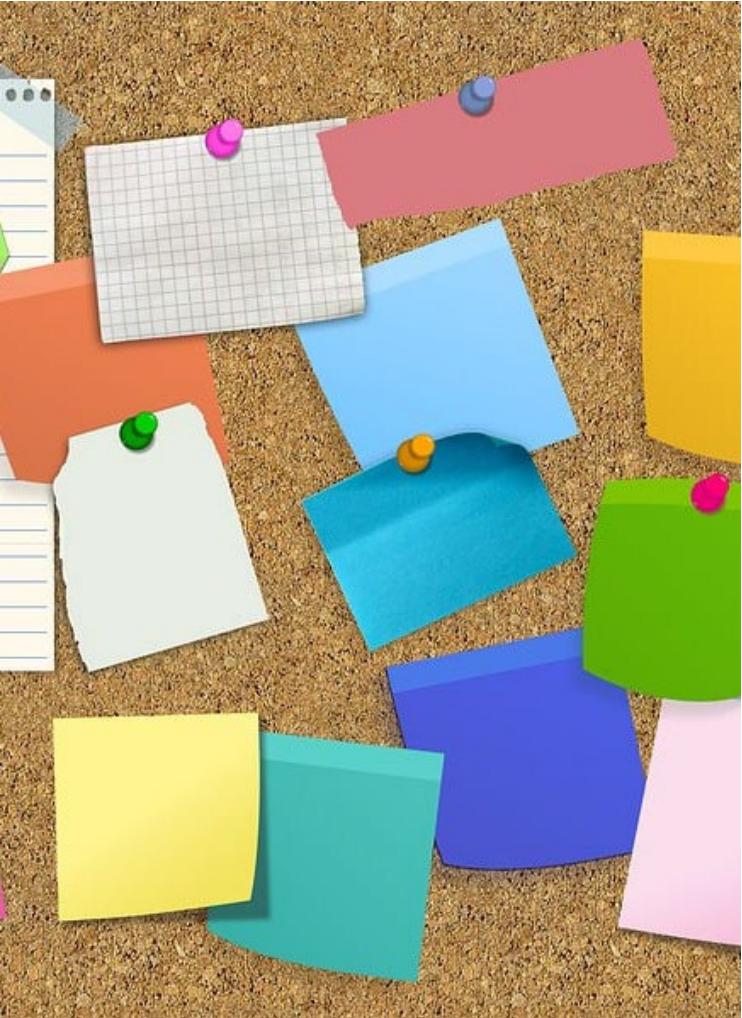
Diese gefundenen Fehlerzustände können dann durch Debugging beseitigt werden, so trägt Testen indirekt zu einer höheren Qualität der Testobjekte bei.

Testen bewertet die Qualität

Durch Testen kann die Qualität eines Testobjekts in verschiedenen Phasen des Softwareentwicklungslebenszyklus bewertet werden.

Diese Messgrößen werden als Teil einer größeren Projektmanagementaktivität verwendet und helfen Entscheidungen für den Übergang zur nächsten Phase zu treffen, z. B. zur Freigabeentscheidung.

Der Beitrag des Testens zum Erfolg – Anforderungsanalyse



Während der Anforderungsanalyse

Das neue Softwareprojekt, dem Sie zugewiesen wurden, befindet sich noch in den frühen Phasen der Anforderungsdefinition.

Businessspezialisten haben potentielle Kunden nach ihren Wünschen befragt und schreiben die ersten Anforderungen zusammen.

- Tester werden in die Anforderungsreviews miteinbezogen. So werden Fehlerzustände bereits in dieser frühen Projektphase aufgedeckt.
- Durch das Identifizieren und Entfernen von Fehlern wird das Risiko reduziert, dass später fehlerhafte oder nicht testbare Features entwickelt werden.

Der Beitrag des Testens zum Erfolg – Systementwurf



Beim Systementwurf

Während des Systementwurfs wird entschieden, aus welchen Teilen die zu entwickelnde Software besteht und wie diese zusammenspielen.

- Die enge Zusammenarbeit von Testern und Systemarchitekten während des Systementwurfs verbessert das gemeinsame Verständnis für den Entwurf und die Möglichkeiten es zu testen.
- Das bessere Verständnis reduziert das Risiko grundlegender Entwurfsfehler.
- Potenzielle Tests werden zu einem frühen Zeitpunkt identifiziert.

Der Beitrag des Testens zum Erfolg II

Testen fördert gemeinsames Verständnis

Testen liefert dem Entwicklungsprojekt eine indirekte Darstellung der Benutzersicht.

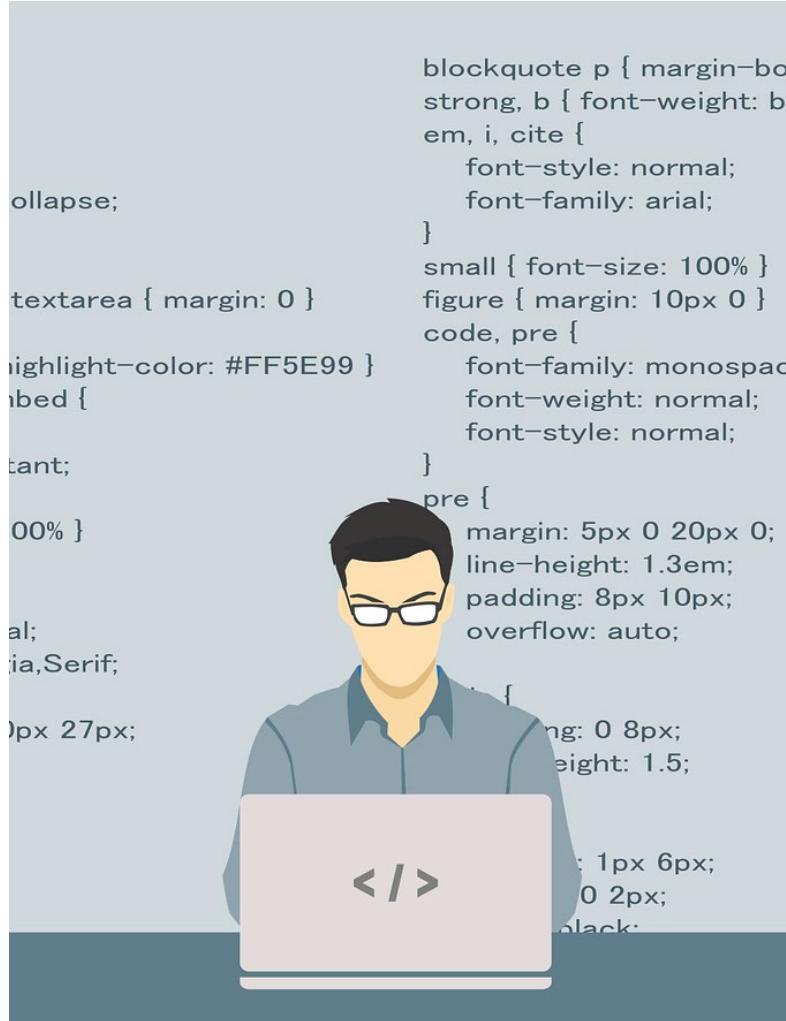
Tester stellen sicher, dass ihr Verständnis für die Bedürfnisse der Benutzer während der gesamten Entwicklung berücksichtigt werden.

Alternativ kann eine repräsentative Gruppe von Benutzern in das Entwicklungsprojekt einbezogen sein, was oft aufgrund hoher Kosten und mangelnder Verfügbarkeit geeigneter Benutzer nicht möglich ist.

Testen prüft, ob Vorgaben erfüllt sind

Testen kann erforderlich sein, um vertragliche oder gesetzliche Anforderungen zu erfüllen oder um regulatorischen Standards zu entsprechen.

Der Beitrag des Testens zum Erfolg – Entwicklung



Während der Programmcode-Entwicklung

Der Programmcode für das neue Programm wird entwickelt.
Das ist typischerweise die Aufgabe des Entwicklungsteams.

- Wenn Tester und Entwicklungsteam während der Entwicklung eng zusammenarbeiten, verbessert das das gemeinsame Verständnis der Beteiligten für den Programmcode und wie dieser zu testen ist.
- Dieses bessere Verständnis verringert das Risiko von Fehlerzuständen sowohl im Programmcode als auch im Test.

Der Beitrag des Testens zum Erfolg – vor der Freigabe



Vor der Freigabe der Software

Die Freigabe des Programms in wenigen Wochen steht bevor.

- Die Verifizierung und Validierung der Software durch Tester vor der Freigabe deckt Fehlerwirkungen auf, die andernfalls übersehen worden wären.
- Die Wahrscheinlichkeit wird erhöht, dass die Software ...
 - den Anforderungen entspricht
 - die Bedürfnisse der Stakeholder erfüllt



Testen und Qualitätssicherung

Testen und Qualitätssicherung (QS)

Obwohl die Begriffe "Testen" und "Qualitätssicherung" (QS) häufig synonym verwendet werden, sind Testen und Qualitätssicherung nicht dasselbe.

Qualitätssicherung (QS)



Qualitätssicherung (QS)

Qualitätssicherung ...

- ... ist ein **prozessorientierter, präventiver Ansatz**, der sich auf die Implementierung und Verbesserung von Prozessen konzentriert.
- ... geht davon aus, dass ein guter Prozess ein gutes Produkt hervorbringt, wenn er korrekt durchgeführt wird.
- ... bezieht sich auf den Entwicklungs- und auf den Testprozess.
- ... liegt in der Verantwortung aller Projektbeteigter.

Testergebnisse liefern Rückmeldungen darüber, wie gut die Entwicklungs- und Testprozesse funktionieren.



Testen

Testen ...

- ... ist ein **produktorientierter, korrigierender Ansatz**.
- ... konzentriert sich auf jene Aktivitäten, die das Erreichen eines angemessenen Qualitätsniveaus unterstützen.
- ... ist eine der wichtigsten Formen der Qualitätssteuerung, andere sind formale Methoden (Modellprüfung und Korrektheitsnachweis), Simulation und Prototyping.

Testergebnisse werden in der Qualitätssicherung verwendet, um Fehlerzustände zu beheben.



Schlüsselbegriffe – Qualität und Qualitätssicherung

Qualität

Der Grad, zu dem ein Arbeitsergebnis die expliziten und impliziten Bedürfnisse seiner Stakeholder erfüllt.

Qualitätssicherung

Aktivitäten, die darauf fokussieren, Vertrauen in die Erfüllung der Qualitätsanforderungen zu erzeugen.

Testen

Der Prozess innerhalb des Softwareentwicklungslebenszyklus, der die Qualität einer Komponente oder eines Systems und der zugehörigen Arbeitsergebnisse bewertet.

Fehlhandlungen, Fehlerzustände, Fehlerwirkungen und Grundursachen

Fehlhandlung



Fehlhandlungen (Irrtümer)

Menschen begehen aus verschiedenen Gründen Fehlhandlungen, wie z. B. durch

- Zeitdruck
- Komplexität von Arbeitsergebnissen
- Prozesse
- Infrastruktur oder Interaktionen
- Erschöpfung
- unzureichende Schulung

Fehlhandlungen können zu Fehlerzuständen (Defekten) führen.

Fehlerzustand



Fehlerzustand (Defekt)

Fehlerzustände können gefunden werden ...

- in der Dokumentation, z. B. in einer Anforderungsspezifikation
- in einem Testskript
- im Quellcode
- in einem unterstützenden Arbeitsergebnis, z. B. Build-Datei

Wenn Fehlerzustände ausgeführt werden, kann es immer, nur unter bestimmten Umständen oder nie zu einer Fehlerwirkung führen.

Ein nicht erkannter Fehlerzustand in einem Arbeitsergebnis kann zu Folgefehlern in weiteren Arbeitsergebnissen führen.

Fehlerwirkung



Fehlerwirkung

Wenn ein Fehlerzustand im Code ausgeführt wird, könnte das System

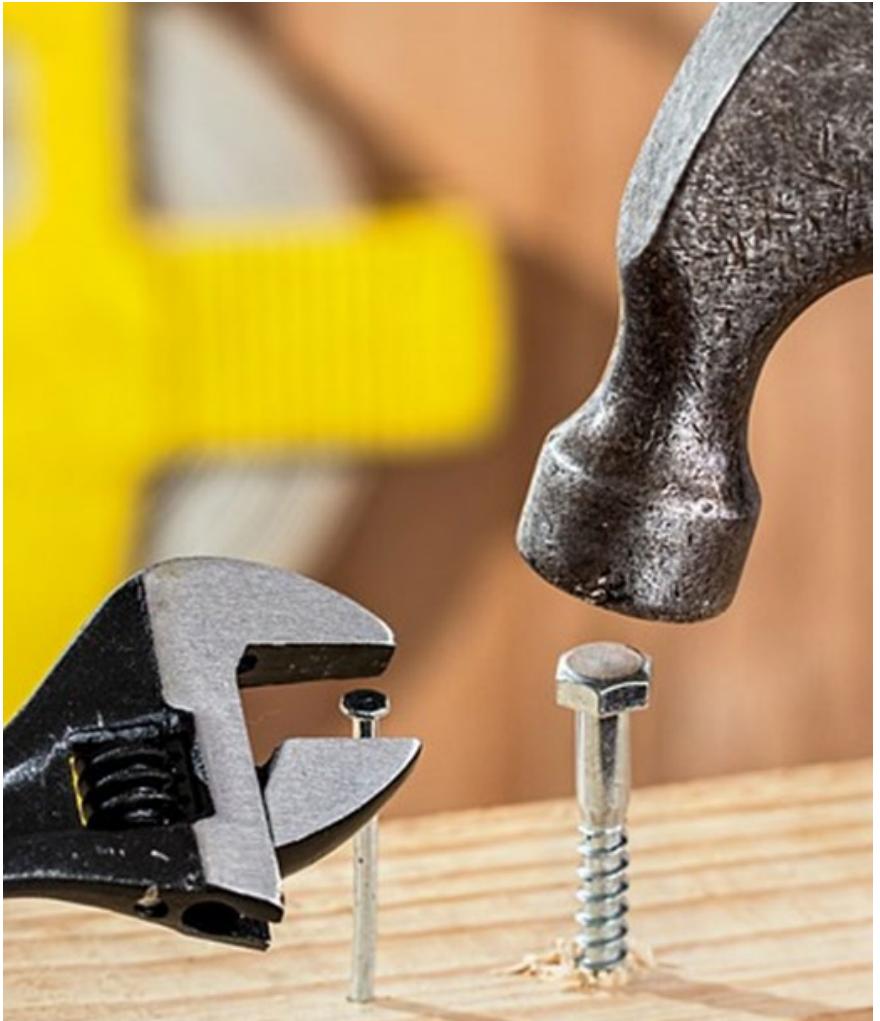
- nicht das tun, was es tun sollte
- etwas tun, was es nicht tun sollte

was zu einer Fehlerwirkung führt.

Fehlerwirkungen können auch durch Umweltbedingungen verursacht werden.

Z. B. Strahlung oder elektromagnetische Felder können Fehlerzustände in der Firmware verursachen.

Grundursache



Grundursache

Eine Grundursache ist ein wesentlicher Grund für das Auftreten eines Problems.

Z. B. eine Situation, die zu einer Fehlhandlung führt

Grundursachen werden durch eine Grundursachenanalyse ermittelt, die normalerweise durchgeführt wird, wenn eine Fehlerwirkung auftritt oder ein Fehlerzustand festgestellt wird.

Es wird davon ausgegangen, dass wenn die Grundursache angegangen wird, weitere ähnliche Fehlerwirkungen oder Fehlerzustände verhindert oder ihre Häufigkeit verringert werden können.

Z. B. durch Beseitigung der Grundursache



Grundursachenanalyse

Szenario

Wegen eines Fehlers einer Banksoftware werden Zinsen falsch ausbezahlt.
Daraus resultieren Kundenbeschwerden.

Fehlerwirkung

Auswirkung

Ergebnis der Fehleranalyse

Fehlerhafter Code

Fehlerzustand

Fehlhandlung

Ergebnis der Grundursachenanalyse

Missverständnis beim Schreiben der User-Story zur Zinsberechnung
wegen einer mehrdeutigen Anforderung, da das Team in der Fachlichkeit unzureichendes Know-how hatte.

Grundursache

→ Verbesserungsmaßnahme

Schulung des Teams in der Fachlichkeit von Zinsberechnungen

Siehe auch Lehrplan ISTQB-ETM Expert Level Test Management und Lehrplan ISTQB-EITP Expert Level Improving the Test Process



Schlüsselbegriffe – Grundlagen II

Grundursache

Die Ursache eines Fehlerzustands. Wenn man sie behebt, dann wird das Vorkommen der Fehlerart reduziert oder eliminiert.

Fehlhandlung

Die menschliche Handlung, die zu einem falschen Ergebnis führt.

Fehlerzustand

Eine Unzulänglichkeit oder ein Mangel in einem Arbeitsergebnis, sodass es seine Anforderungen oder Spezifikationen nicht erfüllt.

Fehlerwirkung

Ein Ereignis in welchem eine Komponente oder ein System eine geforderte Funktion nicht im spezifizierten Rahmen ausführt.



Warum ist Testen notwendig?

Der Beitrag des Testens zum Erfolg

- Testen bewertet die Qualität
- Testen verbessert indirekt die Qualität
- Testen fördert gemeinsames Verständnis
- Testen prüft ob Vorgaben erfüllt sind



Testen und Qualitätssicherung

Qualitätssicherung (QS)

- prozessorientierter, präventiver Ansatz
- konzentriert sich auf die Implementierung und Verbesserung von Prozessen
- bezieht sich auf den Entwicklungs- und Testprozess
- liegt in der Verantwortung aller Projektbeteiligten

Testergebnisse liefern Rückmeldungen darüber, wie gut die Entwicklungs- und Testprozesse funktionieren.

Testen

- produktorientierter, korrigierender Ansatz
- konzentriert sich auf Aktivitäten, die das Erreichen eines angemessenen Qualitätsniveaus unterstützen
- eine der wichtigsten Formen der Qualitätssteuerung

Testergebnisse werden von der QS verwendet, um Fehlerzustände zu beheben.



Fehlhandlung, Fehlerzustand, Fehlerwirkung, Grundursache



Grundursache

Ein wesentlicher Grund für das Auftreten eines Problems.

Wenn man ihn behebt, dann wird das Vorkommen der Fehlerart reduziert oder eliminiert.



Fehlhandlung

Die menschliche Handlung, die zu einem falschen Ergebnis führt.



Fehlerzustand

Eine Unzulänglichkeit oder ein Mangel in einem Arbeitsergebnis, sodass es seine Anforderungen oder Spezifikationen nicht erfüllt.

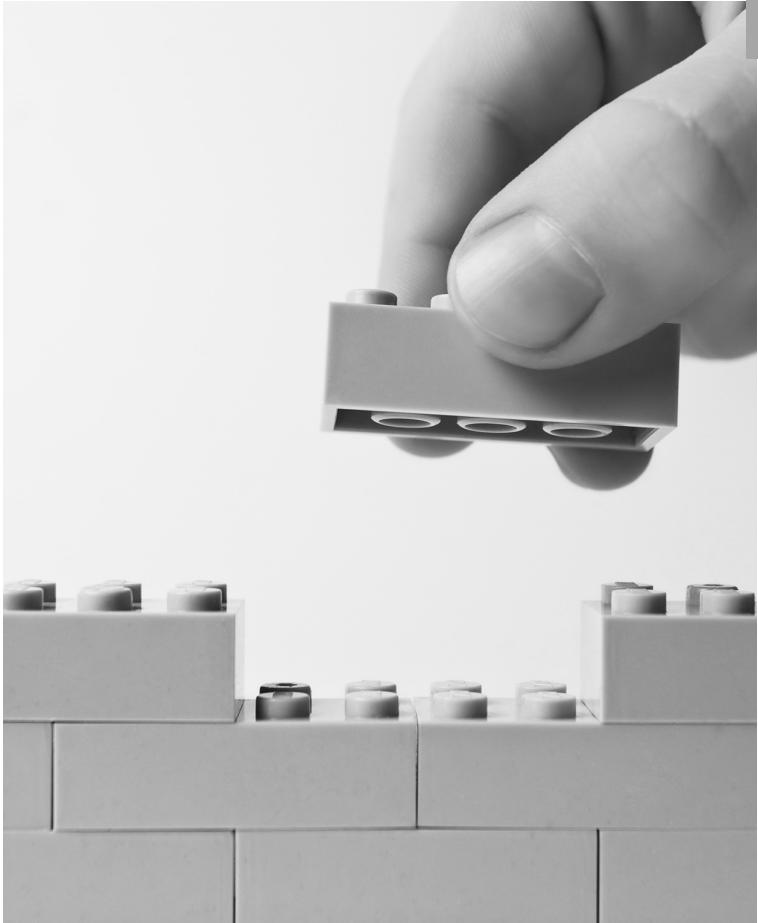


Fehlerwirkung

Ein Ereignis in welchem eine Komponente oder ein System eine geforderte Funktion nicht im spezifizierten Rahmen ausführt.

1 – Grundlagen des Testens

1. Was ist Testen?
2. Warum ist Testen notwendig?
3. Testaktivitäten, Testmittel und Rollen des Testens
4. Wesentliche Kompetenzen und bewährte Praktiken beim Testen



1.4 Testaktivitäten, Testmittel und Rollen des Testens

- FL-1.4.1 (K2) Sie können die verschiedenen Testaktivitäten und die damit verbundenen Aufgaben erklären
- FL-1.4.2 (K2) Sie können die Auswirkungen des Kontexts auf den Testprozess erklären
- FL-1.4.3 (K2) Sie können Testmittel, die die Testaktivitäten unterstützen, unterscheiden
- FL-1.4.4 (K2) Sie können die Bedeutung der Pflege der Verfolgbarkeit erklären
- FL-1.4.5 (K2) Sie können die verschiedenen Rollen beim Testen vergleichen

Testaktivitäten, Testmittel und Rollen des Testens

- Es gibt nicht den einen universellen Softwaretestprozess, es gibt eine Reihe von gebräuchlichen Testaktivitäten, die den Testprozess bilden.
- Im Rahmen der Testplanung werden die Testaktivitäten an die vorgegebene Projektsituation angepasst.
- Ohne Testaktivitäten werden die festgelegten Ziele mit einer weit geringeren Wahrscheinlichkeit erreicht.

Die folgenden Abschnitte beschreiben

- die Testaktivitäten und -aufgaben im Testprozess
- den Einfluss des Kontexts auf den Testprozess
- die Verfolgbarkeit zwischen Testbasis und Testmitteln
- die Rollen im Testen

Die Norm ISO/IEC/IEEE 29119-2 enthält weitere Informationen über Testprozesse.

Testaktivitäten und -aufgaben

Testprozess



Ein Testprozess besteht in der Regel aus Hauptgruppen von Aktivitäten.

Auch wenn hier logisch sequenziell aufgelistet, die Aktivitäten werden oft iterativ oder parallel durchgeführt.

Oft sind Testaktivitäten auf das System und das Projekt zugeschnitten.



Test eines Taschenrechners – Testaktivitäten und -aufgaben



Die nebenstehende Grafik finden Sie in Ihren
Unterlagen als PDF:
Beispiel-Taschenrechner-Testprozess-Verfolgbarkeit.pdf

Testplanung



Aufgaben und Aktivitäten

- Testziele definieren
- eine Testvorgehensweise auswählen, um die Testziele zu erreichen, innerhalb der durch den Gesamtkontext auferlegten Randbedingungen

Die Testplanung wird in Kapitel 5 näher erläutert.

Testanalyse



Die Testanalyse beantwortet die Frage "**Was soll getestet werden?**" in Form von messbaren Überdeckungskriterien.

Aufgaben und Aktivitäten

Prüfung der Testbasis (z. B. Anforderungen, Spezifikationen, User-Storys, ...) und des Testobjektes auf Fehlerzustände und deren Testbarkeit durch:

- Analyse der Testbasis, um testbare Merkmale zu identifizieren
- Bestimmung und Priorisierung der zugehörigen Testbedingungen, zusammen mit den damit verbundenen Risiken und Risikostufen
- Einsatz von unterstützenden Testverfahren

Testentwurf



Der Testentwurf beantwortet die Frage "**Wie soll getestet werden?**"

Aufgaben und Aktivitäten

- Ausarbeiten von Testbedingungen zu Testfällen und anderen Testmitteln (z. B. Test-Chartas)
- Überdeckungselemente identifizieren, diese dienen als Leitfaden für die Spezifizierung der Testfalleingaben
- Einsatz von unterstützenden Testverfahren
- Anforderungen an die Testdaten definieren
- Testumgebung entwerfen und notwendige Infrastruktur und Werkzeuge identifizieren

Testrealisierung



Aufgaben und Aktivitäten

- die für die Testdurchführung erforderlichen Testmittel erstellen oder beschaffen (z. B. Testdaten)
- Testfälle in Testabläufen organisieren und zu Testsuiten zusammenstellen, sofern sinnvoll
- manuelle und automatisierte Testskripte erstellen
- Testabläufe priorisieren und in einem Testausführungsplan für effiziente Testdurchführung anordnen
- die Testumgebung aufbauen und auf korrekte Einrichtung prüfen

Testdurchführung



Aufgaben und Aktivitäten

- Test gemäß dem Testausführungsplan (Testläufe) ausführen, manuell oder automatisiert
- passende Form für den Test wählen, z. B. kontinuierlicher Test, Testsitzungen in Paaren
- tatsächliche Ergebnisse mit erwarteten Ergebnissen vergleichen
- Testergebnisse protokollieren
- Analyse von Fehlerwirkungen, um ihre wahrscheinlichen Ursachen zu ermitteln
- Berichterstattung über die Anomalien auf der Grundlage der beobachteten Fehlerwirkungen und deren Analyse

Testüberwachung und Teststeuerung



Aufgaben und Aktivitäten

Testüberwachung

- laufende Überprüfung aller Testaktivitäten
- Vergleich des tatsächlichen Fortschritts mit dem Plan

Teststeuerung

- Korrekturmaßnahmen ergreifen, um Testziele zu erreichen

Testüberwachung und Teststeuerung werden in Kapitel 5 näher erläutert.

Testabschluss



Der Testabschluss findet in der Regel zu Projektmeilensteinen statt (z. B. Freigabe, Ende der Iteration, Abschluss der Teststufe).

Aufgaben und Aktivitäten

- Änderungsanträge (Change Requests) oder Produkt-Backlog-Einträge für alle nicht behobenen Fehlerzustände erstellen
- alle Testmittel, die für die Zukunft nützlich sein könnten, identifizieren und archivieren oder an die entsprechenden Teams übergeben
- Testumgebung in einen vereinbarten Zustand bringen
- Testaktivitäten analysieren, um Lessons Learned und Verbesserungen für zukünftige Iterationen, Releases oder Projekte zu ermitteln
- Testabschlussbericht erstellen und an Stakeholder kommunizieren



Schlüsselbegriffe – Testaktivitäten

Testabschluss

Die Aktivität, die Testmittel für eine spätere Anwendung verfügbar macht, Testumgebungen in einem zufriedenstellenden Zustand hinterlässt, und die Testergebnisse an die relevanten Stakeholder übermittelt.

Testplanung

Eine Aktivität im Testprozess zur Erstellung und Fortschreibung des Testkonzepts.

Testanalyse

Die Aktivität, die Testbedingungen durch eine Analyse der Testbasis identifiziert.

Testrealisierung

Die Aktivität, die auf Basis der Testanalyse und des -Entwurfs die Testmittel vorbereitet, welche für die Testdurchführung benötigt werden.

Testdurchführung

Die Aktivität der Ausführung eines Tests für eine Komponente oder ein System, die Istergebnisse erzeugt.

Teststeuerung

Die Aktivität, die Korrekturmaßnahmen entwickelt und anwendet, um ein Testprojekt auf den richtigen Weg zu bringen, wenn es vom Plan abweicht.

Testentwurf

Die Aktivität, die Testfälle aus Testbedingungen ableitet und spezifiziert.

Testüberwachung

Die Aktivität, die den Status von Testaktivitäten überprüft, alle Abweichungen vom Plan oder der Erwartung identifiziert und den Status an die Stakeholder meldet.

Testprozess im Kontext

Testprozess im Kontext I



Testen wird nicht isoliert durchgeführt.

Alle Testaktivitäten sind ein integraler Bestandteil der Entwicklungsprozesse innerhalb einer Organisation.

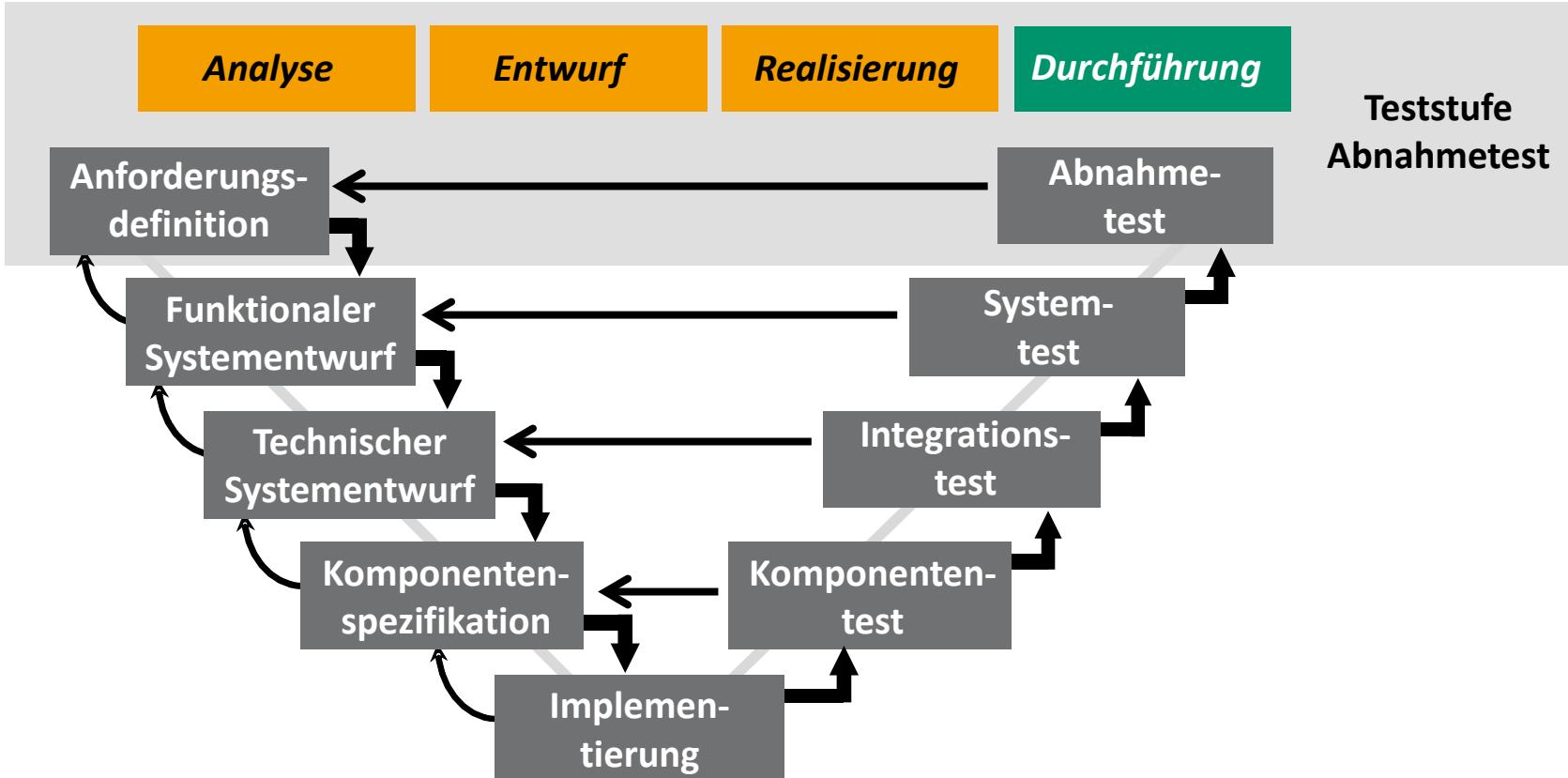
Das Testen wird auch von den Stakeholdern finanziert und soll letztendlich dazu beitragen, die Geschäftsanforderungen der Stakeholder zu erfüllen.

Daher hängt die Art und Weise, wie das Testen durchgeführt wird, von einer Reihe von Kontextfaktoren ab.

Realisierung des Testprozesses I



Sequielles Softwareentwicklungsmodell



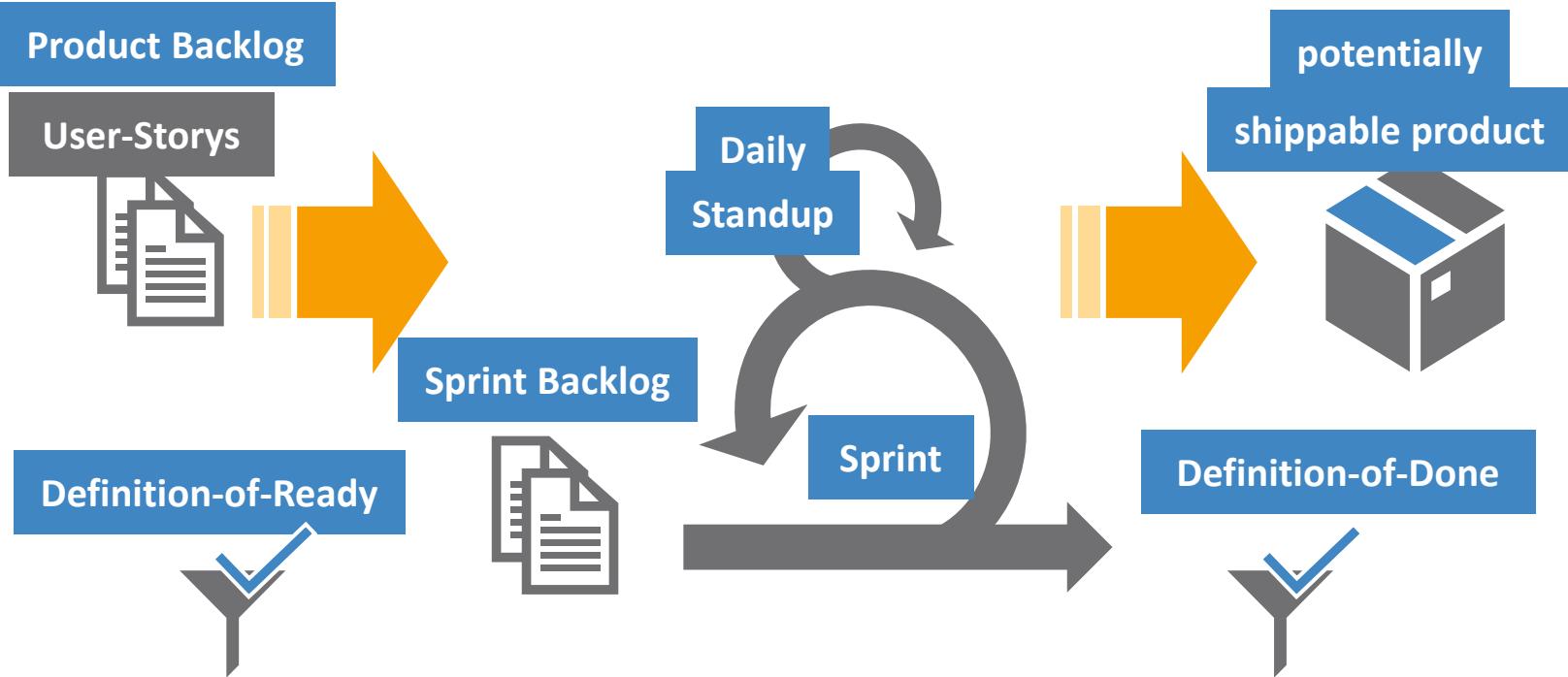
Die schrittweise logische Abfolge der Aktivitäten beinhaltet Überschneidungen, Kombinationen, Parallelität oder Auslassungen.

→ In der Regel ist eine Anpassung dieser Hauptaktivitäten an den Kontext des Systems und des Projekts erforderlich.



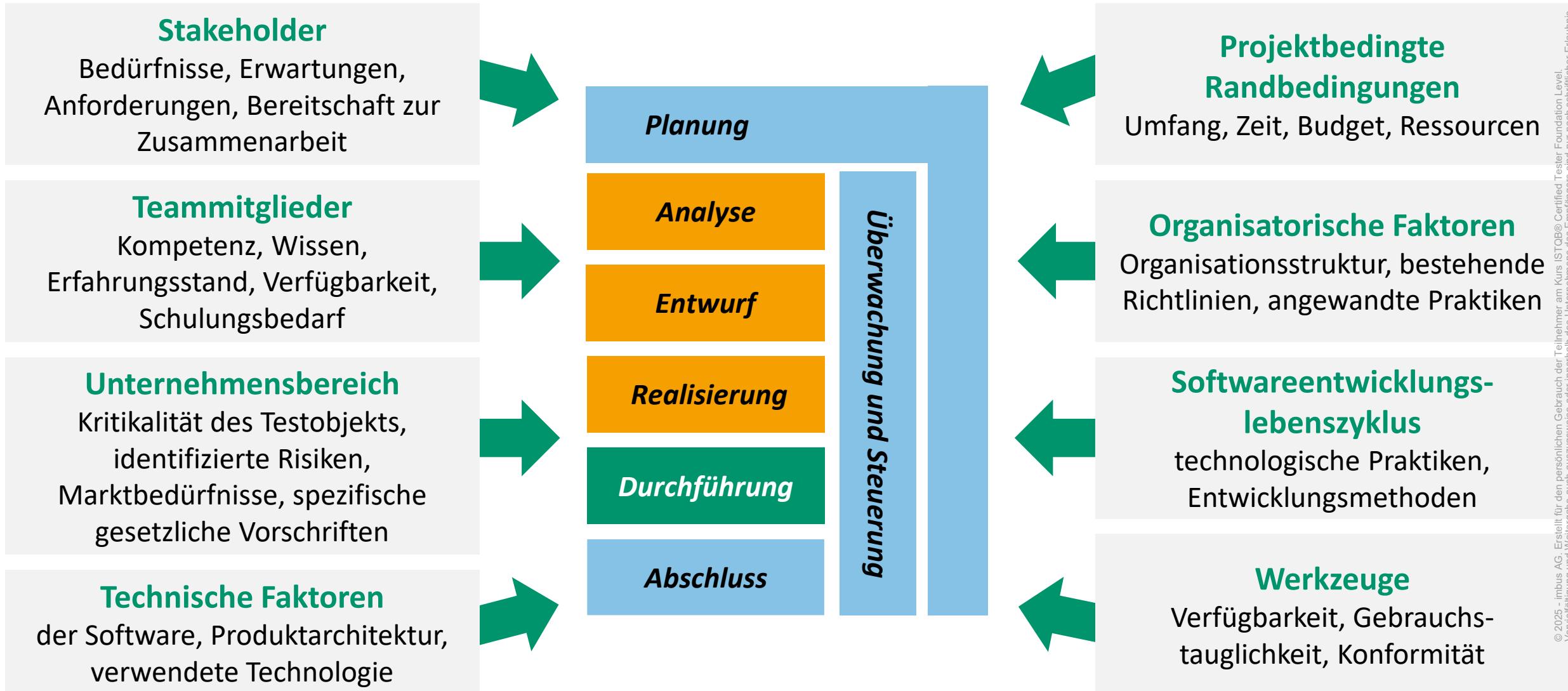
Realisierung des Testprozesses II

Agiles Softwareentwicklungsmodell



- Kleine Iterationen von Software-Design, -Erstellung und -Test
 - Unterstützt durch fortlaufende Planung
- Die Testaktivitäten werden bei diesem Entwicklungsansatz iterativ und kontinuierlich durchgeführt.

Testprozess im Kontext II



Testprozess im Kontext III

Diese Faktoren haben Auswirkungen auf viele testbezogene Aspekte, darunter:

- Teststrategie
- verwendete Testverfahren
- Grad der Testautomatisierung
- geforderte Überdeckung
- Detaillierungsgrad der Testmittel
- Testberichterstattung

Testmittel



Arbeitsergebnisse

Testmittel werden als Arbeitsergebnisse aus den Testaktivitäten erstellt.

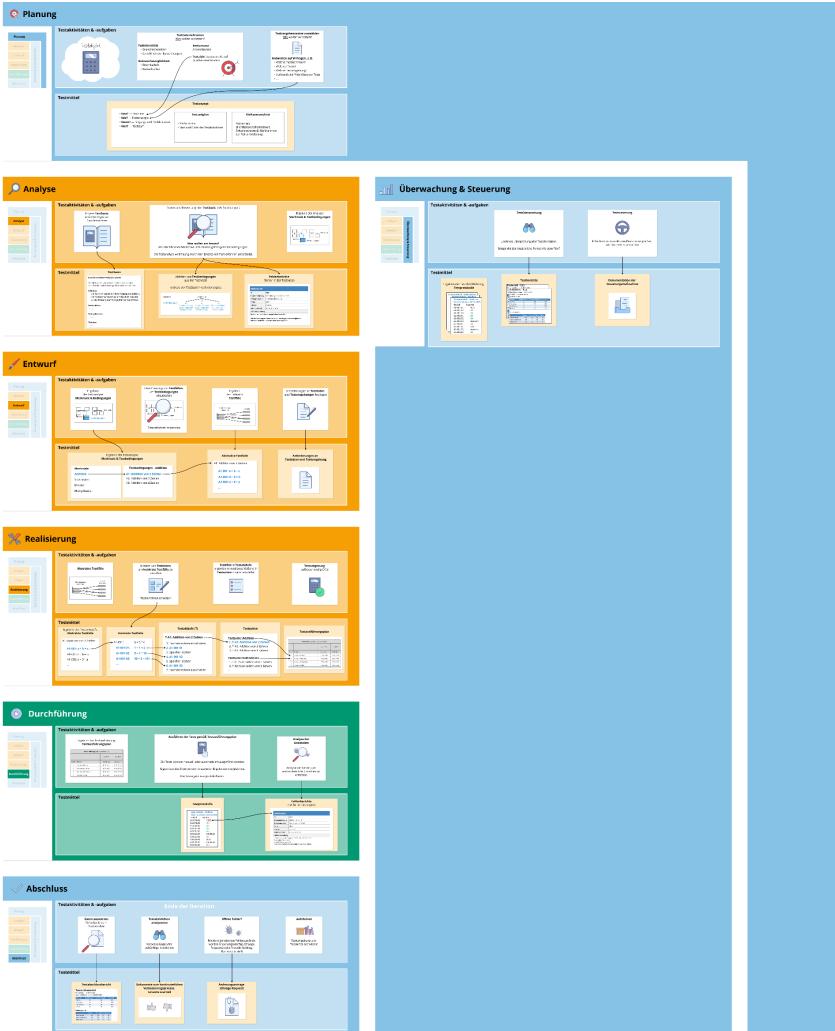
Es gibt erhebliche Unterschiede in der Art und Weise, wie verschiedene Organisationen ihre Arbeitsergebnisse erstellen, gestalten, benennen, organisieren und verwalten.

Die folgende Liste der Arbeitsergebnisse erhebt keinen Anspruch auf Vollständigkeit.



Test eines Taschenrechners – Testmittel

Testmittel, Testaktivitäten und -aufgaben im Testprozess



Die nebenstehende Grafik finden Sie in Ihren
Unterlagen als PDF:
Beispiel-Taschenrechner-Testprozess-Verfolgbarkeit.pdf

Testplanung – Arbeitsergebnisse



Arbeitsergebnisse der Testplanung

- Testkonzept
 - Testzeitplan
 - Risikoverzeichnis
 - Eingangskriterien und Endekriterien

Testzeitplan, Risikoverzeichnis sowie Eingangskriterien und Endekriterien sind häufig **Teil des Testkonzepts**.

Das **Risikoverzeichnis** ist eine Liste von Risiken zusammen mit der Eintrittswahrscheinlichkeit des Risikos, den Auswirkungen des Risikos und Informationen zur Risikominderung.

Testanalyse – Arbeitsergebnisse



Arbeitsergebnisse der Testanalyse

- (priorisierte) Testbedingungen
- Akzeptanzkriterien
- Fehlerberichte über Fehlerzustände in der Testbasis
(falls nicht direkt behoben)

Testentwurf – Arbeitsergebnisse



Arbeitsergebnisse des Testentwurfs

- (priorisierte) Testfälle
- Test-Chartas
- Überdeckungselemente
- Anforderungen an Testdaten
- Anforderungen an Testumgebungen

Testrealisierung – Arbeitsergebnisse



Arbeitsergebnisse der Testrealisierung

- Testabläufe
- manuelle und automatisierte Testskripte
- Testsuiten
- Testdaten
- Testausführungspläne
- Bestandteile der Testumgebung
z. B. Platzhalter, Treiber, Simulatoren, Service-Virtualisierung

Testdurchführung – Arbeitsergebnisse



Arbeitsergebnisse der Testdurchführung

- Testprotokolle
- Fehlerberichte

Testüberwachung und Teststeuerung – Arbeitsergebnisse



Arbeitsergebnisse der Testüberwachung und Teststeuerung

- Testfortschrittsberichte
- Dokumentation der Steuerungsmaßnahmen
- Informationen über Risiken

Testabschluss – Arbeitsergebnisse



Arbeitsergebnisse des Testabschlusses

- Testabschlussbericht
- Maßnahmen zur Verbesserung nachfolgender Projekte oder Iterationen
- Change Requests (z. B. als Elemente des Product-Backlogs)
- dokumentierte Lessons Learned

Umfrage



- Wie sieht der Testprozess in ihrem Projekt aus?
- Welche Aktivitäten gibt es?
- Welche Arbeitsergebnisse werden erstellt?
- Wie heißen die Arbeitsergebnisse?
- Wie werden die Arbeitsergebnisse dokumentiert?



Schlüsselbegriffe – Testmittel

Testbasis

Alle Informationen, die als Basis für die Testanalyse und den Testentwurf verwendet werden können.

Testergebnis

Das Ergebnis und die Konsequenz der Durchführung eines Tests.

Testbedingung

Ein testbarer Aspekt einer Komponente oder eines Systems, der als Grundlage für das Testen identifiziert wurde.

Testdaten

Für die Testdurchführung benötigte Daten.

Testablauf

Eine Folge von Testfällen in der Reihenfolge ihrer Durchführung, mit allen erforderlichen Aktionen zur Herstellung der Vorbedingungen und zum Aufräumen nach der Durchführung.

Testfall

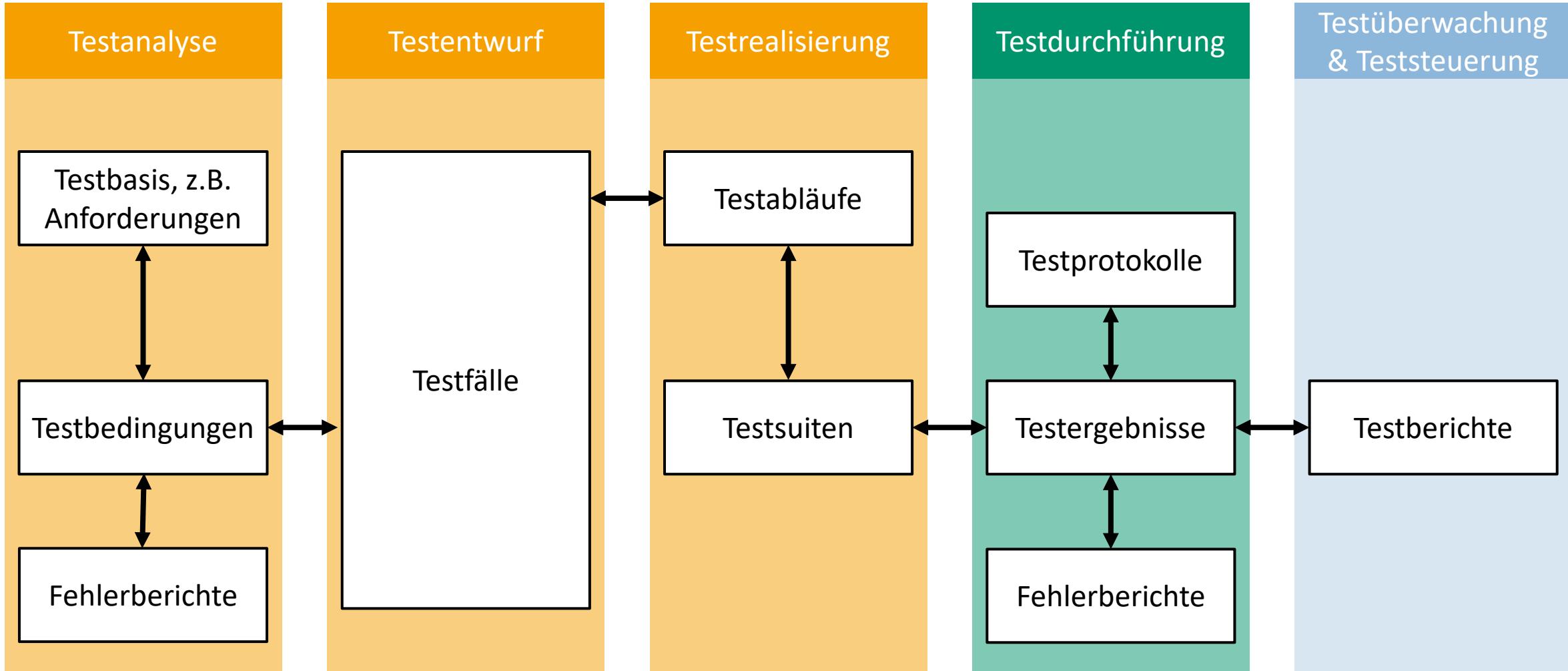
Eine Menge von Vorbedingungen, Eingaben, Aktionen (falls anwendbar), erwarteten Ergebnissen und Nachbedingungen, welche auf Basis von Testbedingungen entwickelt wurden.

Testmittel

Die Arbeitsergebnisse, die während des Testprozesses erstellt werden und dazu gebraucht werden, um die Tests zu planen, zu entwerfen, auszuführen, auszuwerten und darüber zu berichten.

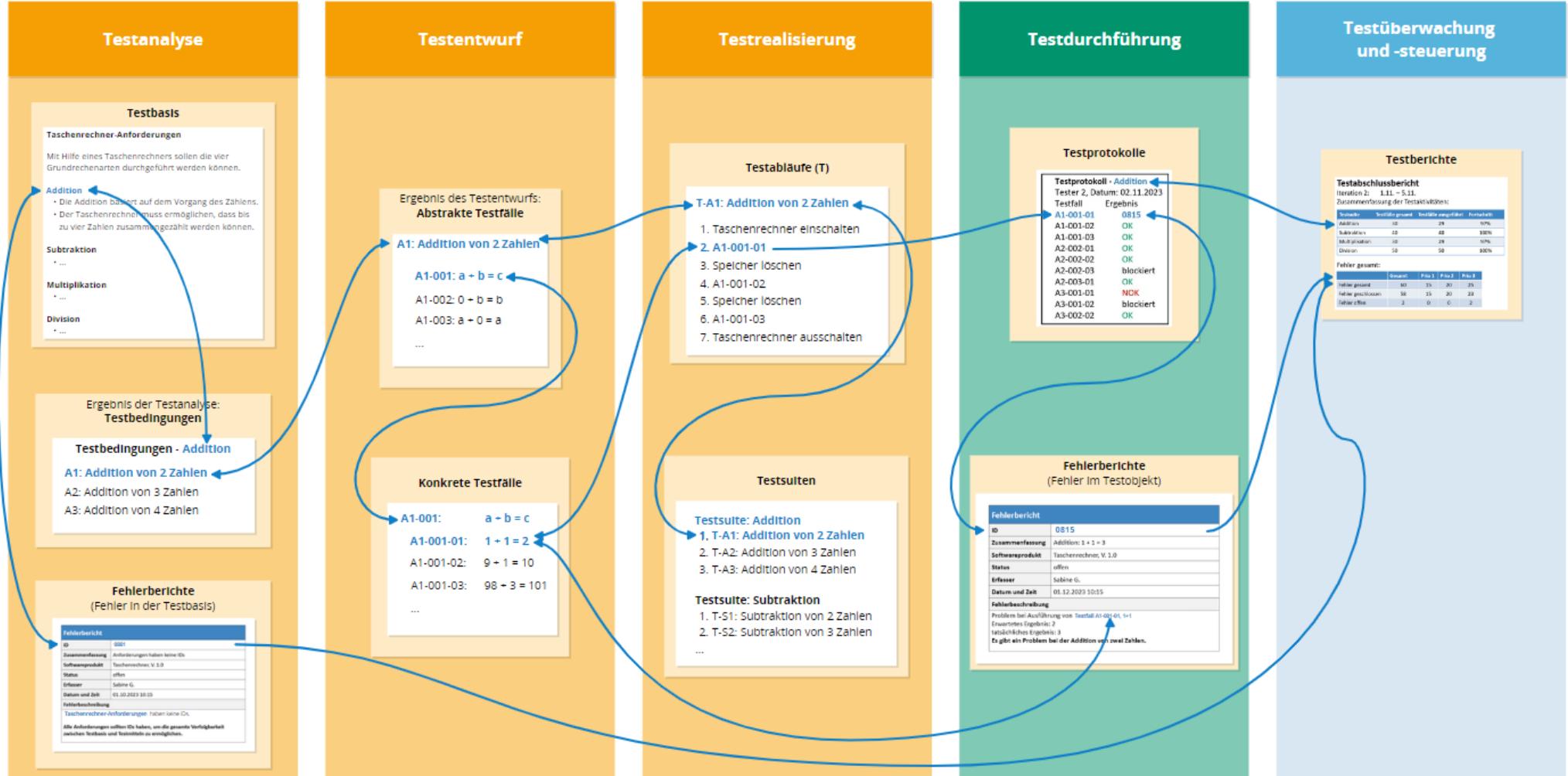
Verfolgbarkeit zwischen Testbasis und Testmitteln

Testmittel im Testprozess





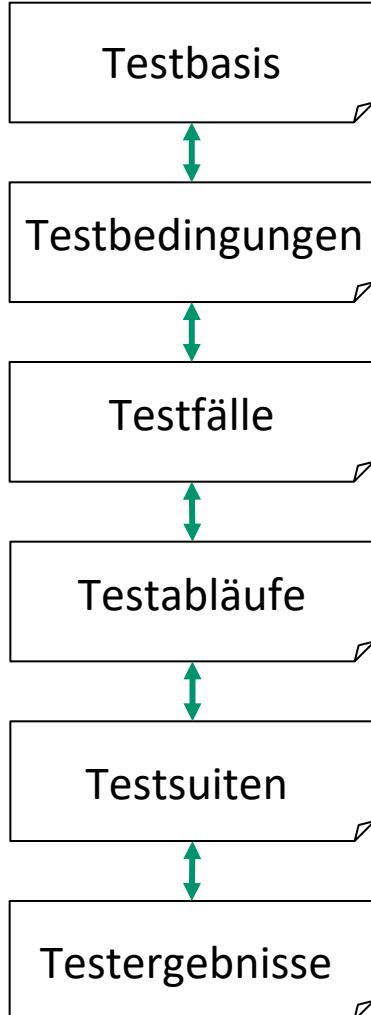
Test eines Taschenrechners – Verfolgbarkeit



Die nebenstehende Grafik finden Sie in Ihren Unterlagen als PDF:

Beispiel-Taschenrechner-Testprozess-Verfolgbarkeit.pdf

Verfolgbarkeit I



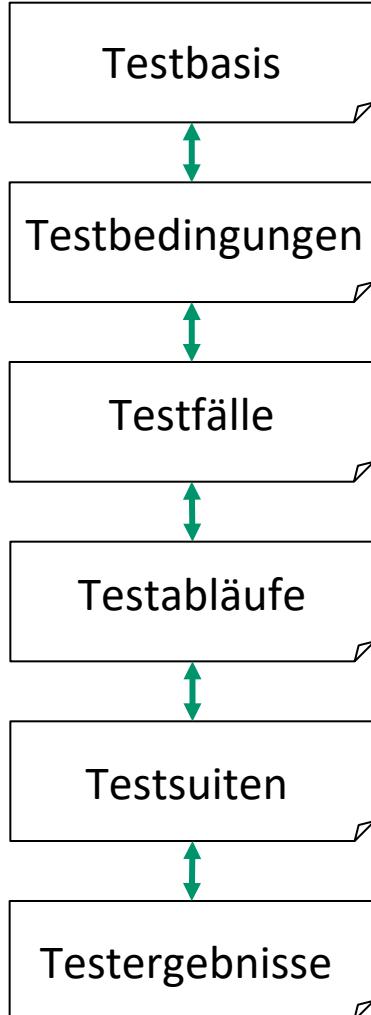
Verfolgbarkeit ermöglicht die **Bewertung der Überdeckung**.
Da ist es sehr nützlich, wenn messbare Überdeckungskriterien in der Testbasis definiert sind.

Die Überdeckungskriterien können als wichtige Key-Performance-Indikatoren (KPI) dienen, um die Aktivitäten zu steuern, die zeigen, inwieweit die Testziele erreicht wurden.

Beispiele für die Verwendung der Verfolgbarkeit

- Testfälle zu Anforderungen → zeigen ob die Anforderungen durch Testfälle abgedeckt werden
- Testergebnisse zu Risiken → bewerten das Ausmaß des Restrisikos des Testobjekts

Verfolgbarkeit II



Neben der Bewertung der Überdeckung ermöglicht eine gute Verfolgbarkeit:

- Ermittlung der Auswirkungen von Änderungen
- Erleichtert Audits
- Hilft bei der Erfüllung von IT-Governance-Kriterien
- Macht Testfortschrittsberichte und Testabschlussberichte leichter verständlich, indem sie den Status der Elemente der Testbasis enthält
 - Dies kann auch dabei helfen, den Stakeholdern die technischen Aspekte des Testens auf verständliche Weise zu vermitteln.
- Liefert Informationen zur Bewertung der Produktqualität, der Prozessfähigkeit und des Projektfortschritts im Vergleich zu den Unternehmenszielen



Schlüsselbegriffe – Verfolgbarkeit

Testprozess

Die Menge zusammenhängender Aktivitäten bestehend aus Testplanung, Testüberwachung und -steuerung, Testanalyse, Testentwurf, Testrealisierung, Testdurchführung und Testabschluss.

Verfolgbarkeit

Die Fähigkeit, explizite Beziehungen zwischen Arbeitsergebnissen oder zwischen Elementen von Arbeitsergebnissen darzustellen.

Rollen des Testens

PRODUCTION
DIRECTOR

Rollen des Testens

In diesem Seminar werden zwei Hauptrollen im Test behandelt

Testmanagement

Die Rolle des Testmanagements kann zum Beispiel von der Teamleitung, dem Testmanagement oder von der Entwicklungsleitung ausgefüllt werden.

Tester

Diese Rolle kann in verschiedenen Formen existieren, z. B. als Testanalytiker, Testprogrammierer, Spezialist.

Wie die Aktivitäten und Aufgaben der beiden Rollen zugewiesen werden, ist abhängig von

- der Organisation
- dem Projekt- und Produktkontext
- den Kompetenzen der Personen

Es ist möglich, dass eine Person gleichzeitig beide Rollen übernimmt.

Testmanagement

Das Testmanagement übernimmt die Gesamtverantwortung für

- den Testprozess
- das Testteam
- die Leitung der Testaktivitäten

Aufgaben

- Testplanung
- Testüberwachung und Teststeuerung
- Testabschlussaktivitäten

Je nach Kontext variiert die Art und Weise, wie die Rolle des Testmanagements ausgeübt wird.



Testmanagement

Tester

Tester übernehmen die Gesamtverantwortung für

- operative Aspekte des Testens

Aufgaben

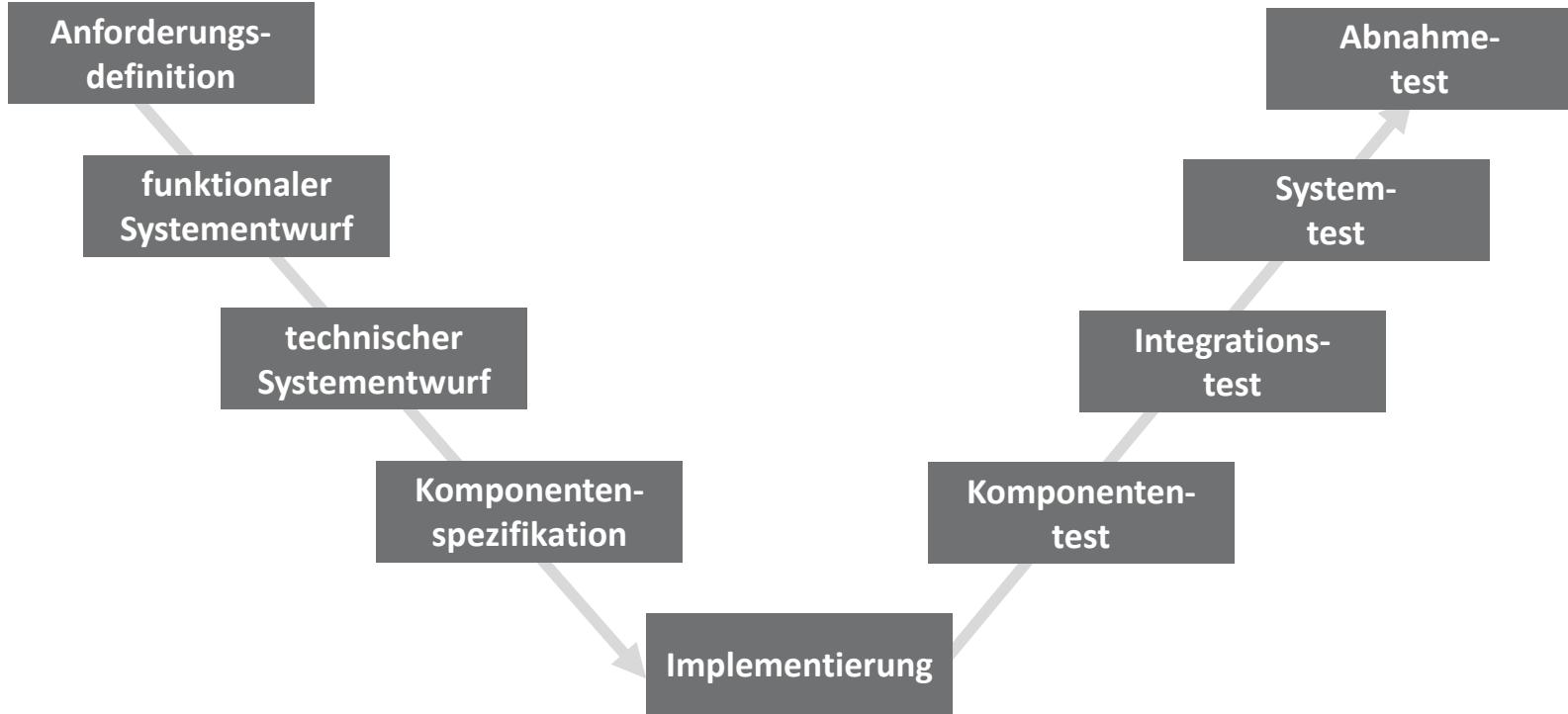
- Testanalyse
- Testentwurf
- Testrealisierung
- Testdurchführungsaktivitäten



Testen



Rollen im sequenziellen Entwicklungsmodell



Die Testmanagement-Rolle ist zuständig für

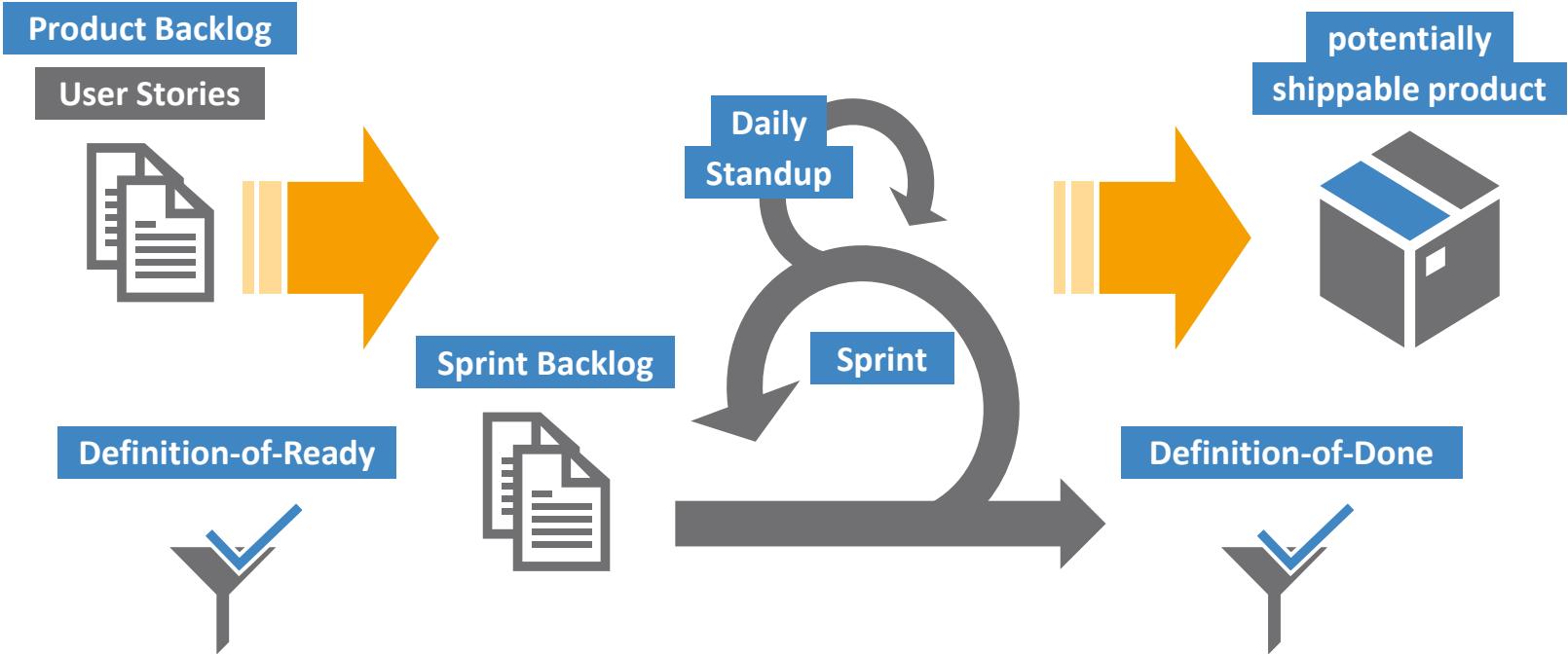
- Testplanung
- Testüberwachung
- Teststeuerung
- Testabschluss

Die Tester-Rolle ist verantwortlich für

- Testanalyse
- Testentwurf
- Testrealisierung
- Testdurchführung



Rollen im agilen Entwicklungsmodell



Testmanagement-Rolle

- In agilen Projekten kann ein Teil der Testmanagement-aufgaben vom agilen Team übernommen werden.
- Aufgaben, die sich über mehrere Teams oder die gesamte Organisation erstrecken, können von Testmanagern außerhalb des Teams durchgeführt werden.

Tester sind Teil des Teams



Testaktivitäten und Aufgaben

Testanalyse <i>„Was ist zu testen?“</i>	Testentwurf <i>„Wie wird getestet?“</i>	Testrealisierung <i>„Ist alles für die Durchführung der Tests bereit?“</i>	Testdurchführung <i>Ausführen der Tests</i>
<ul style="list-style-type: none">▪ Analyse der Testbasis▪ Identifizieren von testbaren Merkmalen▪ Testbedingungen bestimmen und priorisieren	<ul style="list-style-type: none">▪ Testfälle entwerfen, um Testbedingungen abzudecken▪ Testverfahren einsetzen▪ Anforderungen an Testdaten und Testumgebung identifizieren	<ul style="list-style-type: none">▪ Testmittel (z.B. Testdaten) bereitstellen▪ Erstellung von Testabläufen und Testsuiten▪ automatisierte Testskripte erstellen▪ Testausführungsplan erzeugen▪ Aufbau und Prüfung der Testumgebung	<ul style="list-style-type: none">▪ Ausführen der Tests nach Plan (manuell oder werkzeuggestützt)▪ Ergebnisse vergleichen▪ Analyse der Anomalien▪ Aufzeichnung der Testergebnisse▪ Fehlerberichterstattung



Testprozess im Kontext

Die Art und Weise, wie das Testen durchgeführt wird, hängt von einer Reihe von Kontextfaktoren ab:

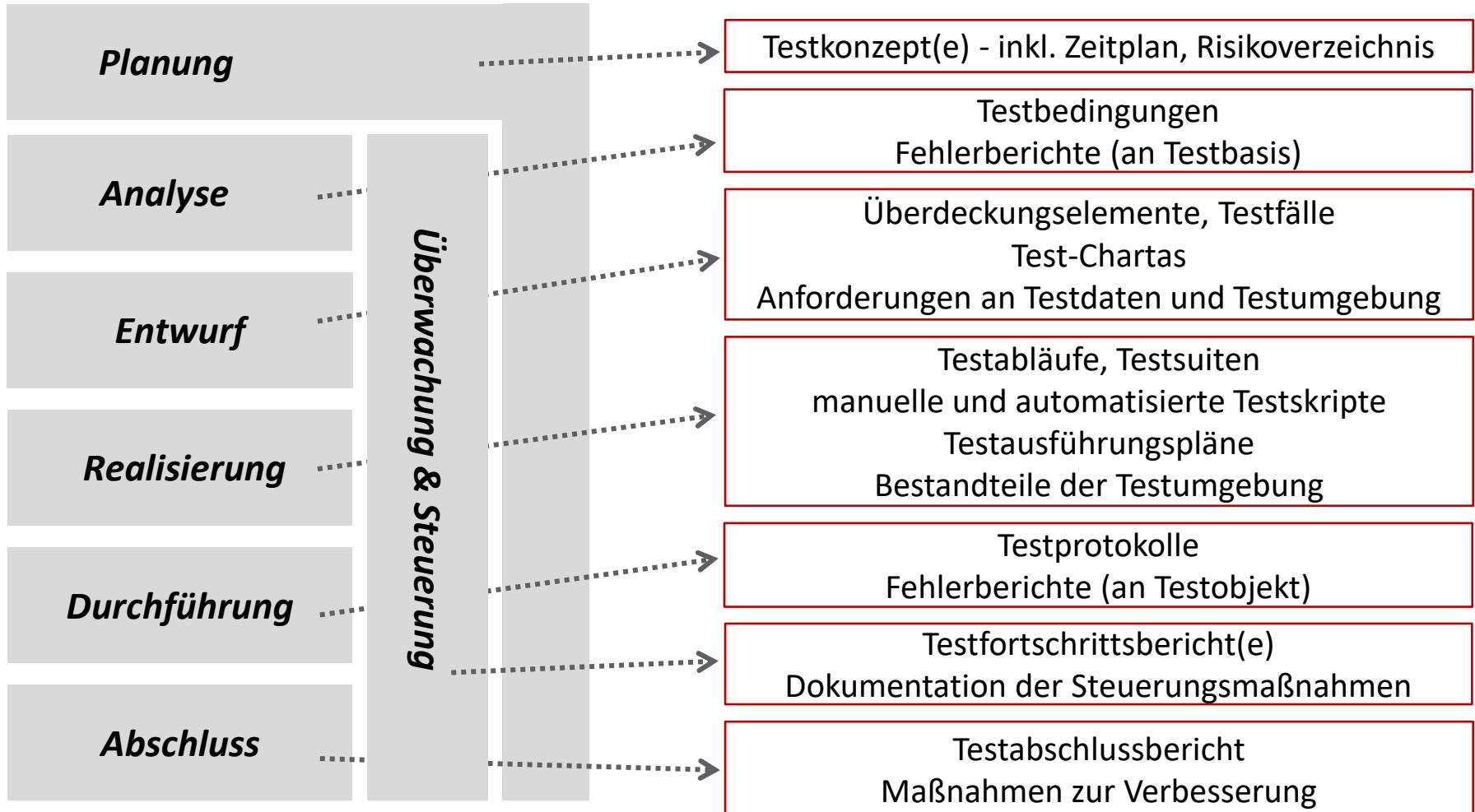
- Stakeholder
- Teammitglieder
- Unternehmensbereich
- technische Faktoren
- projektbedingte Randbedingungen
- organisatorische Faktoren
- Softwareentwicklungslebenszyklus
- Werkzeuge

Diese Faktoren wirken sich auf viele testbezogene Aspekte aus, darunter:

- Teststrategie
- verwendete Testverfahren
- Grad der Testautomatisierung
- geforderte Überdeckung
- Detaillierungsgrad der Testmittel
- Berichterstattung
- usw.

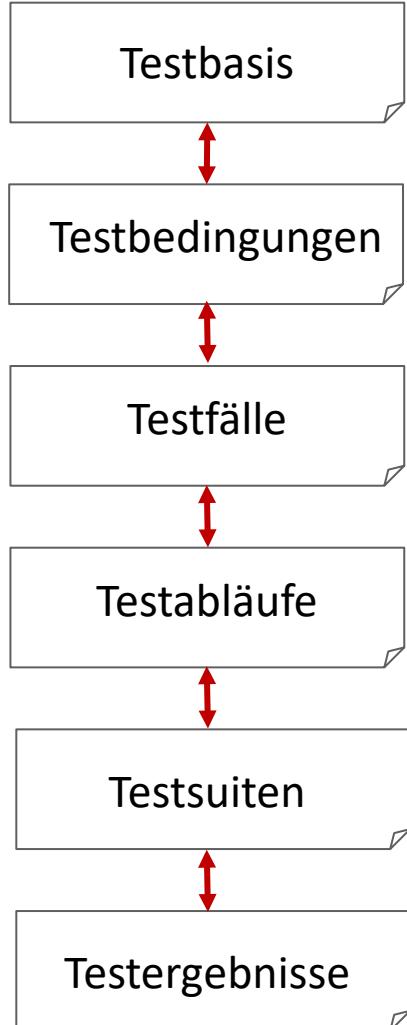


Testprozess – Testmittel





Verfolgbarkeit



- unterstützt die Bewertung der Überdeckung (z. B. von Anforderungen, Testbedingungen, Testfällen)
- ermöglicht Auswirkungsanalyse von Änderungen (Kapitel 2)
- Erfüllung von IT-Governance-Kriterien
- Liefern von Informationen zur Beurteilung von Produktqualität, Prozessfähigkeit und Projektfortschritt gegenüber Geschäftszielen
- verständliche Testberichte für Stakeholder



Rollen beim Testen

Tester

übernehmen die Gesamtverantwortung für

- operative Aspekte des Testens

Aktivitäten

- Testanalyse
- Testentwurfs
- Testrealisierung
- Testdurchführung

Testmanagement

übernimmt die Gesamtverantwortung für

- den Testprozess
- das Testteam
- die Leitung der Testaktivitäten

Aktivitäten

- Testplanung
- Testüberwachung und Teststeuerung
- Testabschluss

Wie die Rolle ausgeübt wird, variiert je nach Kontext.

1 – Grundlagen des Testens

1. Was ist Testen?
2. Warum ist Testen notwendig?
3. Testaktivitäten, Testmittel und Rollen des Testens
4. Wesentliche Kompetenzen und bewährte Praktiken beim Testen



1.5 Wesentliche Kompetenzen und bewährte Praktiken beim Testen

- FL-1.5.1 (K2) Sie können Beispiele für die allgemeinen Kompetenzen, die für das Testen erforderlich sind, geben
- FL-1.5.2 (K1) Sie können die Vorteile des Whole-Team-Ansatzes wiedergeben
- FL-1.5.3 (K2) Sie können die Vor- und Nachteile des unabhängigen Testens unterscheiden

Wesentliche Kompetenzen und bewährte Praktiken beim Testen

Die folgenden Kompetenzen sind allgemeiner Art, aber für Tester besonders wichtig



Gründlichkeit,
Sorgfalt

Neugier

Detailgenauigkeit

methodisches
Vorgehen

aktives
Zuhören

Gute
Kommunikations-
fähigkeit

Technische
Kenntnisse



Wissen in der
Anwendungsdomäne

Analytisches und
kritisches Denken

Teamfähigkeit

Kreativität

Testwissen

Psychologische Aspekte

Es ist ein allgemeiner menschlicher Charakterzug, den Überbringer schlechter Nachrichten zu tadeln.

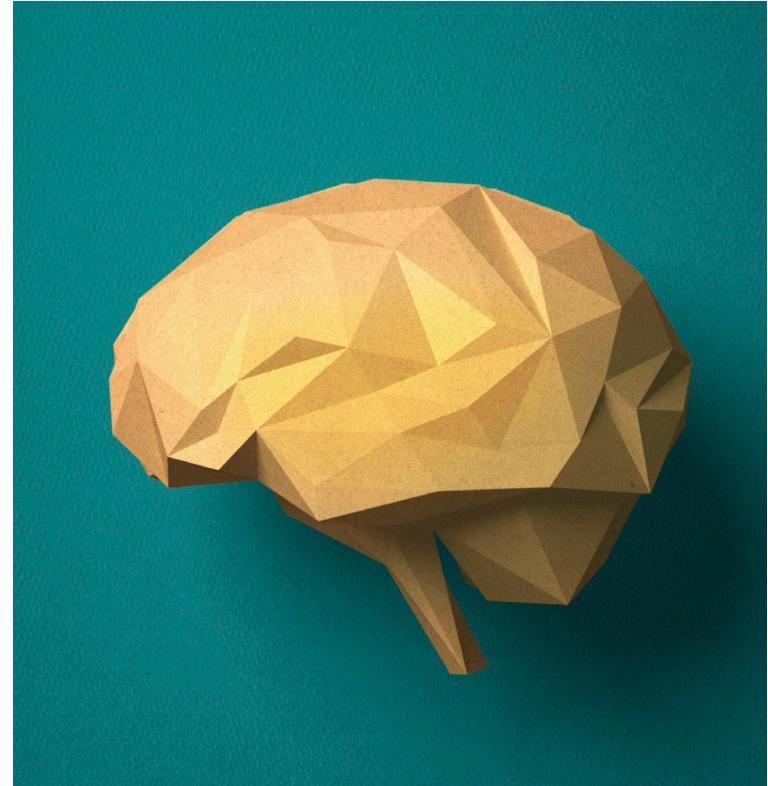
→ **Kommunikationsfähigkeit ist für Tester von entscheidender Bedeutung.**

Die Kommunikation von Testergebnissen kann als Kritik an dem Produkt und seinem Autor aufgefasst werden. Voreingenommenheit kann dazu führen, dass es schwierig ist, Informationen zu akzeptieren, die nicht mit den bereits bestehenden Überzeugungen übereinstimmen.

→ **Bestätigungsfehler***

Manche Menschen empfinden das Testen als eine destruktive Aktivität, obwohl es in hohem Maße zum Projekterfolg und zur Qualität des Produkts beiträgt.

→ **Fehler konstruktiv kommunizieren, um diese Sichtweise zu verbessern.**



* Der **Bestätigungsfehler** entsteht durch den direkten Einfluss des Wunsches auf die Überzeugungen. Wenn Personen sich wünschen, dass eine bestimmte Idee/ein bestimmtes Konzept wahr ist, glauben sie schließlich, dass sie wahr ist. Sie werden durch Wunschdenken motiviert. Dieser Fehler führt dazu, dass der Einzelne aufhört, Informationen zu sammeln, wenn die bisher gesammelten Beweise die Ansichten (Vorurteile) bestätigen, die man gerne wahr haben möchte.

[Quelle: <https://www.psychologytoday.com/us/blog/science-choice/201504/what-is-confirmation-bias>]



Gute Kommunikation

Beispiele für positive Ansätze

Zusammenarbeit statt Streit

Alle Beteiligten haben ein gemeinsames Ziel: „Ein System von höherer Qualität“.

Testen ist sinnvoll

Betone den Nutzen: Informationen über Fehler helfen, Arbeitsergebnisse und Fähigkeiten zu verbessern. Das spart Zeit und Geld und reduziert das allgemeine Risiko.

Empathie

Versuche die Gefühle und die Gründe zu verstehen, wenn eine andere Person negativ auf Informationen reagiert.

Neutrale und faktenorientierte Kommunikation

Kommuniziere Testergebnisse und andere Erkenntnisse konstruktiv, ohne zu kritisieren. Schreibe objektive und tatsächenbasierte Fehlerberichte und Review-Befunde.

Die richtigen Testziele definieren

Die meisten Menschen neigen dazu, ihre Pläne und Verhaltensweisen an den gesetzten Zielen auszurichten.



Whole-Team-Ansatz



Whole-Team-Ansatz

Whole-Team-Ansatz = Alle Teammitglieder sind gemeinsam für den Erfolg des Projekts verantwortlich.

Interdisziplinär

Das Team ist selbstorganisiert.
Jeder bringt sein Fachwissen ein.

Zusammenarbeit

Jeder Tester arbeitet effektiv mit anderen Teammitgliedern zusammen.
Jeder trägt positiv zu den Zielen des Teams bei.

Verantwortung für Qualität

Alle Mitglieder, die die nötigen Fähigkeiten haben, können jede Aufgabe übernehmen.
Jeder ist für die Qualität des Produkts verantwortlich.

Gemeinsamer Arbeitsbereich

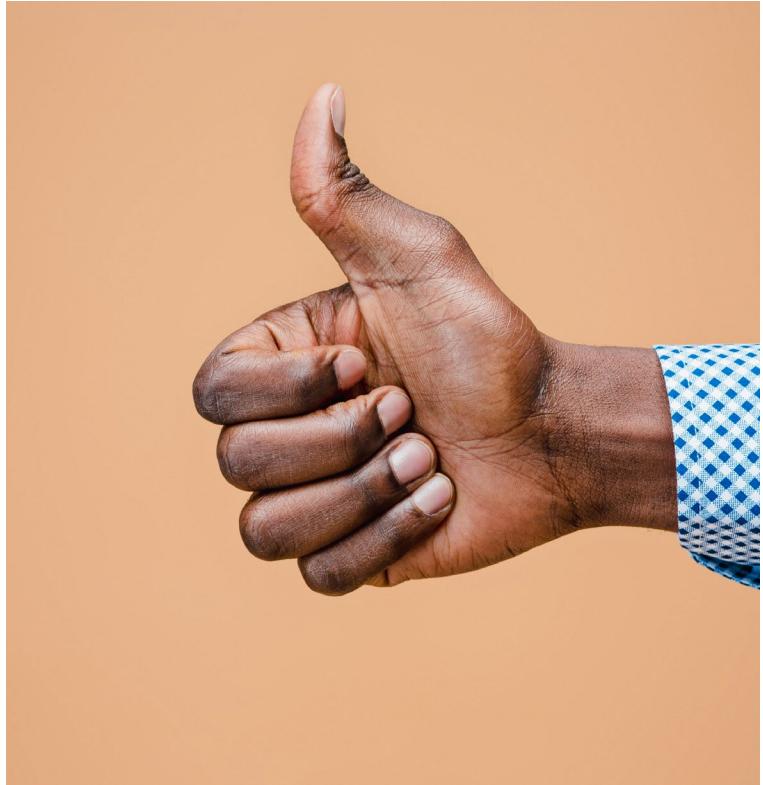
Das Team arbeitet in einem gemeinsamen Arbeitsbereich (physisch oder virtuell), um die Kommunikation und Zusammenarbeit zu verbessern.

i Ausnahme: In sicherheitskritischen Bereichen kann es notwendig sein, dass Tests unabhängig durchgeführt werden. In solchen Fällen ist der Whole-Team-Ansatz nicht geeignet.

Whole-Team-Ansatz – Vorteile

Vorteile des Whole-Team-Ansatzes

- verbessert die Teamdynamik
- fördert die Kommunikation und Zusammenarbeit innerhalb des Teams
- schafft Synergien,
da die verschiedenen Kompetenzen innerhalb des Teams zum Nutzen des Projekts eingesetzt werden können



Whole-Team-Ansatz – Tester

Tester ...

- arbeiten eng mit anderen Teammitgliedern zusammen, um sicherzustellen, dass die gewünschte Qualität erreicht wird.
- arbeiten mit **Fachbereichsvertretern** zusammen, um sie bei der Erstellung **geeigneter Abnahmetests** zu unterstützen.
- arbeiten mit **Entwicklern** zusammen, um die Teststrategie abzustimmen und über **Ansätze der Testautomatisierung** zu entscheiden.
- geben ihr Wissen über das Testen an andere Teammitglieder weiter.
- beeinflussen die positive Entwicklung des Produkts.

Unabhängigkeit des Testens

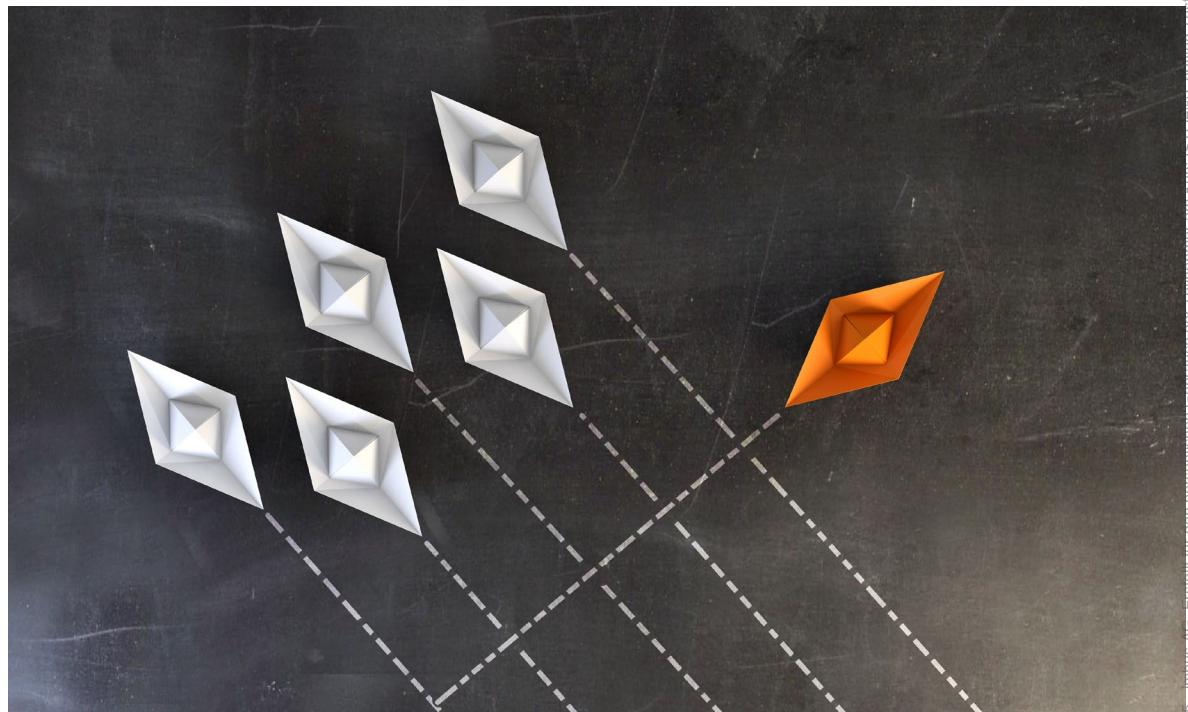
Unabhängigkeit des Testens – Personal

Ein gewisser Grad an Unabhängigkeit macht die Tester aufgrund von Unterschieden zwischen den kognitiven Verzerrungen der Autoren und der Tester effektiver bei der Fehlerfindung.

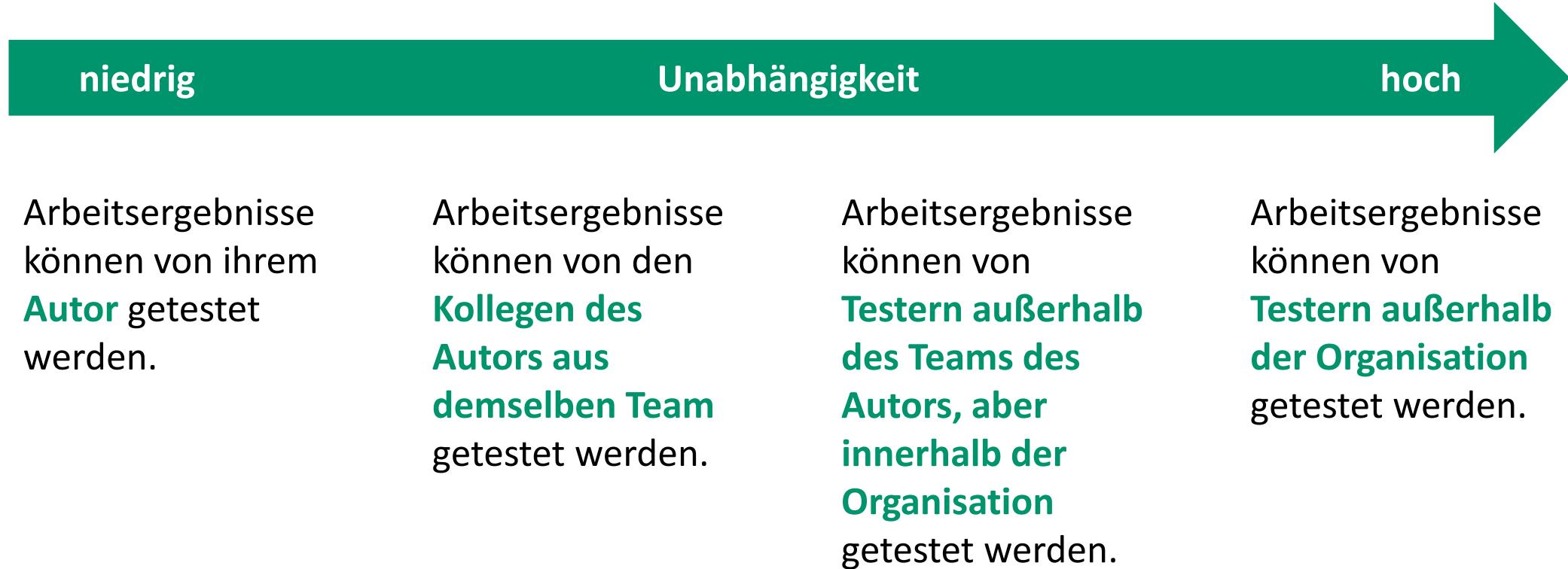
Unabhängigkeit ist jedoch kein Ersatz für Nähe zum System, z. B. können Entwickler viele Fehlerzustände in ihrem eigenen Code effizient finden.

Beispiele für kognitive Verzerrungen:

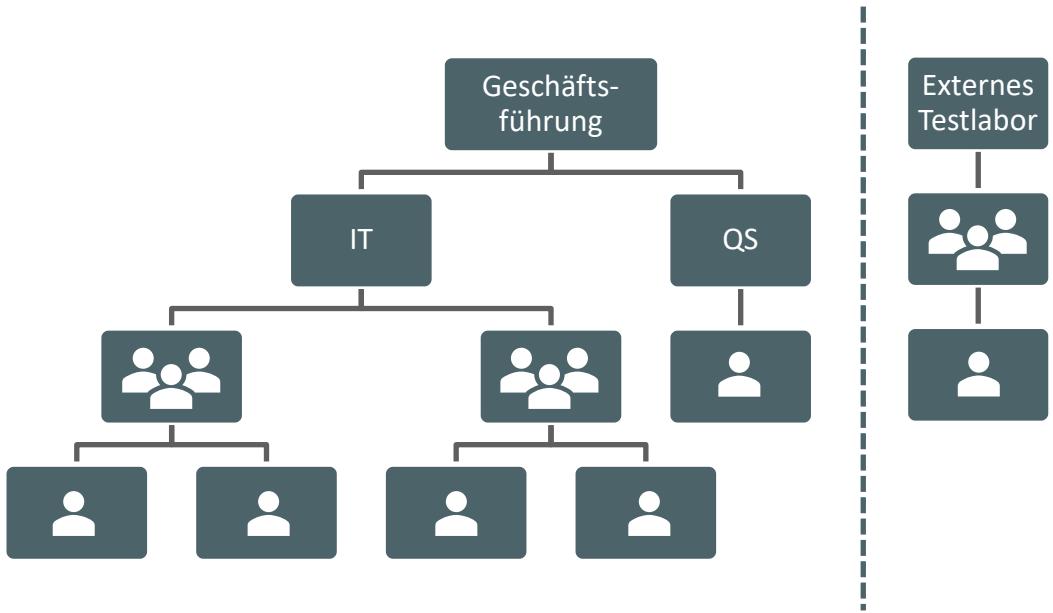
- Bestätigungsfehler
- Verfügbarkeitsheuristik
- Ankereffekt



Grad an Unabhängigkeit



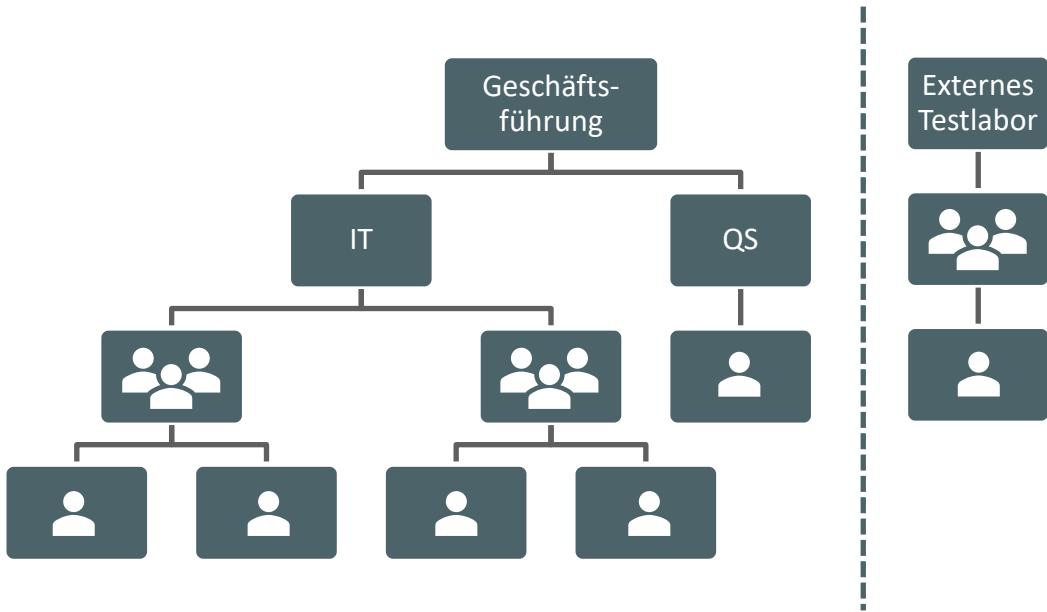
Unabhängiges Testen – Mögliche Vorteile



Unabhängige Tester ...

- ... erkennen wahrscheinlich andere Arten von Fehlerwirkungen und Fehlerzuständen als Entwickler. Mögliche Gründe:
 - unterschiedlicher Hintergrund
 - technische Perspektive
 - Bestätigungsfehler der Entwickler
- ... können die Annahmen, die von den Stakeholdern während der Spezifikation und Implementierung des Systems gemacht wurden,
 - überprüfen
 - in Frage stellen
 - widerlegen

Unabhängiges Testen – Mögliche Nachteile



Unabhängige Tester können vom Entwicklungsteam isoliert sein, was zu Problemen führen kann:

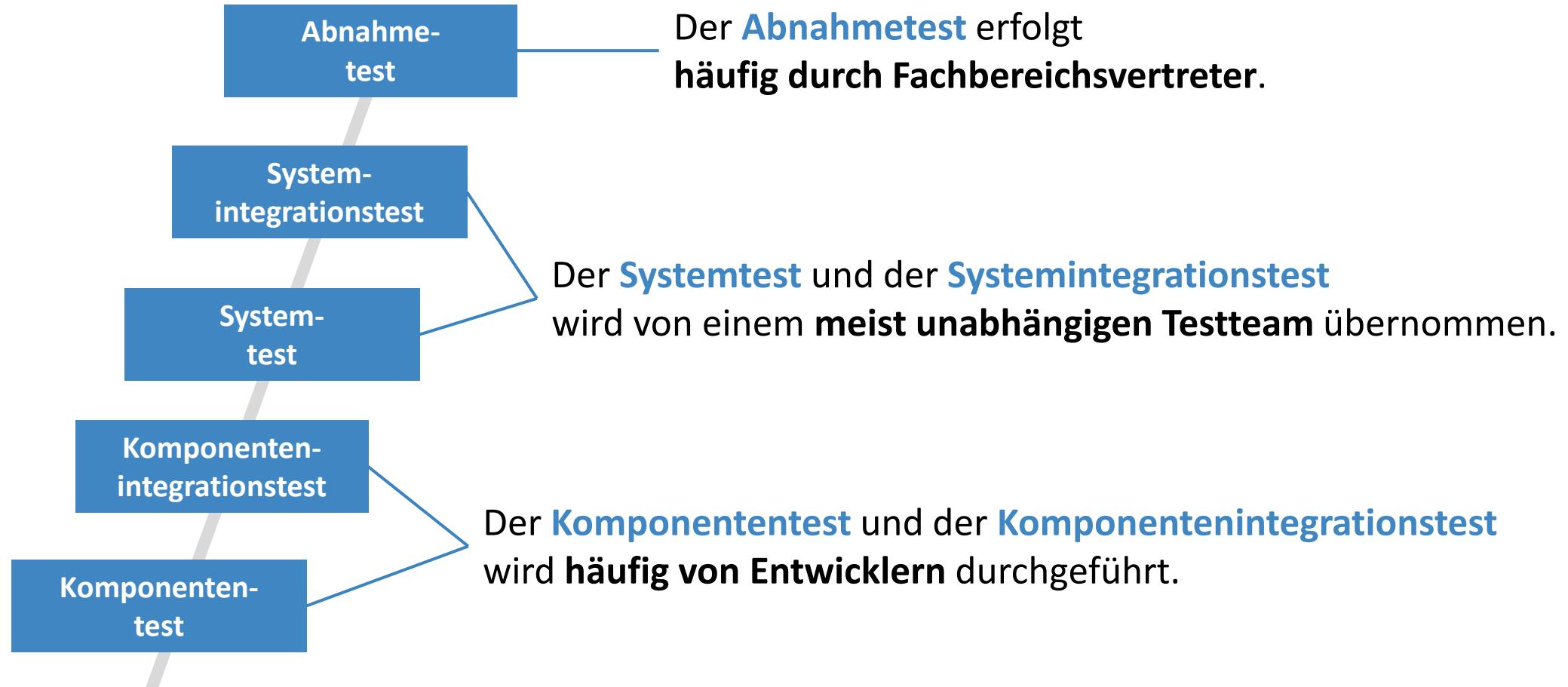
- mangelnde Zusammenarbeit
- Kommunikationsprobleme
- gegnerische Beziehung mit dem Entwicklungsteam
- Entwickler verlieren möglicherweise das Gefühl der Verantwortung für die Qualität
- Unabhängige Tester können als Engpass angesehen oder für Verzögerungen bei der Freigabe verantwortlich gemacht werden

Es ist möglich, die Vorteile der Unabhängigkeit zu nutzen und die Nachteile zu vermeiden.

Unabhängigkeitsstufen



Bei den meisten Projekten ist es sinnvoll, das Testen mit mehreren Unabhängigkeitsstufen durchzuführen.





Testen – erforderliche Kompetenzen

Für das Testen erforderliche Kompetenzen

- Testwissen
- Gründlichkeit, Sorgfalt, Neugier, Detailgenauigkeit, methodisches Vorgehen
- gute Kommunikationsfähigkeit, aktives Zuhören, Teamfähigkeit
- analytisches Denken, kritisches Denken, Kreativität
- technische Kenntnisse
- Wissen in der Anwendungsdomäne



Testen – Whole-Team Ansatz

Vorteile

- verbessert die Teamdynamik
- fördert die Kommunikation und Zusammenarbeit innerhalb des Teams
- schafft Synergien, da die verschiedenen Kompetenzen innerhalb des Teams zum Nutzen des Projekts eingesetzt werden können



Unabhängiges Testen

Vorteile

- erkennen andere Arten von Fehlerwirkungen und Fehlerzuständen
- unterschiedliche Hintergründe und technische Perspektiven
- Verifizieren von Annahmen anderer

Nachteile

- Isolation des Tests
- mangelnde Zusammenarbeit
- Kommunikationsprobleme
- gegnerische Beziehung mit dem Entwicklungsteam
- Gefahr, dass Entwickler das Verantwortungsgefühl für Qualität verlieren
- Testen wird als Engpass angesehen oder für Verzögerungen bei der Freigabe verantwortlich gemacht