



GTB

German Testing Board

Software. Testing. Excellence.



Basiswissen Softwaretest Certified Tester

Grundlagen des Softwaretestens

HS@GTB
2019
Version 3.1



Nach diesem Kapitel (Vorlesung) sollten Sie ...

- Begriffe kennen, anhand von Beispielen erklären und vergleichen können: Mangel, Fehler, Fehlerwirkung, Fehlerzustand/Defekt und Fehlhandlung
- Die (Haupt-)Qualitätsmerkmale nach dem ISO Qualitätsmodell kennen
- Die allgemeinen Zielsetzungen und Grundsätze des Testens kennen
- Beschreiben können, warum Testen Teil der Qualitätssicherung ist
- Beispiele angeben können, wie Testen zu einer höheren Qualität beiträgt
- Zwischen Testen und Debugging unterscheiden können
- Den Testprozess und seine Arbeitsergebnisse erklären können
- Bedeutung der Pflege der Rückverfolgbarkeit zwischen Testbasis und Testergebnissen erklären können
- Die psychologischen Probleme beim Testen kennen und charakterisieren können

Lernziele für den Abschnitt Grundlagen des Softwaretestens

(nach Certified Tester Foundation Level Syllabus,
deutschsprachige Ausgabe, Version 2018)



Lernziele für die Grundlagen des Testens (1 von 2)

- 1.1 Was ist Testen?
 - FL-1.1.1 (K1) Typische Ziele des Testens identifizieren können
 - FL-1.1.2 (K2) Testen von Debugging unterscheiden können
- 1.2 Warum ist Testen notwendig? (K2)
 - FL-1.2.1 (K2) Beispiele dafür geben können, warum Testen notwendig ist
 - FL-1.2.2 (K2) Die Beziehung zwischen Testen und Qualitätssicherung beschreiben können und Beispiele dafür geben können, wie Testen zu höherer Qualität beiträgt
 - FL-1.2.3 (K2) Zwischen Fehlhandlung, Fehlerzustand und Fehlerwirkung unterscheiden können
 - FL-1.2.4 (K2) Zwischen der Grundursache eines Fehlerzustands und seinen Auswirkungen unterscheiden können.

Lernziele für den Abschnitt Grundlagen des Softwaretestens

(nach Certified Tester Foundation Level Syllabus,
deutschsprachige Ausgabe, Version 2018)



Lernziele für die Grundlagen des Testens (2 von 2)

- 1.3 Sieben Grundsätze des Testens
 - FL-1.3.1 (K2) Die sieben Grundsätze des Testens erklären können
- 1.4 Testprozess
 - FL-1.4.1 (K2) Die Auswirkungen des Kontexts auf den Testprozess erklären können
 - FL-1.4.2 (K2) Die Testaktivitäten und zugehörigen Aufgaben innerhalb des Testprozesses beschreiben können
 - FL-1.4.3 (K2) Arbeitsergebnisse unterscheiden können, die den Testprozess unterstützen
 - FL-1.4.4 (K2) Die Bedeutung der Pflege der Rückverfolgbarkeit zwischen Testbasis und Testarbeitsergebnissen erklären können.
- 1.5 Die Psychologie des Testens
 - FL-1.5.1 (K1) Die psychologischen Faktoren identifizieren können, die den Erfolg des Testens beeinflussen
 - FL-1.5.2 (K2) Den Unterschied zwischen der für Testaktivitäten erforderlichen Denkweise und der für Entwicklungsaktivitäten erforderlichen Denkweise erklären können.

Kapitel 1

Grundlagen des Softwaretestens

Begriffe und Motivation

Grundsätze des Softwaretestens

Testprozess

Testfälle, Erwartete Ergebnisse und Testorakel

Psychologie des Testens

Was gilt als Fehler oder Mangel?

- Situation fehlerhaft? → vorab die erwartete Situation festlegen
- **Fehler:**
 - Nichterfüllung einer festgelegten Anforderung
 - Abweichung zwischen Istergebnis und erwartetem Ergebnis
- **Mangel:**
 - Anforderung / berechnigte Erwartung nicht angemessen erfüllt
 - Nichtkonformität bzgl. beabsichtigtem oder festgelegtem Gebrauch



Ursachenkette für Fehler

- **Fehler/Mangel**
 - seit Fertigstellung der Software vorhanden,
 - kommt erst bei Ausführung der Software zum Tragen.
- **Fehlerwirkung** (*failure*): Das sichtbare Auftreten des Fehlers
- **Fehlerzustand** (*defect*): Ursache einer Fehlerwirkung
- **Fehlhandlung** (*error*) einer Person: Ursache eines Fehlerzustands
- **Fehlermaskierung**
 - ein Fehlerzustand verhindert die Aufdeckung eines anderen

Was soll sichergestellt werden?

Fehlerwirkung (Failure)

Nach außen sichtbare Fehlverhalten



Fehlerzustand (Bug)

Zustand in der Anwendung der zu einer Fehlerwirkung führen kann.



Fehlhandlung (Error, Mistake)

Irrtum bei der Software-Entwicklung.

Fehlermaskierung

Wirkung eines Fehlers (Fehlerzustand) ist nach außen nicht sichtbar, weil er durch einen weiteren Fehler verborgen (überlagert / maskiert) wird.



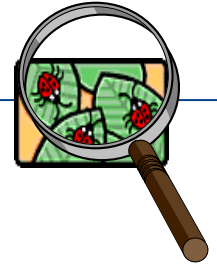
Fehlerwirkung - Definition

Fehlerwirkung (*failure*)

- 1. Ein Ereignis in welchem eine Komponente oder ein System eine geforderte Funktion nicht im spezifizierten Rahmen ausführt.**
[ISO 24765, Glossar 3.2]
2. Wirkung eines Fehlerzustands, die nach »außen« sichtbar auftritt.
3. Abweichung zwischen erwartetem Ergebnis und Istergebnis.
4. Abweichung eines Systems von der erwarteten Lieferung/Leistung.

- Anmerkungen zu Fehlerwirkung:
 - (selten) auch durch Höhenstrahlung, elektromagnetische Felder oder Hardwarefehler hervorgerufen.
 - Kann falscher Ausgabe, zu langsamer Ausführung oder Abbruch führen.

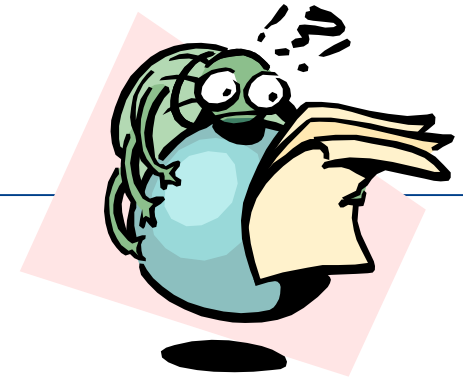
Fehlerzustand - Definition



Fehlerzustand (*defect*)

1. **Eine Unzulänglichkeit oder ein Mangel in einem Arbeitsergebnis, sodass es seine Anforderungen oder Spezifikationen nicht erfüllt [Glossar V3.2]**
2. Inkorrektes Teilprogramm
 - inkorrekte Anweisung
 - Inkorrekte Datendefinition
3. Zustand eines Programms, der ggf. (z.B. hohe Last) eine geforderte Funktion beeinträchtigt oder zu Fehlerwirkung führt

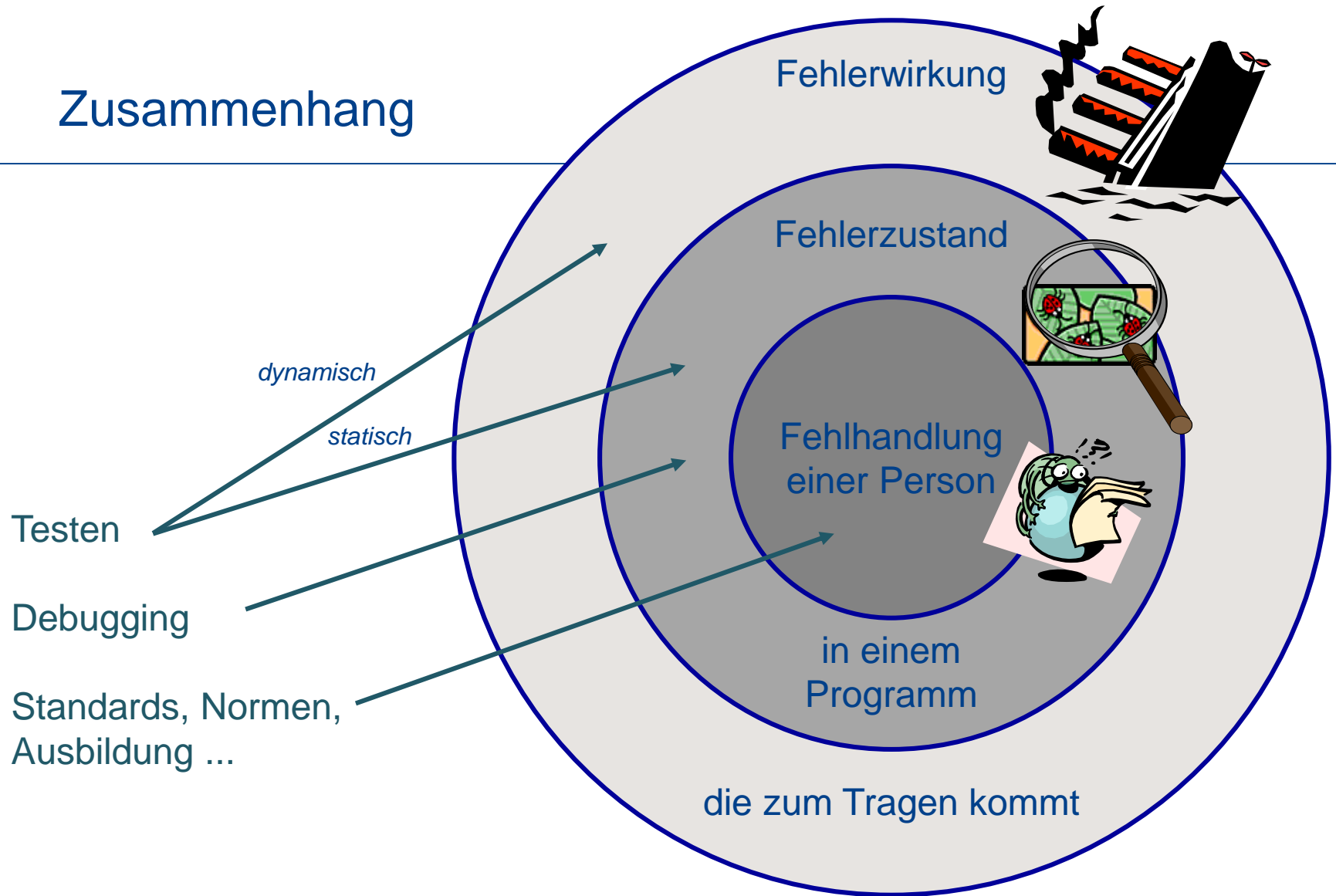
Fehlhandlung - Definition



Fehlhandlung (*error*)

1. **Die menschliche Handlung, die zu einem falschen Ergebnis führt.** [IEEE 610, Glossar 3.2]
2. Die menschliche Handlung
 - Entwickler: führt zu Fehlerzustand in der Software
 - Anwenders: Fehlbedienung mit unerwünschtem Ergebnis
3. Unwissentlich, versehentlich oder absichtlich ausgeführte Handlung oder Unterlassung, die ggf. zu einer Beeinträchtigung der Softwarefunktion führt

Zusammenhang



Testen – Definition (nach Glossar 3.2)

Der Prozess, der aus allen Aktivitäten des Lebenszyklus besteht (sowohl statisch als auch dynamisch), die sich mit

- der Planung,
- Vorbereitung und
- Bewertung

eines Softwareprodukts und dazugehöriger Arbeitsergebnisse befassen.



Ziel des Prozesses: Sicherstellen, dass

- diese allen festgelegten Anforderungen genügen,
- ihren Zweck erfüllen und
- etwaige Fehlerzustände finden.

Ziele des Testens

- Mögliche Ziele des Testens:
 - Arbeitsergebnisse bewerten: Anforderungen, User-Stories, Architekturdesign und Code
 - Vollständigkeit des Testobjekts prüfen
 - Fehlerzustände vermeiden
 - Die Risikostufe mangelhafter Softwarequalität reduzieren (z.B. für unentdeckte Fehlerwirkungen)
 - Validieren des Testobjekts gegen Erwartungen der Stakeholder
 - Den Stakeholdern Informationen für fundierte Entscheidungen geben, insbesondere bezüglich des Qualitätsniveaus des Testobjekts
 - Vertrauen in das Qualitätsniveau der Software schaffen



Validierung und Verifizierung - Definitionen

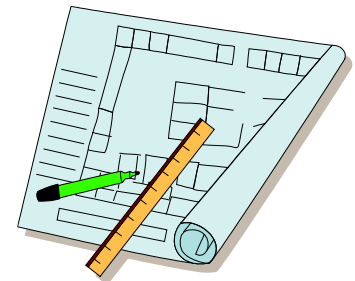
Validierung

- Bestätigung durch Bereitstellung eines objektiven Nachweises, dass die Anforderungen für einen spezifischen beabsichtigten Gebrauch oder eine spezifische beabsichtigte Anwendung erfüllt worden sind. [ISO 9000, Glossar 3.2]
- Haben wir das **richtige System** realisiert?

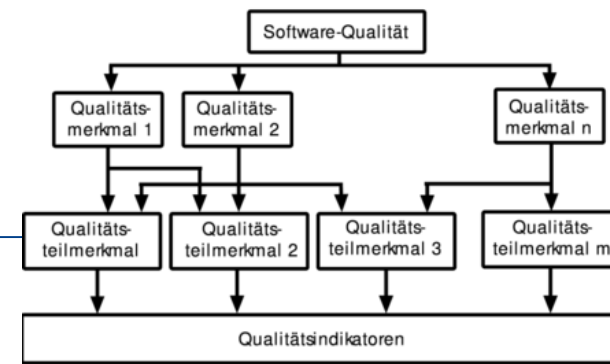


Verifizierung

- Bestätigung durch Bereitstellung eines objektiven Nachweises, dass festgelegte Anforderungen erfüllt worden sind. [ISO 9000, Glossar 3.2]
- Haben wir das **System richtig** realisiert?



Was ist Softwarequalität?



Qualität:

- Der Grad, in dem ein System, eine Komponente oder ein Prozess die Kundenerwartungen und -bedürfnisse erfüllt. [IEEE 610, Glossar 3.2]
- Der Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt. [ISO 9000:2000]

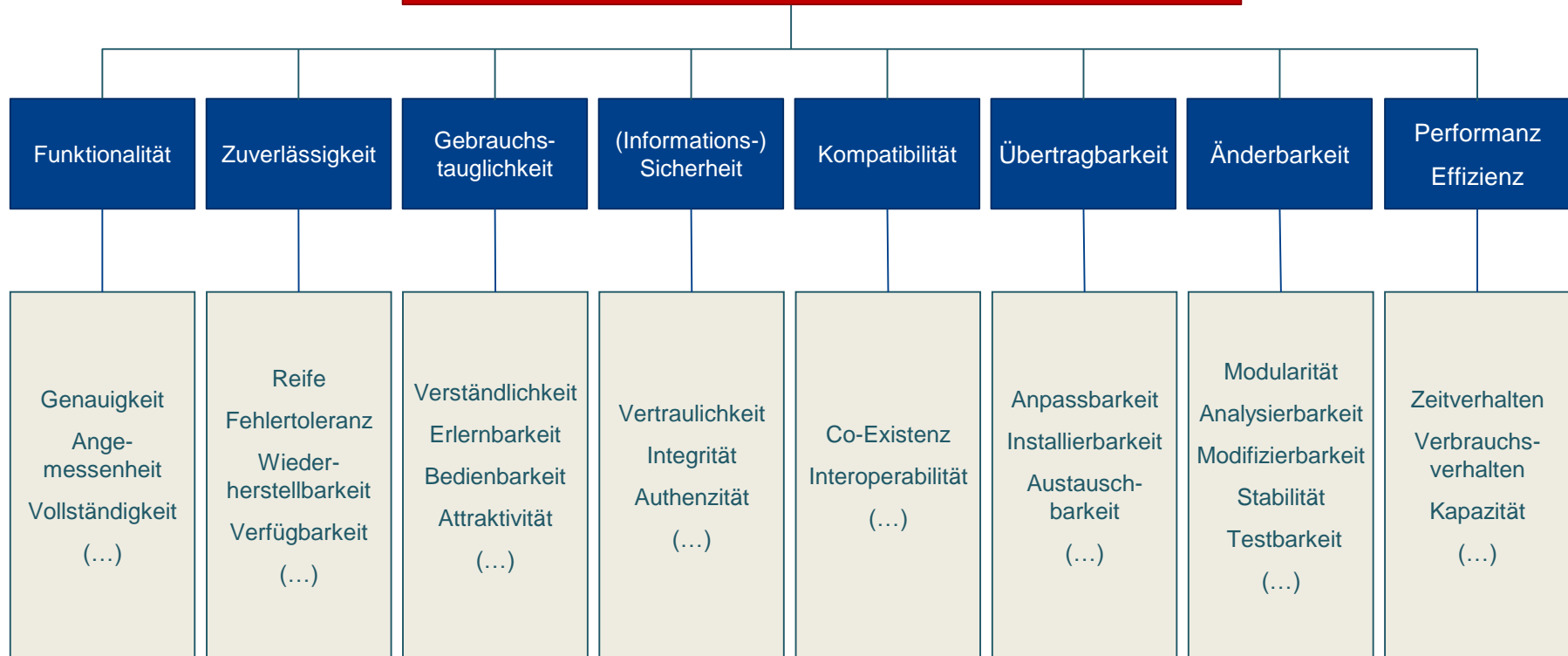
Softwarequalität:

- Gesamtheit der Funktionalitäten und Merkmale eines Softwareprodukts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen. [Glossar 3.2]
- Qualitätsmerkmale beziehen sich auf Anforderungen
 - Funktionale Anforderungen
(Fachlichkeit, Funktionen, Schnittstellen, ...)
 - Nicht-Funktionale Anforderungen
(Qualitäts- und Realisierungsanforderungen, Projektspezifische Anforderungen, ...)



Qualität nach ISO 25010

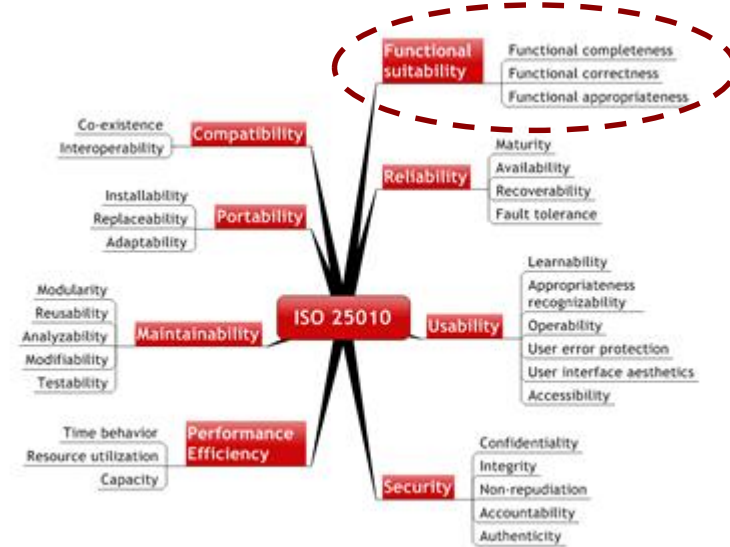
Produkt-Qualitätsmerkmale nach ISO 25010



ISO 25010: Software-Engineering - Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) - Qualitätsmodell und Leitlinien

Qualitätsmerkmal Funktionalität

Vorhandensein von Funktionen,
die festgelegte Anforderungen erfüllen



Angemessenheit

- Die Fähigkeit eines Softwareprodukts für spezifizierte Aufgaben und Zielsetzungen der Benutzer einen geeigneten Satz Funktionen zu liefern. [Glossar 3.2]

Genauigkeit

- Die Fähigkeit eines Softwareprodukts, die richtigen oder vereinbarten Ergebnisse oder Wirkungen mit dem benötigten Grad an Genauigkeit zu liefern. [Glossar 3.2]

Qualitätsmerkmal Zuverlässigkeit

Der Grad, zu dem spezifizierte Funktionen ausgeführt werden [ISO 25010]

- unter den festgelegten Bedingungen
- während einer bestimmten Zeitspanne



Reife

- System erfüllt die Anforderungen an die Zuverlässigkeit im Normalbetrieb (nach [ISO 25010, Glossar 3.2])

Fehlertoleranz

- System erhält ein spezifiziertes Leistungsniveau aufrecht ([ISO 9126], Glossar 3.2)
 - auch bei Fehlfunktionen
 - oder trotz Fehleingaben (z.B. falsche Bedienung)

Wiederherstellbarkeit

- System stellt bei einer Fehlerwirkung das spezifizierte Leistungsniveau wieder her und gewinnt die direkt betroffenen Daten wieder (nach [ISO 9126, Glossar 3.2])

Qualitätsmerkmal Gebrauchstauglichkeit

Der Grad, zu dem ein System genutzt werden kann
(nach [Glossar 3.2])

- durch bestimmte Benutzer,
- in einem bestimmten Nutzungskontext,
- um festgelegte Ziele effektiv, effizient und zufriedenstellend zu erreichen.



Verständlichkeit

- Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, das Konzept und die Anwendung zu verstehen.

Erlernbarkeit

- Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, die Anwendung zu erlernen.

Bedienbarkeit

- Merkmale von Software, die sich auf den Aufwand für den Benutzer beziehen, der bei der Bedienung und Ablaufsteuerung entsteht.

Qualitätsmerkmal (Informations-)Sicherheit

Merkmale von Software, die sich auf ihre Eignung beziehen, unberechtigten Zugriff, sowohl versehentlich als auch vorsätzlich, auf Programme und Daten zu verhindern.



Vertraulichkeit

- Merkmale von Software, bestimmte Informationen nur einem eingeschränkten Empfängerkreis zugänglich zu machen

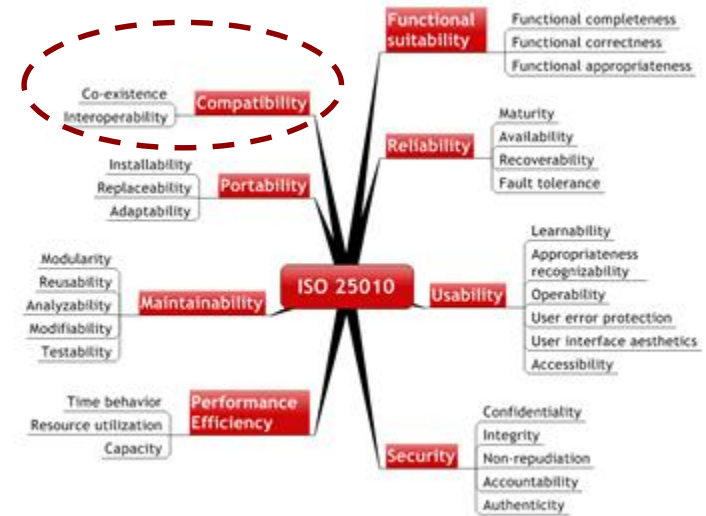
Integrität

- Merkmale von Software, alle Funktionen ohne unbeabsichtigte Nebenwirkungen durchzuführen

Authentizität

- Merkmale von Software, ihren Ursprung korrekten zu belegen (Signaturen, Prüfsummen)

Qualitätsmerkmal Kompatibilität

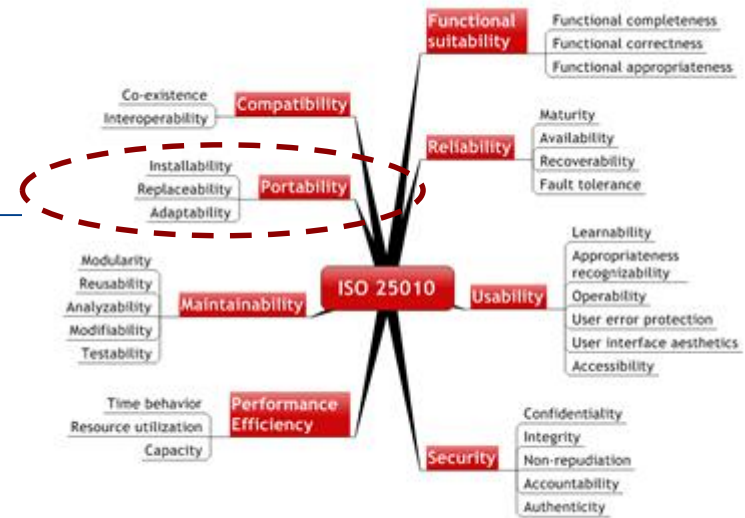


Interoperabilität

- Merkmale von Software, die sich auf ihre Eignung beziehen, mit vorgegebenen Systemen zusammenzuwirken.
- Die Fähigkeit eines Softwareprodukts, mit einer oder mehreren spezifizierten Komponenten zusammenzuwirken [nach ISO 9126].
- **Der Grad, zu dem zwei oder mehr Komponenten oder Systeme Informationen austauschen und diese nutzen können. ([nach ISO 25010], Glossar 3.2)**

Qualitätsmerkmal Übertragbarkeit (1 von 2)

Merkmale, die sich auf die Eignung der Software beziehen, von einer Umgebung in eine andere übertragen zu werden.



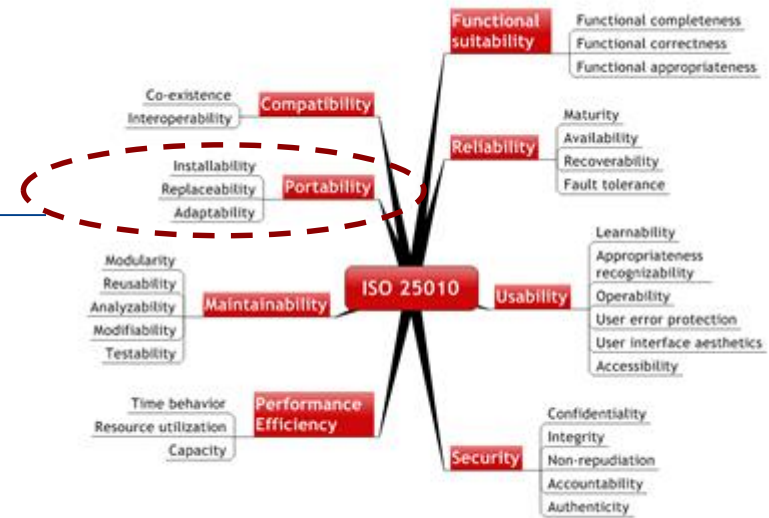
Anpassbarkeit

- Der Grad, zu dem eine Komponente oder ein System auf unterschiedliche oder sich weiterentwickelnde Hardware- und Softwareumgebungen angepasst werden kann. ([nach ISO 25010], Glossar 3.2)

Installierbarkeit

- Die Fähigkeit eines Softwareprodukts, in einer spezifizierten Umgebung installierbar zu sein ([ISO 9126]. Glossar 3.2)

Qualitätsmerkmal Übertragbarkeit (2 von 2)



Konformität

- Die Fähigkeit eines Softwareprodukts, anwendungsspezifische Normen oder Vereinbarungen oder gesetzliche Bestimmungen und ähnliche Vorschriften zu erfüllen. (Glossar 3.2)

Austauschbarkeit

- Die Fähigkeit eines Softwareprodukts, an Stelle einer anderen spezifizierten Software zum selben Zweck in der gleichen Umgebung genutzt zu werden. (Glossar 3.2)

Qualitätsmerkmal Änderbarkeit (1 von 2)

Merkmale, die sich auf den Aufwand beziehen, der zur Durchführung vorgegebener Änderungen notwendig ist.



Analysierbarkeit

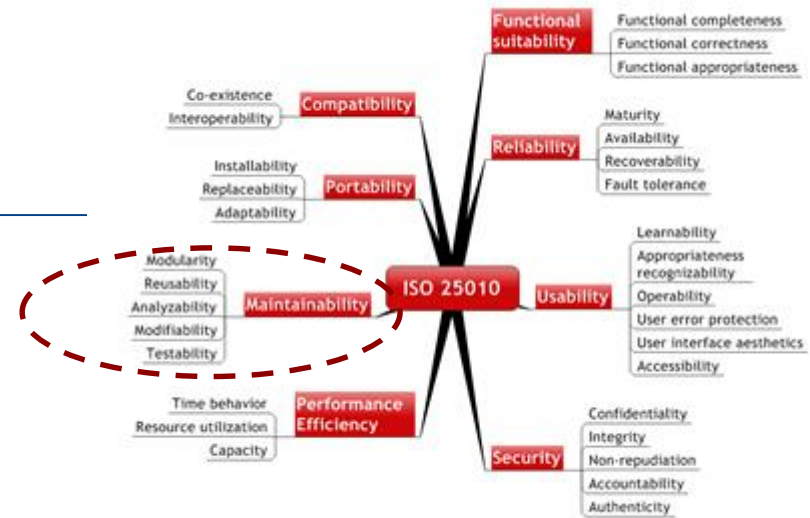
- Der Grad, zu dem für eine Komponente oder ein System die Auswirkungen beabsichtigter Änderungen bewertet, die Ursachen von Mängeln oder Fehlerwirkungen diagnostiziert, oder die zu ändernden Teile identifiziert werden können. ([nach ISO 25010], Glossar 3.2)

Modifizierbarkeit

- Die Fähigkeit eines Softwareprodukts, die Durchführung spezifizierter Änderungen zu ermöglichen ([ISO 9126], [Glossar 3.2])

Qualitätsmerkmal Änderbarkeit (2 von 2)

Merkmale, die sich auf den Aufwand beziehen, der zur Durchführung vorgegebener Änderungen notwendig ist.



Stabilität

- Der Grad, zu dem eine Komponente oder System effektiv und effizient geändert werden kann, ohne dabei Fehlerzustände einzubauen oder die Qualität zu mindern. ([ISO 25010], Glossar 3.2).

Testbarkeit

- Der Grad der Effektivität und Effizienz, zu dem Tests für eine Komponente oder ein System entworfen und durchgeführt werden können. ([nach ISO 25010], Glossar 3.2)

Qualitätsmerkmal Performanz/Effizienz



Performanz

- Der Grad, zu dem eine Komponente oder ein System Zeit,
- Ressourcen und Kapazität verbraucht während sie/es seine vorgesehenen Funktionen ausführt. ([nach ISO 25010], Glossar 3.2)

Effizienz

- Eingesetzte Mittel im Verhältnis zu dem Ausmaß, in dem Benutzer spezifische Ziele erreichen. ([DIN ISO 9241-11], Glossar 3.2)

Testen und Qualität

Testen misst die Qualität.

- z.B. anhand der Anzahl gefundener Fehlerwirkungen.

Testen erhöht indirekt die Qualität.

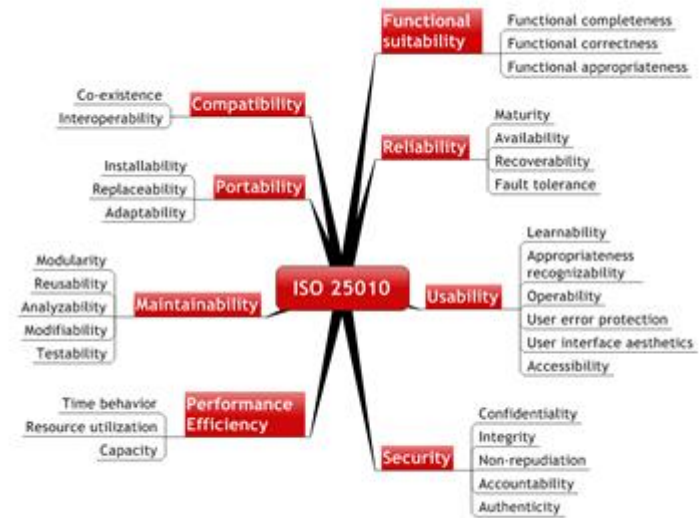
- da Fehler(zustände) vor der Auslieferung entdeckt und korrigiert werden können.

Testen erhöht indirekt die Prozessqualität.

- da Fehler dokumentiert, analysiert und damit Fehlhandlungen in Zukunft vermieden werden können.

Testen erhöht das Vertrauen in die Qualität des Systems.

- wenn wenige oder keine Fehler gefunden werden.



Qualitätsanforderungen

Qualitätsanforderungen geben vor, welche Qualitätsmerkmale das Produkt in welcher Güte aufweisen soll (Qualitätsniveau).

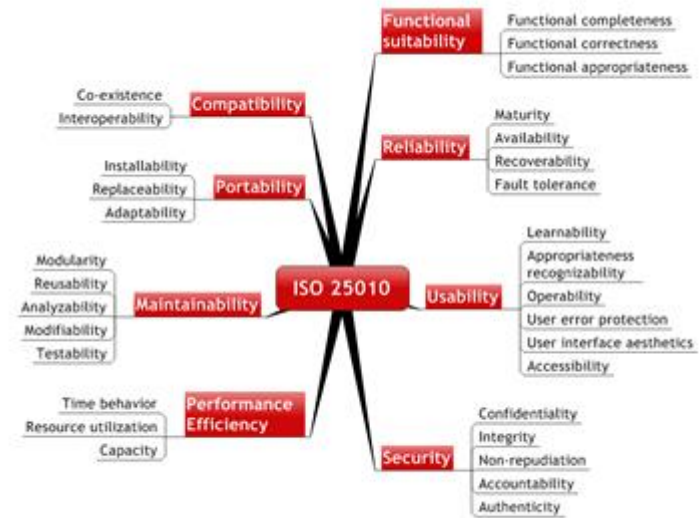
- Gesamtheit aller Qualitätsmerkmale und deren geforderte Ausprägung

Nicht alle Qualitätsmerkmale lassen sich gleich gut erfüllen.

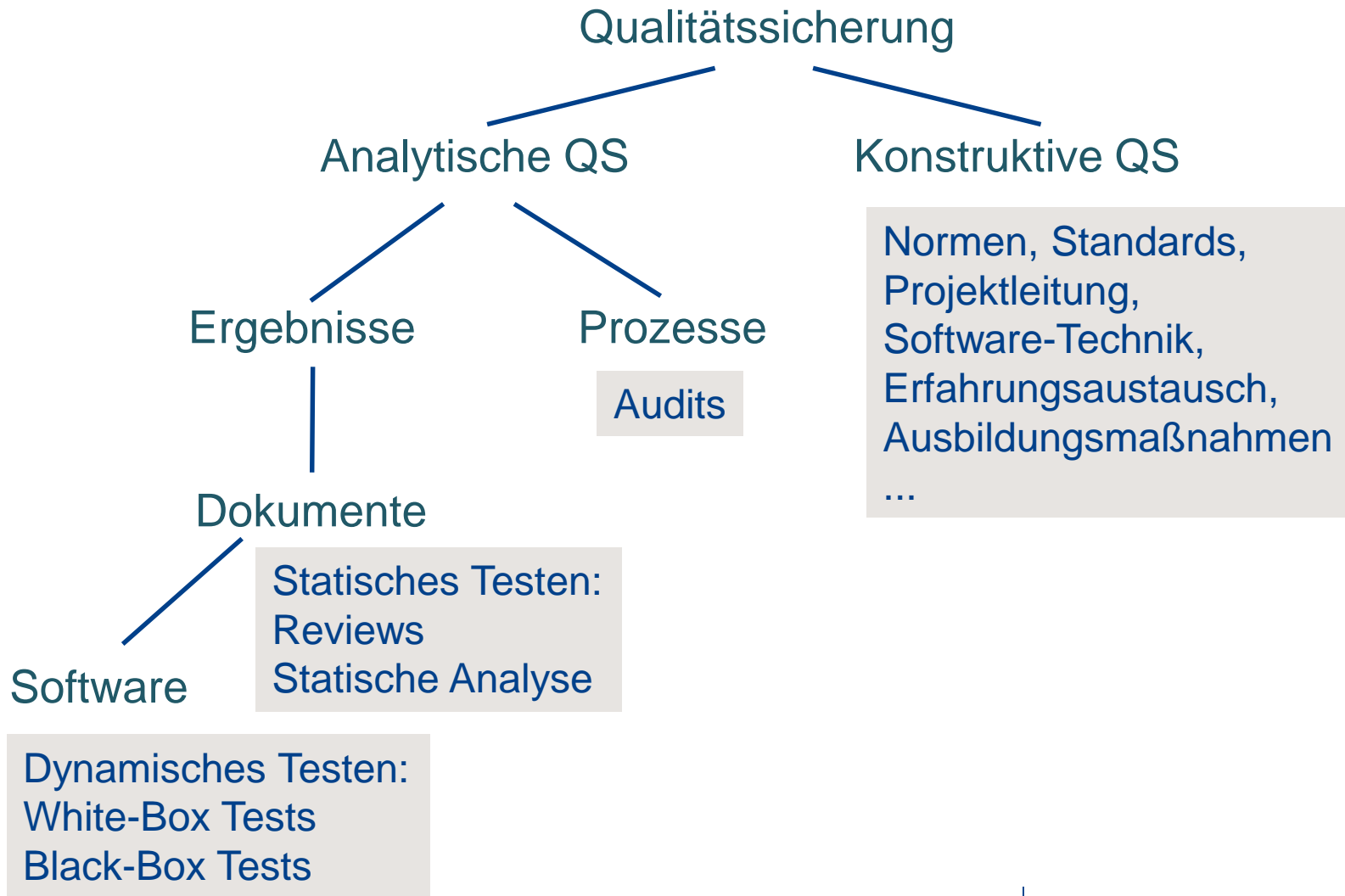
- Z.B. kann Effizienz zu Lasten der Wartbarkeit gehen.

Prioritäten festlegen!

- In engster Absprache mit Auftraggebern und Anwendern.
- Qualitätsanforderungen sind Bestandteil der nicht-funktionalen Anforderungen im Pflichtenheft.

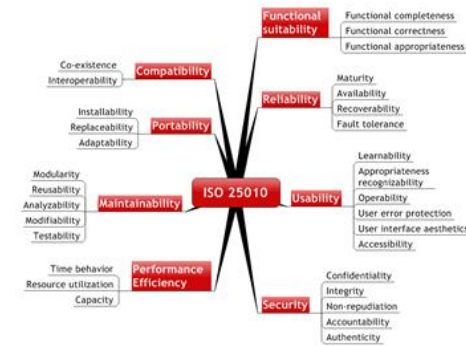


Qualitätssicherung - QS

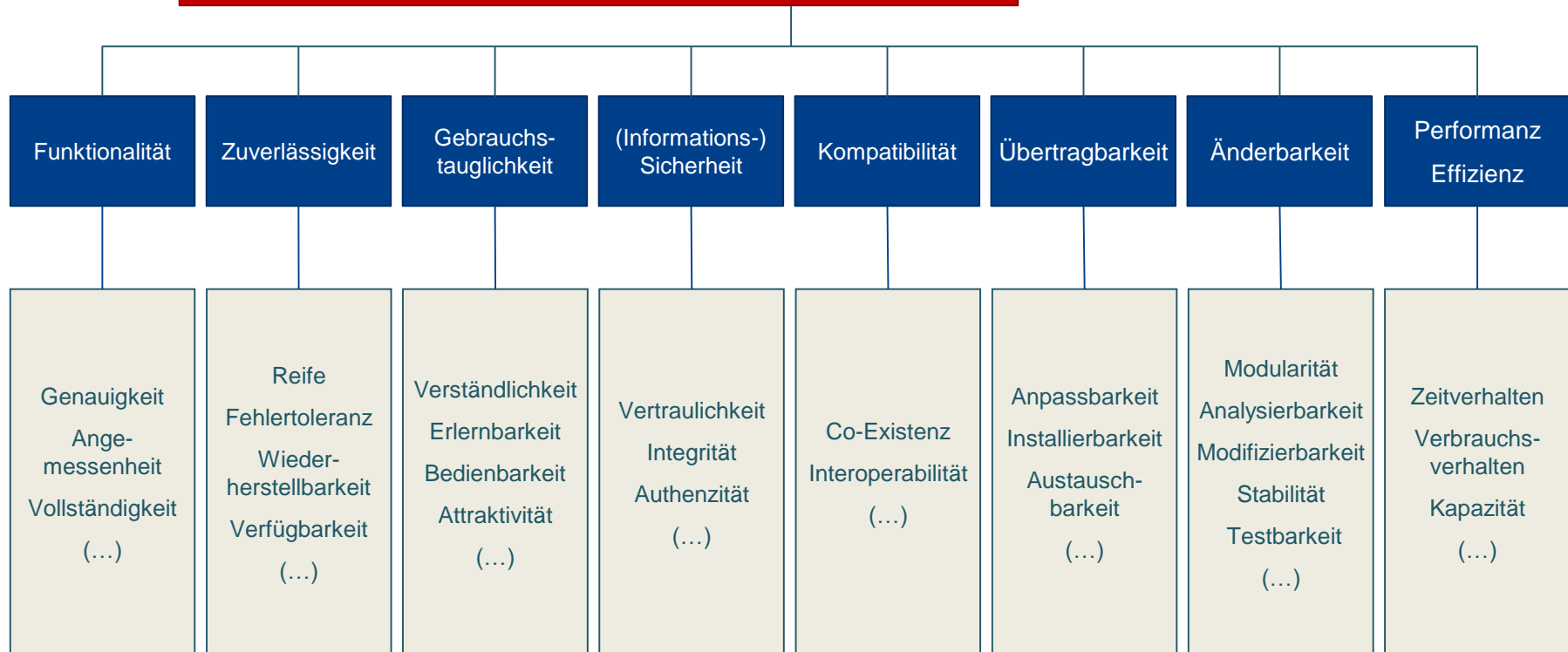




Qualität nach ISO 25010



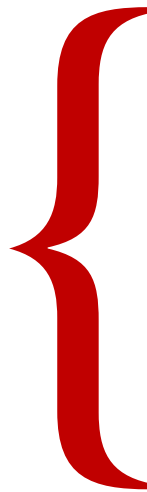
Produkt-Qualitätsmerkmale nach ISO 25010



ISO 25010: Software-Engineering - Qualitätskriterien und Bewertung von Softwareprodukten (SQuaRE) - Qualitätsmodell und Leitlinien

Kapitel 1

Grundlagen des Softwaretestens



Begriffe und Motivation

Grundsätze des Softwaretestens

Testprozess

Testfälle, erwartete Ergebnisse und Testorakel

Psychologie des Testens

Grundsätze zum Testen (im Überblick)

In den letzten 50 Jahren haben sich folgende Grundsätze zum Testen herauskristallisiert und können somit als Leitlinien dienen:

Grundsatz 1: »**Testen zeigt die Anwesenheit von Fehlerzuständen, nicht deren Abwesenheit**«

Grundsatz 2: »**Vollständiges Testen ist nicht möglich**«

Grundsatz 3: »**Frühes Testen spart Zeit und Geld**«

Grundsatz 4: »**Häufung von Fehlerzuständen**«

Grundsatz 5: »**Vorsicht vor dem Pestizid-Paradoxon**«

Grundsatz 6: »**Testen ist kontextabhängig**«

Grundsatz 7: »**Trugschluss: „Keine Fehler“ bedeutet ein brauchbares System**«



Testaufwand?

- »Testen ist ökonomisch sinnvoll, solange die Kosten für das Finden und Beseitigen eines Fehlers im Test niedriger sind als die Kosten, die mit dem Auftreten eines Fehlers bei der Nutzung verbunden sind.«

M. Pol, T. Koomen, A. Spillner:
Management und Optimierung des
Testprozesses. dpunkt-Verlag, 2. Auflage, 2002

- »Ein guter Test ist wie eine Haftpflichtversicherung: Er kostet richtig Geld, lässt aber den Projektleiter und den Kunden ruhig schlafen. Zum guten Schlaf gehört auch eine gute Versicherung, die alle möglichen Risiken abdeckt. Zum Vertrauen in die Software gehört ein guter Test, der die ganze Produktionswirklichkeit abdeckt.«

Siedersleben, J. (Hrsg): Softwaretechnik,
Praxiswissen für Softwareingenieure. 2. Auflage,
Hanser, 2003

Weiteres zum Testen (1)

- Testaufwand in der Praxis:
25% bis 50% des Entwicklungsaufwands
- Testintensität und -umfang in Abhängigkeit vom Risiko und der Kritikalität festlegen
- Fehler verursachen wirklich hohe Kosten
 - Geschätzte Verluste durch Softwarefehler in Mittelstands- und Großunternehmen in Deutschland ca. 84,4 Mrd. Euro p.a.
 - Produktivitätsverluste durch Computerausfälle aufgrund fehlerhafter Software ca. 2,6% des Umsatzes - 70 Mrd. Euro p.a.

Studie der LOT Consulting Karlsruhe
IT-Services 3/2001, S. 31

Weiteres zum Testen (2)

Testen unterliegt immer beschränkten Ressourcen (besonders Zeit, wenn Testen erst am Ende der Entwicklung in Angriff genommen wird)

Besonders wichtig:

- Adäquate (zum Testobjekt und den Qualitätszielen passende) Testverfahren auswählen
- Unnötige Tests (die keine neuen Erkenntnisse liefern) vermeiden
- Sowohl Positiv- als auch Negativtests berücksichtigen
- Auch Tests auf Funktionalität, die nicht gefordert (evtl. sogar absichtlich schädlich) sind, müssen einbezogen werden



Priorisierung der Tests

- Limitierte Ressourcen (Zeit und Personal) erfordern, dass die wichtigsten Testfälle zuerst durchgeführt werden!
- Kriterien für die Priorisierung:
 - Erwartete Fehlerschwere bzw. Höhe des Schadens
 - Eintrittswahrscheinlichkeit einer Fehlerwirkung
 - Kombinierte Betrachtung von Schwere und Eintrittswahrscheinlichkeit (Risiko = Schadenshöhe * Eintrittswahrscheinlichkeit)
 - Wahrnehmung der Fehlerwirkung
 - Priorität der Anforderungen durch den Kunden
 - Gewichtung der Qualitätsmerkmale durch den Kunden
 - Priorität der Testfälle aus Sicht der Entwicklung (schwerwiegende Auswirkungen oder/und Komplexes zuerst)
 - Hohes Projektrisiko
 - Wo viele Fehler sind, finden sich meist noch mehr!
Eine Änderung der Priorität muss möglich sein!



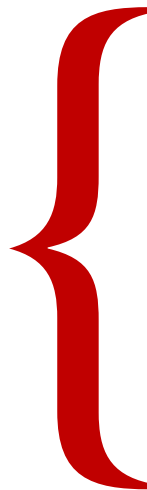
Prioritise tests so that
whenever you **stop testing**

you have done the **best testing**
in the **time available**.

Quelle (2013): <http://kehittaminen.turkuamk.fi/msuni/Testing/TestingCourse03-01-slides.pdf>

Kapitel 1

Grundlagen des Softwaretestens



Begriffe und Motivation

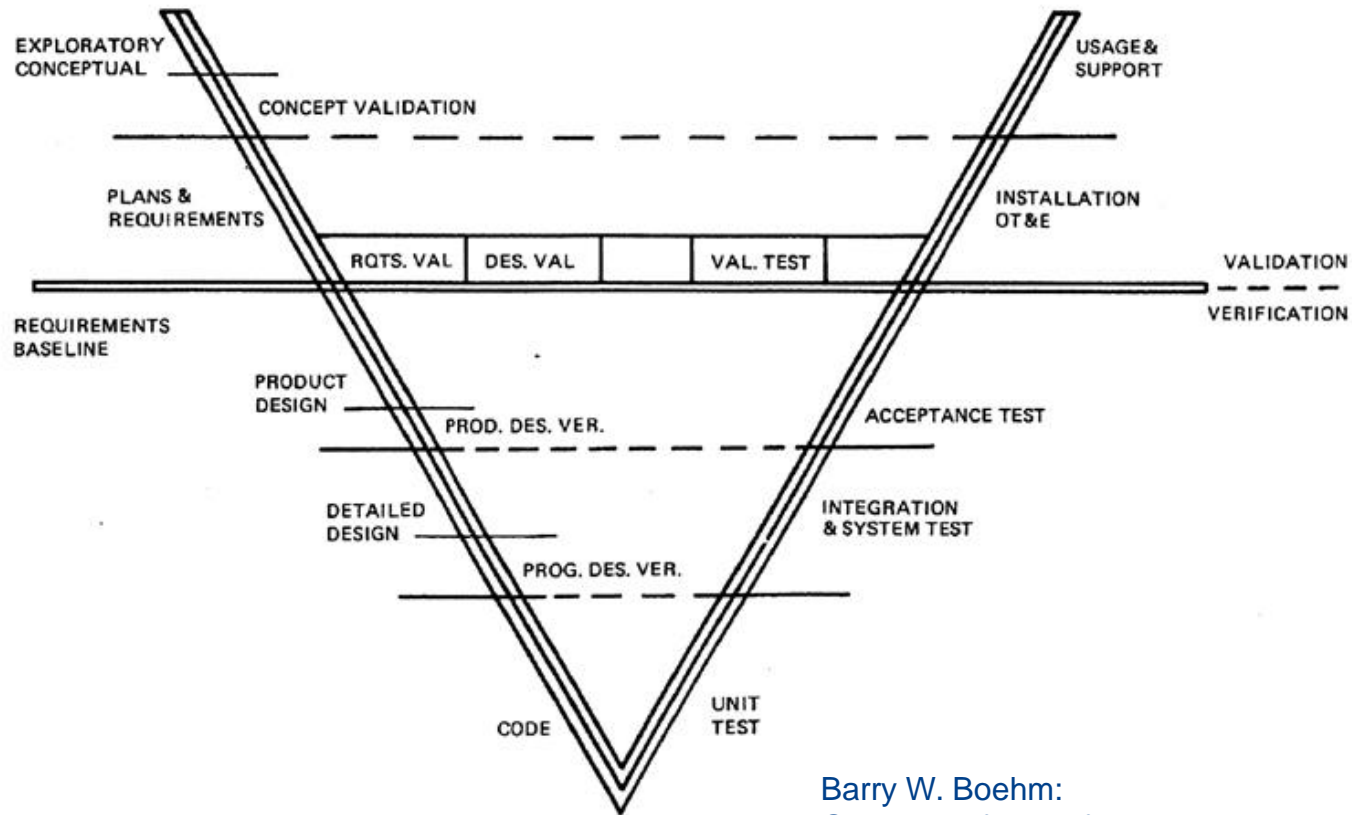
Grundsätze des Softwaretestens

Testprozess

Testfälle, Sollwerte und Testorakel

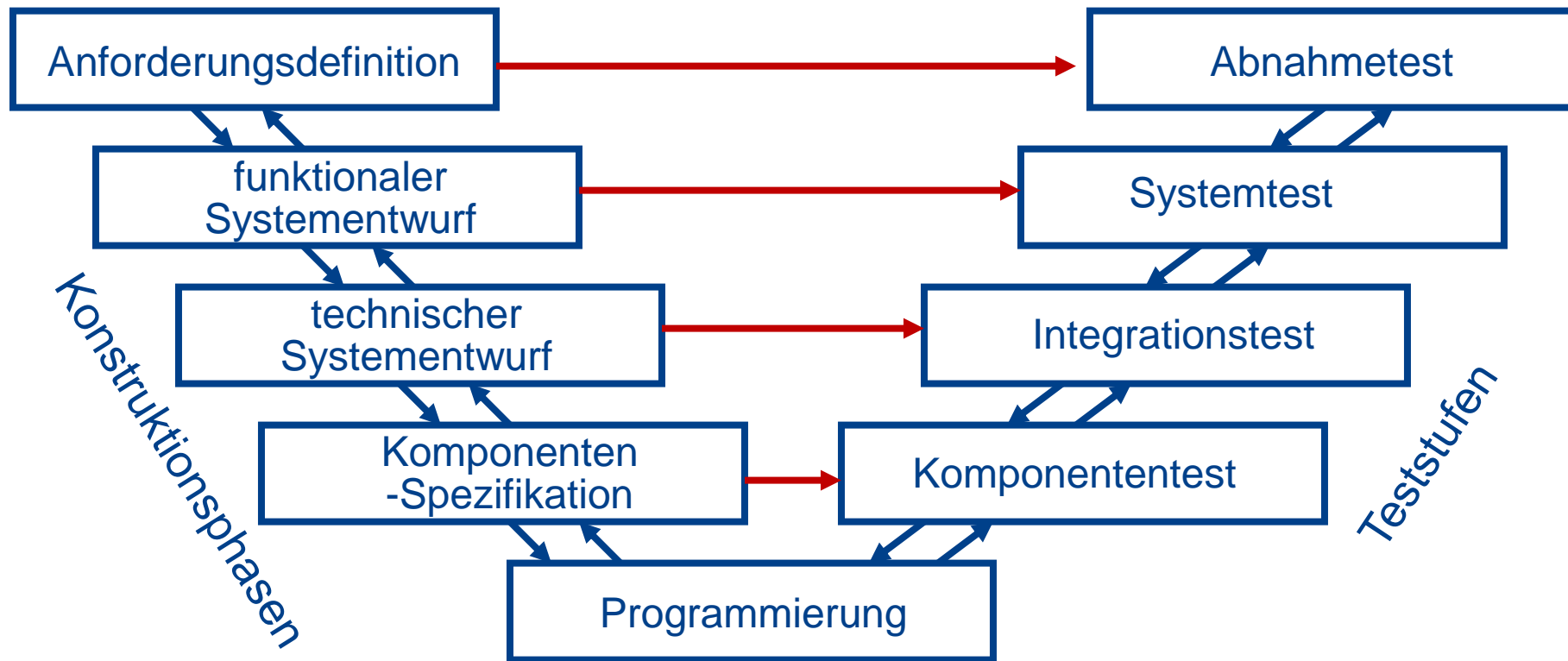
Psychologie des Testens

V-Modell (B. Boehm, 1979)



Barry W. Boehm:
Guidelines for Verifying and Validating Software
Requirements and Design Specification.
EURO IFIP 79, P.A. Samet (eds.) North-Holland, IFIP
1979, 711-719

Allgemeines V-Modell



Legende

→ Testfälle basieren auf den entsprechenden Dokumenten

Testprozess

- eigenständiger Prozess
- eng verzahnt mit Softwareentwicklung.
- verfeinerter Ablaufplan für die Tests jeder Teststufe notwendig
- Die Entwicklungsaufgabe »Test« in kleine Arbeitsabschnitte (Aktivitäten) aufteilen
- Testprozess: Menge der Testaktivitäten
- spezifischer Softwaretestprozess von vielen Faktoren abhängig
- Teststrategie eines Unternehmens:
 - Welche Testaktivitäten im Testprozess?
 - Einsatz der Aktivitäten
 - Wann finden welche Aktivitäten statt?

Testprozess im Kontext

Kontextabhängige Faktoren, die den Testprozess beeinflussen:

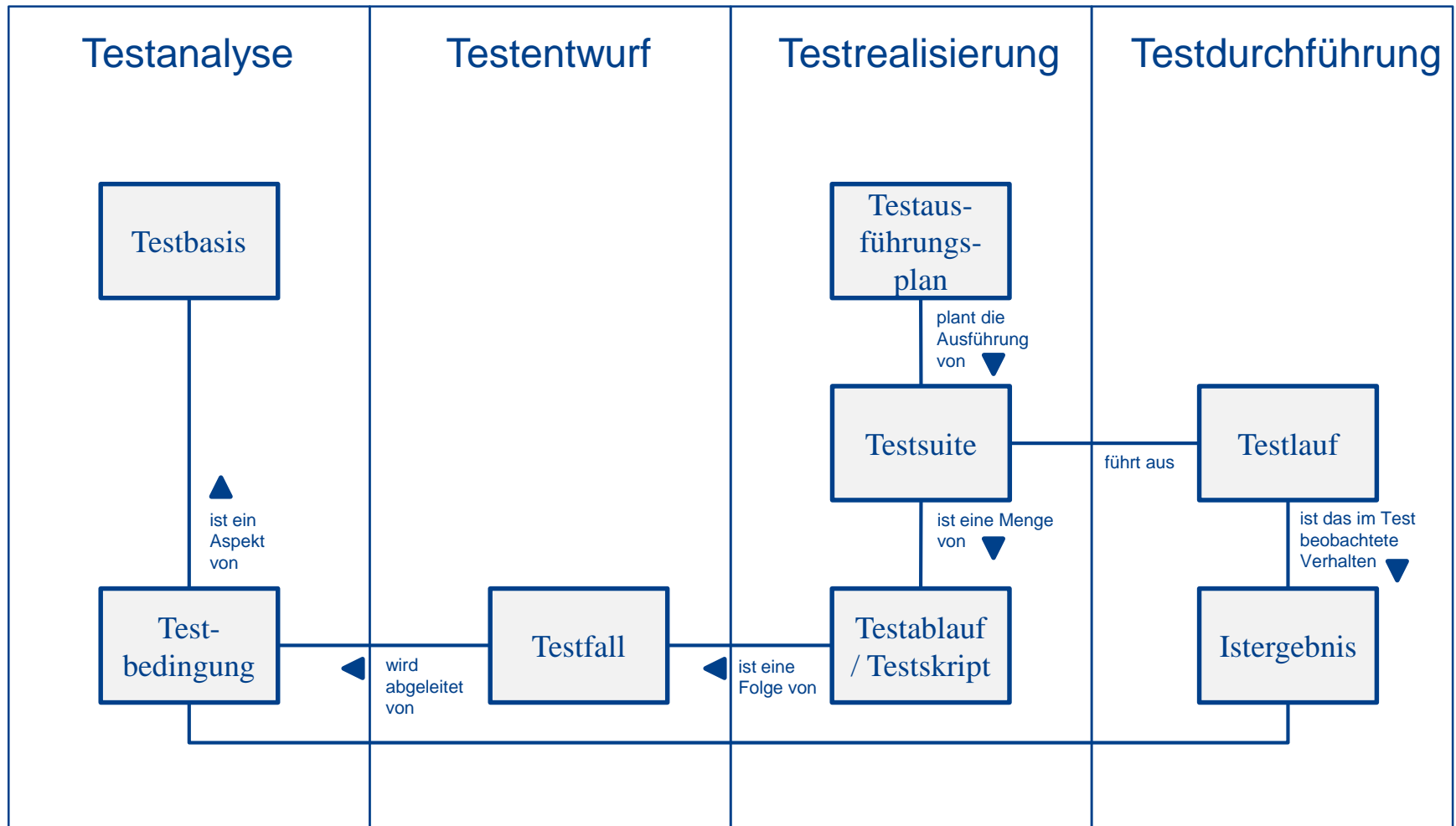
- Verwendetes Softwareentwicklungslebenszyklus-Modell
- Mögliche Teststufen und Testarten
- Produkt- und Projektrisiken
- Geschäftsbereich
- Betriebliche Restriktionen
 - Budget und Ressourcen
 - Komplexität
 - Deadlines, Fristen
 - Vertragliche und regulatorische Anforderungen
- Richtlinien und Praktiken des jeweiligen Unternehmens
- Geforderte interne und externe Standards

Testprozess im Kontext

Die folgenden Abschnitte beschreiben allgemeine Aspekte von Testprozessen in Unternehmen in Bezug auf Folgendes:

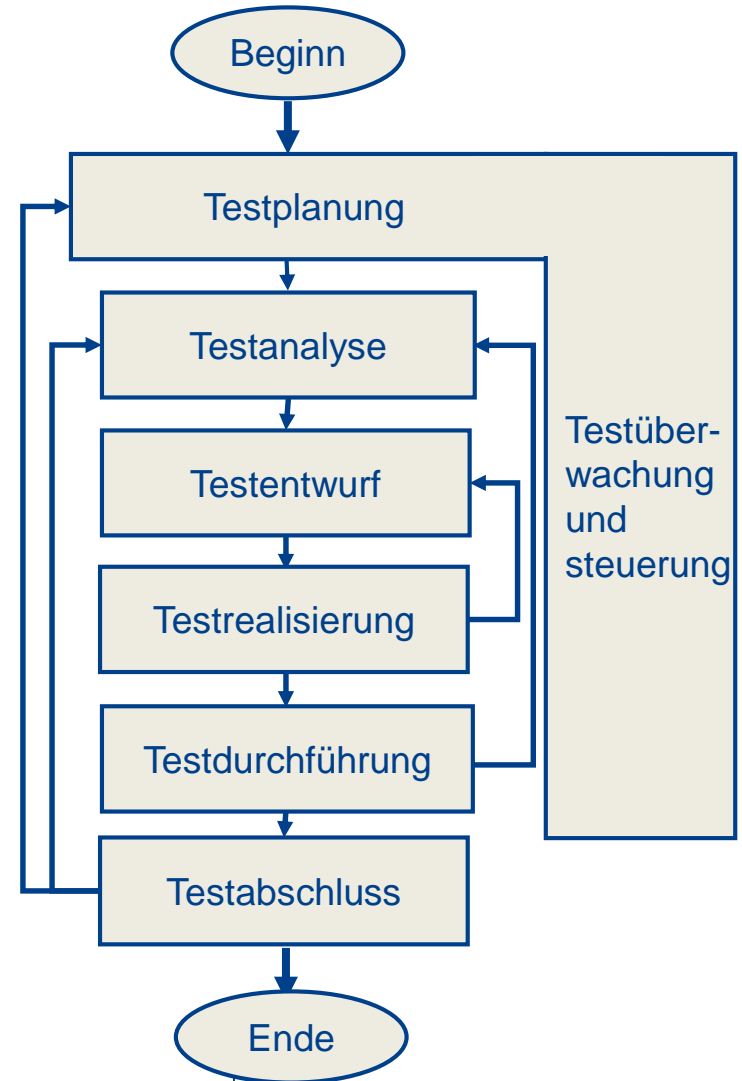
- Testaktivitäten und Aufgaben
- Testarbeitsergebnisse
- Rückverfolgbarkeit zwischen Testbasis und Testarbeitsergebnissen

Rückverfolgbarkeit



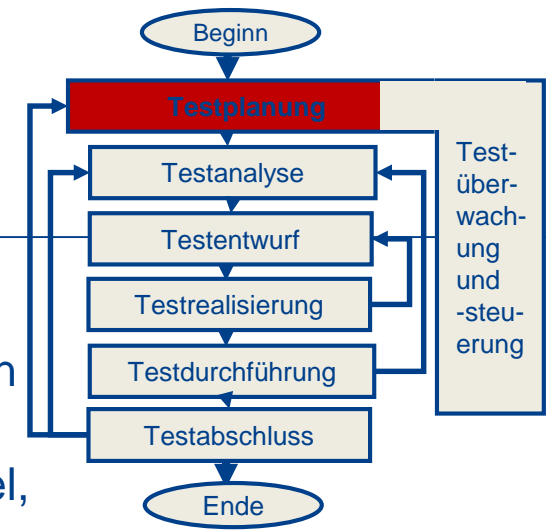
Aktivitäten des Testprozesses nach CTFL Lehrplan 2018

- Testplanung
- Testüberwachung und -steuerung
- Testanalyse
- Testentwurf
- Testrealisierung
- Testdurchführung
- Testabschluss
- Diese Aktivitäten werden z.T. zeitlich überlappend oder parallel ausgeführt.
- Der Testprozess ist für jede Teststufe geeignet zu gestalten.



Testaufgaben: Testplanung

- Festlegung der Teststrategie
 - Testziele, Umfang, Risiken des Testens festlegen
 - **Testvorgehensweise spezifizieren**
(Techniken, Testobjekte, Überdeckung, Testmittel, sowie Teams, die am Test beteiligt sind).
- Detaillierung der Testplanung
 - Testressourcen (z.B. Personal, Testumgebung, PCs).
 - Wie intensiv sollen welche Systemteile getestet werden?
 - Welche Testverfahren sollen verwendet werden?
 - Endekriterien bestimmen. Welcher Überdeckungsgrad soll erreicht werden?
 - Priorisierung der Tests (unter Berücksichtigung des Risikos im Fehlerfall).
 - Werkzeugunterstützung einplanen (Einsatz, Beschaffung, Einarbeitung).
 - Grobe Testplanung festlegen
- Testplanung und Teststrategie im (Master-)testkonzept (engl. test plan) festhalten



Testverfahren = Eine Kombination von Tätigkeiten zum systematischen Erzeugen eines Testproduktes. Testverfahren sind unter Anderem verfügbar für: Testschätzung, Fehlermanagement, Produktrisikoaanalyse, Testentwurf, Testdurchführung und Reviews [Glossar]

Arbeitsergebnisse: Testplanung

- Arbeitsergebnisse der Testplanung beinhalten Testkonzepte
- Testkonzept
 - Informationen über Testbasis, auf die sich die anderen Testarbeitsergebnisse via Rückverfolgbarkeitsinformationen beziehen werden
 - Endekriterien, die während der Testüberwachung und -steuerung genutzt werden.

Inhalts eines Testkonzepts:

- Hintergrund/Testziele
- Referenzierte Dokumente
- Testvorgehensweise
- Berichterstattung
- Abhängigkeiten
- Endekriterien
- (...)



Testaufgaben: Testüberwachung und -steuerung

- **Testüberwachung**

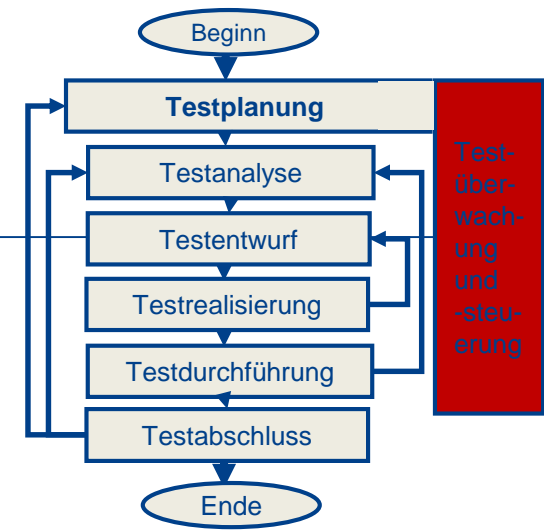
- beinhaltet Vergleich des tatsächlichen Fortschritts mit der Testplanung
- Nutzung von Überwachungsmetriken und Endekriterien, die im Testkonzept definiert wurden

- **Teststeuerung**

- Ergreifen notwendiger Maßnahmen zur Erreichung der Ziele, die im Testkonzept definiert wurden

- **Testüberwachung und -steuerung**

- Messen und Analysieren der Ergebnisse
- Überwachen und Dokumentieren von Testfortschritt, erreichter Testabdeckung und der Endekriterien
- Anstoß von Korrekturmaßnahmen
- Anpassung der Zeit- und Ressourcenplanung
- Treffen von Entscheidungen, ob mehr Tests notwendig sind



Arbeitsergebnisse: Testüberwachung und -steuerung

- Arbeitsergebnisse der Testüberwachung und –steuerung beinhalten üblicherweise verschiedene Arten von Testberichten, unter anderem:
 - **Testfortschrittsberichte**
(erstellt auf kontinuierlicher und/oder regelmäßiger Basis)
 - **Testabschlussberichte**
(erstellt bei verschiedenen Abschlussmeilensteinen)

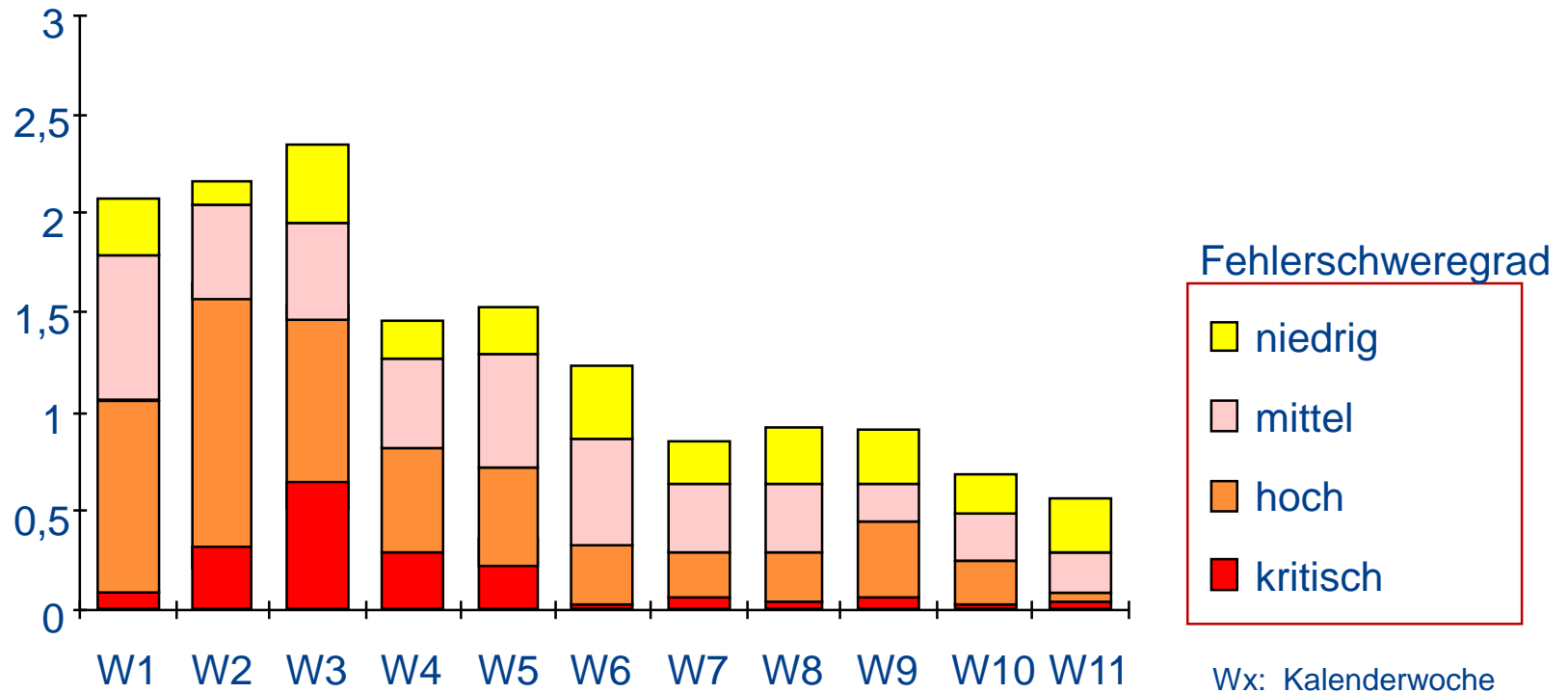
Inhalt:

- zielgruppenrelevante Details über den Testfortschritt zum Zeitpunkt des Berichts
- Testdurchführungsergebnisse, sobald diese verfügbar sind.

Arbeitsergebnis: Testbericht (1)

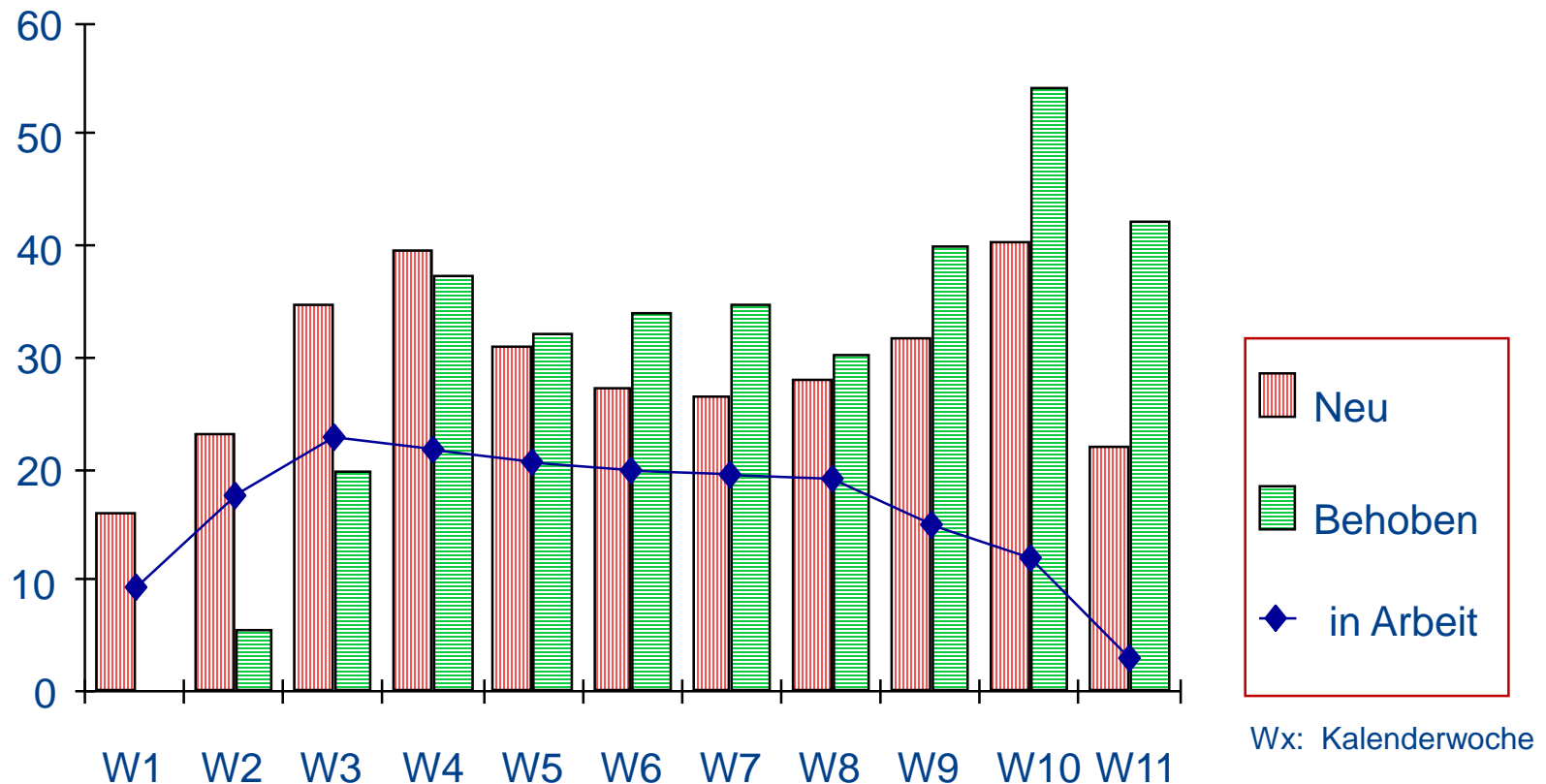
- Beispiel für Endekriterien:
maximale Anzahl gefundener Fehler nach Fehlerschweregrad

Neu / Teststunde



Arbeitsergebnis: Testbericht (2)

- Beispiel für Endekriterien:
Vergleich gefundene / behobene Fehler



Testaufgaben und Arbeitsergebnisse: Testanalyse

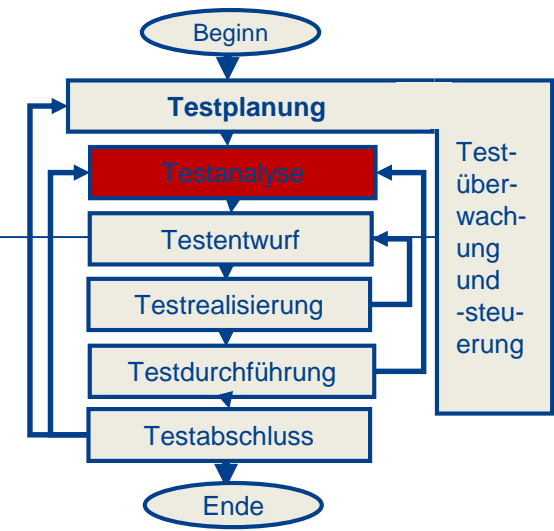
Testanalyse: „Was wird getestet?“

Testaufgaben

- Review der Testbasis (Anforderungen, Architektur, Entwurf, Schnittstellen, ...).
- Bewertung der Testbarkeit von Anforderungen und System
- Risikoanalyse, die funktionale, nicht-funktionale und strukturelle Aspekte der Software in Betracht zieht
- Identifikation von Features und Feature-Sets, die getestet werden müssen
- Detaillierung von Testzielen zu konkreten Testbedingungen
- Definition und Priorisierung der Testbedingungen für jedes Feature
- Erfassung von bidirektionaler Rückverfolgbarkeit zwischen jedem Element der Testbasis und den zugehörigen Testbedingungen

Arbeitsergebnisse

- definierte und priorisierte Testbedingungen
- Testbedingungen sind rückverfolgbar zu den abgedeckten Elementen der Testbasis

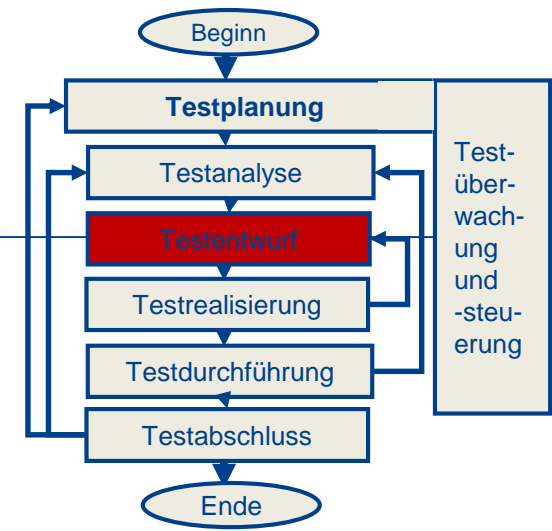


Testaufgaben: Testentwurf (1/2)

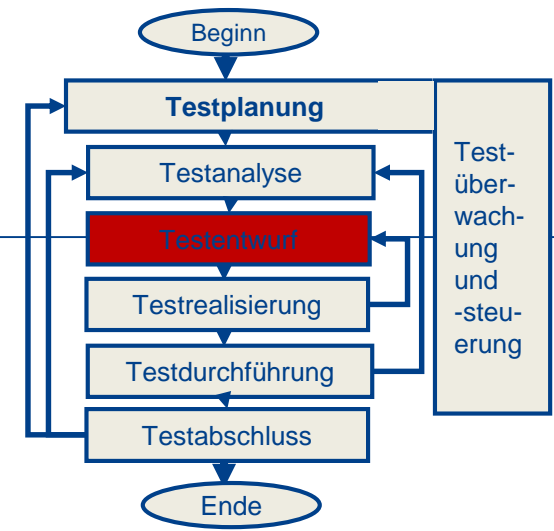
Testentwurf: „Wie wird es getestet?“

Testaufgaben

- Testentwurfsverfahren: Spezifikation von (Mengen von) Testfällen
- Priorisierung von (Mengen von) Testfällen
- Identifizierung von notwendigen Testdaten
- Entwurf der Testumgebung (Infrastruktur, Werkzeuge, ...).
- Sicherstellung der Rückverfolgbarkeit



Testaufgaben



- Testfälle umfassen mehr als nur Testdaten (die Eingaben)!
 - Vorbedingung (vor dem Test erfüllt)
 - Erwartetes Ergebnis (vor der Testdurchführung festlegen!)
 - Nachbedingung (nach dem Test erfüllt)
- Testrealisierung: Verknüpfung von Testfällen zu Testsuiten
 - Testfall: nur wenige elementare Bearbeitungsschritte umfassen
 - bei Testentwurf beachten: welche Nachbedingungen von Testfällen erfüllen welche Vorbedingungen von Testfällen?

Arbeitsergebnisse

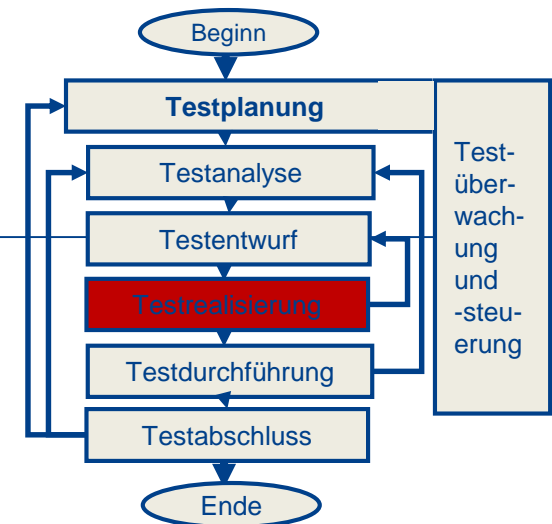
-
- ```
graph TD;
 A([Beginn]) --> B[Testplanung];
 B --> C[Testanalyse];
 C --> D[Testentwurf];
 D --> E[Testrealisierung];
 E --> F[Testdurchführung];
 F --> G[Testabschluss];
 G --> H([Ende]);
 I[Testüberwachung und -steuerung] --> B;
 I --> C;
 I --> D;
 I --> E;
 I --> F;
 G --> B;
 F --> C;
 E --> D;
```

# Testaufgaben: Testrealisierung

**Testrealisierung:**  
„Ist alles für die Durchführung der Tests bereit?“

## Testaufgaben

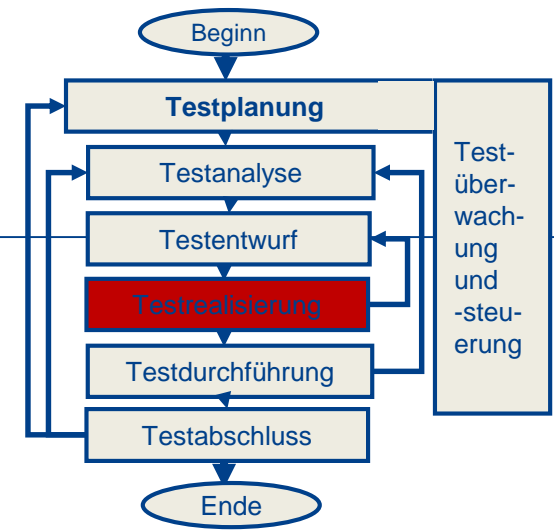
- Entwicklung und Priorisierung von Testabläufen
- Zusammenstellung/Erstellung der Testsuiten aus den Testabläufen.
- Anordnen der Testsuiten innerhalb eines Testausführungsplans
- Aufbau der Testumgebung
- Verifizierung, dass alles, was benötigt wird, korrekt aufgesetzt ist
- Vorbereitung von Testdaten und Sicherstellung, dass sie ordnungsgemäß in die Testumgebung geladen sind
- Verifizierung und Aktualisierung der bidirektionalen Rückverfolgbarkeit
- Die Erstellung automatisierter Testskripte





# Arbeitsergebnisse: Testrealisierung

## Arbeitsergebnisse

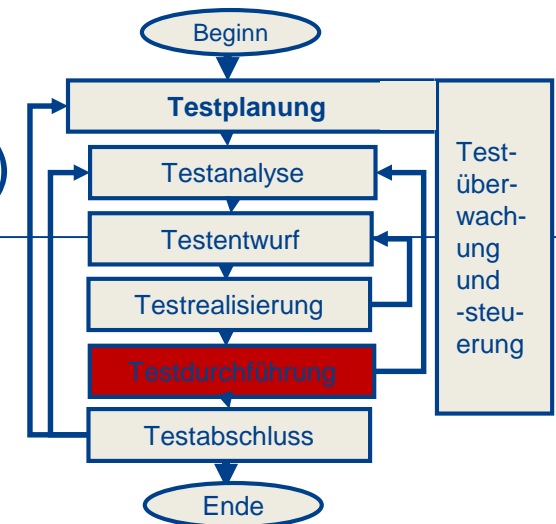


## Testabläufe und die Aneinanderreihung dieser Testabläufe

- **Testsuiten**  
mehrere Testfälle, bei der die Nachbedingungen des einen Tests die Vorbedingungen des folgenden Tests erfüllen
- **Testausführungsplan**  
Plan für die Ausführung von Testabläufen & Testabläufe mit ihrem Kontext

# Testaufgaben: Testdurchführung (1 von 2)

Testdurchführung: Testsuiten entsprechend des Testausführungsplan durchführen.

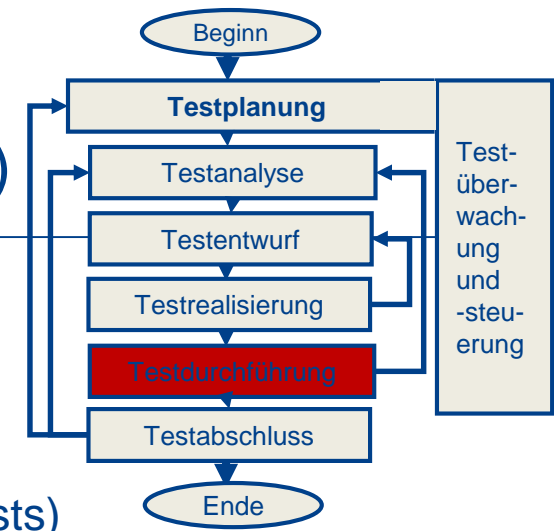


## Testaufgaben

- Aufzeichnung der IDs und Versionen der Testelemente des Testobjekts, der Testwerkzeuge und Testmittel
- Durchführung der Tests entweder manuell oder durch Nutzung von Testausführungswerkzeugen
- **Vergleich der Istergebnisse mit den erwarteten Ergebnissen**
- Fehlerschweregrad (Fehlerklasse) bestimmen
- Priorität für die Beseitigung festlegen
- Ursachenanalyse der Anomalien (z.B. Fehlerzustände im Code, Fehler im Testfall, falsch positive Ergebnisse, fehlerhafte Testumgebung)
- Bericht über Fehlerzustände auf Grundlage der beobachteten Fehlerwirkungen

# Testaufgaben: Testdurchführung (2 von 2)

- Aufzeichnung der Ergebnisse der Testdurchführung (z.B. bestanden, fehlgeschlagen, blockiert)
- Wiederholung von Testaktivitäten (z.B. korrigierter Test, Fehlernachtests, Regressionstests)
- Verifizierung und Aktualisierung der bidirektionalen Rückverfolgbarkeit zwischen:
  - Testbasis
  - Testbedingungen
  - Testfällen
  - Testabläufen
  - Testergebnissen
- Testende erreicht?
- Überdeckungsgrad erreicht?
  - Problem: unerreichbarer Programmcode

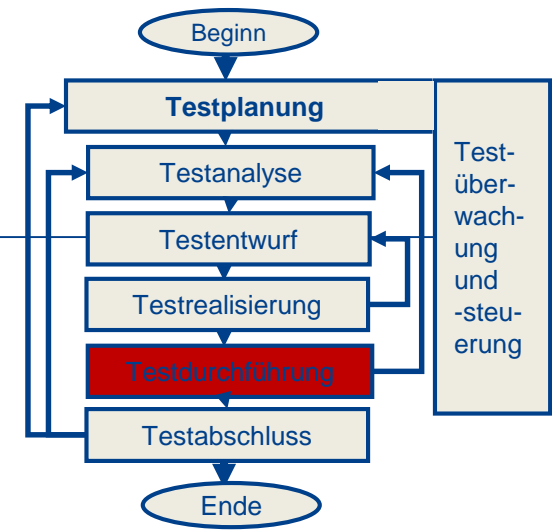


# Arbeitsergebnisse: Testdurchführung

## Arbeitsergebnisse

- Dokumentation des Status individueller Testfälle oder Testabläufe (z.B. ausführbar, bestanden, fehlgeschlagen, blockiert, geplant, ausgelassen usw.)
- Ggfs. Fehlerberichte
- Testprotokoll:
  - Was wurde wann, von wem, wie intensiv und mit welchem Ergebnis getestet?
  - Testdurchführung muss auch für nicht beteiligte Personen nachvollziehbar sein
  - genutzte Testelemente, Testwerkzeuge und Testmittel
  - Auswertung mit Hinblick auf die im Testkonzept festgelegten Endekriterien
- Nachweis, ob die geplante Teststrategie tatsächlich umgesetzt wurde
- Weiterer Aufwand gerechtfertigt?
  - Faktoren für das Testende in der Praxis: Zeit und Kosten
  - Entscheidung: mehr Tests? Definierte Endekriterien anpassen?.

Anmerkung: Ein Testbericht kann als Teststatusbericht oder ein Testabschlussbericht erstellt werden.



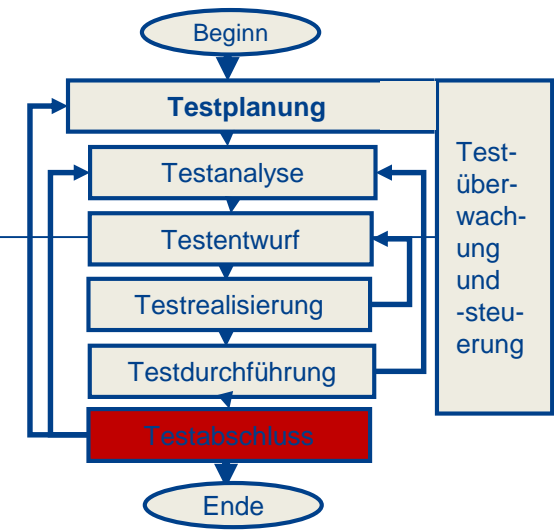
# Testaufgaben: Testabschluss

## Testabschluss

Daten aus beendeten Testaktivitäten sammeln, um Erfahrungen, Testmittel und andere relevante Informationen zu konsolidieren.

## Testaufgaben

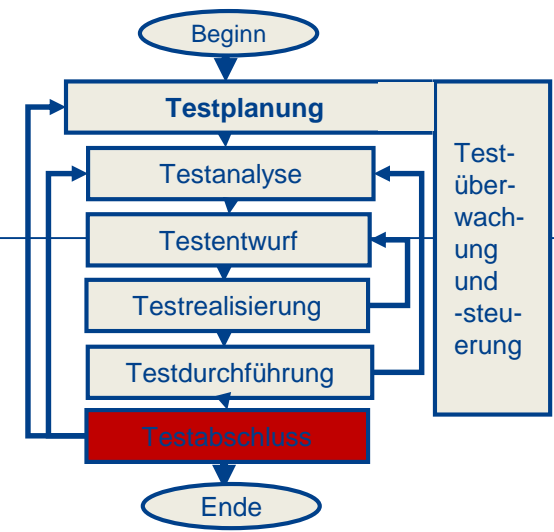
- Prüfung:
  - alle Fehlerberichte abgeschlossen?
  - Erstellung von Änderungsanforderungen für bestehende Fehler
- Abschluss und Archivieren der Testumgebung, der Testdaten, der Testinfrastruktur und anderer Testmittel für spätere Wiederverwendung
- Übergabe der Testmittel an die Wartungsteams, andere Projektteams oder Stakeholder
- „lessons learned“: gewonnene Erkenntnisse analysieren, notwendige Änderungen für zukünftige Iterationen, Produktreleases und Projekte bestimmen
- Verbesserung des Testprozesses durch Umsetzung der Erkenntnisse
- Erstellung des Testabschlussberichts



## Arbeitsergebnisse: Testabschluss

## Arbeitsergebnisse

- Testabschlussberichte
- offene Punkte zur Verbesserung in nachfolgenden Projekten
- finalisierte Testmittel



# Rückverfolgbarkeit zwischen Testbasis und Testarbeitsergebnissen

## Rückverfolgbarkeit

- Der Grad, zu dem eine Beziehung zwischen mehreren Arbeitsergebnissen hergestellt werden kann [nach ISO 19506].

## Vertikale Rückverfolgbarkeit

- Rückverfolgung von Anforderungen von der Entwicklungsdokumentation bis zu den Komponenten.

## Horizontale Rückverfolgbarkeit

- Rückverfolgung von Anforderungen einer Teststufe über die Ebenen der Testdokumentation (z.B. Testkonzept, Testentwurfsspezifikation, Testfallspezifikation, Testablaufspezifikation oder Testskripte).

## Vorteile der Rückverfolgbarkeit

- Bewertung der Testüberdeckung
- Auswirkungsanalyse von Änderungen
- Nachvollziehbarkeit von Tests
- Verständlichkeit von Testfortschrittsberichten und Testabschlussberichten verbessern
- IT-Governance-Kriterien erfüllen
- Weitergabe von Informationen zur Beurteilung von Produktqualität, Prozessfähigkeit und Projektfortschritt gegenüber Geschäftszielen

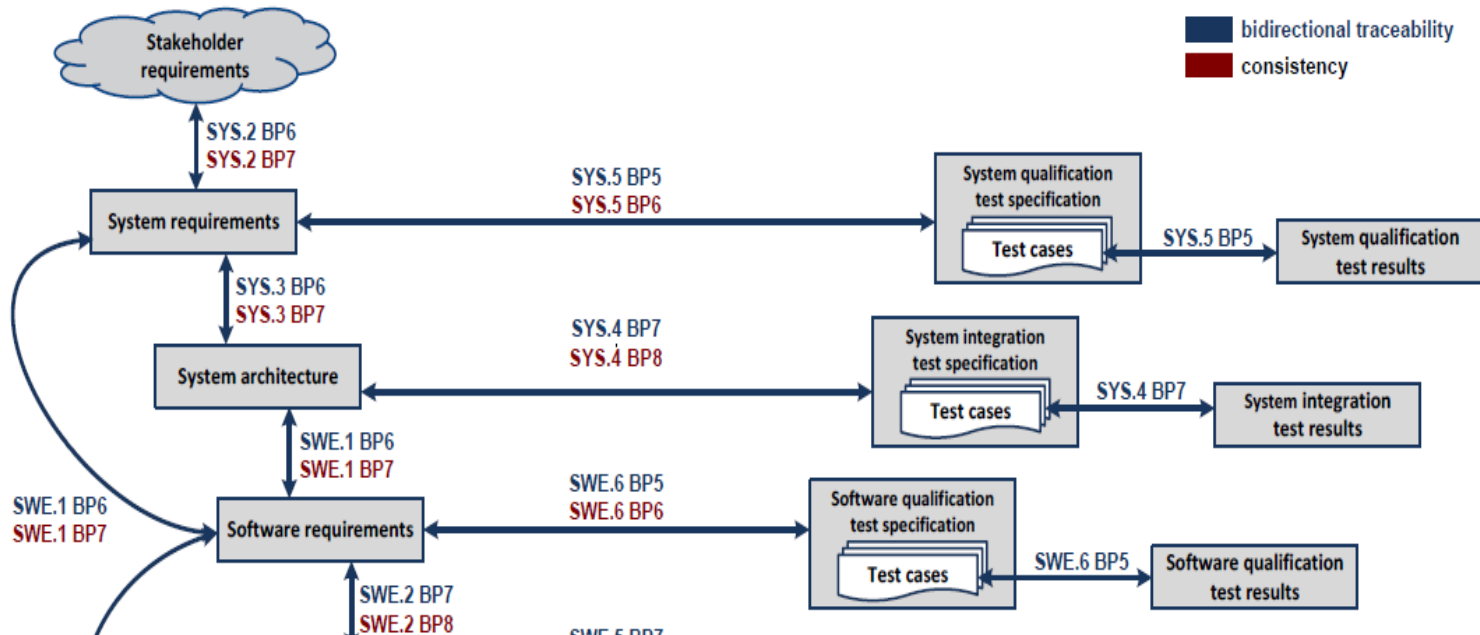


# Rückverfolgbarkeit zwischen Testbasis und Testarbeitsergebnissen

Furthermore, bidirectional traceability has been explicitly defined between

- test cases and test results, and
- change requests and work products affected by these change requests.

An overview of bidirectional traceability and consistency is depicted in the following figure.

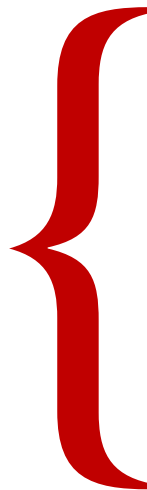


Source: Automotive SPICE 3.1



# Kapitel 1

## Grundlagen des Softwaretestens



Begriffe und Motivation

Grundsätze des Softwaretestens

Testprozess

Testfälle, erwartete Ergebnisse und Testorakel

Psychologie des Testens

# Kriterien für Testfälle

- Testfälle zur Prüfung des erwarteten, spezifizierten Verhaltens
  - „Positivtest“
  - erwartete Eingaben bzw. erwartete Bedienung
- Testfälle zur Prüfung der spezifizierten Ausnahme- und Fehlersituationen
  - „Negativtest“
  - Fehleingaben bzw. Fehlbedienung (nicht immer einfach, z.B. Überlast)
  - Einstellung des Testers: Test soll zeigen, dass eine Komponente oder ein System nicht funktioniert
- Testfälle zur Prüfung von unspezifizierten Ausnahmen
  - „Negativtest“ bzw. „Robustheitstest“
  - unerwartete Fehleingaben bzw. unerwartete Fehlbedienung



# Testspezifikation – abstrakte und konkrete Testfälle (1)

## Beispiel

Eine Firma hat ein Programm in Auftrag gegeben, das die Weihnachtsgratifikation der Mitarbeiter in Abhängigkeit von der Firmenzugehörigkeit berechnen soll.  
In der Beschreibung der Anforderungen findet sich folgende Textpassage:

»Mitarbeiter erhalten ab einer Zugehörigkeit zur Firma von mehr als drei Jahren 50 % des Monatsgehalts als Weihnachtsgratifikation. Mitarbeiter, die länger als fünf Jahre in der Firma tätig sind, erhalten 75 %. Bei einer Firmenzugehörigkeit von mehr als acht Jahren wird eine Gratifikation von 100 % gewährt.«

Wie sehen adäquate Testfälle aus?



## Testspezifikation – abstrakte und konkrete Testfälle (2)

- Aus dem Text lassen sich folgende Zusammenhänge für die Gewährung der Gratifikation in Abhängigkeit der Firmenzugehörigkeit aufstellen:

Firmenzugehörigkeit  $\leq 3$  ergibt Gratifikation = 0 %

$3 < \text{Firmenzugehörigkeit} \leq 5$  ergibt Gratifikation = 50 %

$5 < \text{Firmenzugehörigkeit} \leq 8$  ergibt Gratifikation = 75 %

Firmenzugehörigkeit  $> 8$  ergibt Gratifikation = 100 %



## Testspezifikation – abstrakte und konkrete Testfälle (3)

| Abstrakter Testfall                         | 1          | 2              | 3              | 4       |
|---------------------------------------------|------------|----------------|----------------|---------|
| Eingabewert x<br>(Firmenzugehörigkeit)      | $x \leq 3$ | $3 < x \leq 5$ | $5 < x \leq 8$ | $x > 8$ |
| Erwartetes Ergebnis<br>(Gratifikation in %) | 0          | 50             | 75             | 100     |

| Konkreter Testfall                          | 1 | 2  | 3  | 4   |
|---------------------------------------------|---|----|----|-----|
| Eingabewert x<br>(Firmenzugehörigkeit)      | 2 | 4  | 7  | 12  |
| Erwartetes Ergebnis<br>(Gratifikation in %) | 0 | 50 | 75 | 100 |

Anmerkung: Es wurde keine Rand-, Vor- und Nachbedingungen vermerkt, die Testfälle wurden nicht systematisch hergeleitet, nur positive Tests mit erwarteten Ergebnissen

# Erwartete Ergebnisse und Testorakel

- Nach jeder Testfalldurchführung: Fehlerwirkung ja / nein?
  - Ist-Ergebnis mit erwartetem Ergebnis vergleichen
- Das erwartete Ergebnis des Testobjekts muss vorab spezifiziert werden.
  - Aufgabe des Testers: geeignete Quellen finden
- "Testorakel" wird befragt, um die erwarteten Ergebnisse vorherzusagen.
- Haben Sie gemerkt, dass der Fall „Betriebszugehörigkeit  $\leq 3$  Jahre“ nicht spezifiziert wurde?



# Testorakel

- erwarteten Ergebnisse müssen aus der Spezifikation abgeleitet werden.
- Drei Möglichkeiten:
  - **Tester leitet das erwartete Ergebnis ab**
    - Basis: Eingabedaten & Spezifikation des Testobjekts
    - in der Praxis das übliche Vorgehen
  - **Formale Spezifikation vorhanden?**
    - ausführbarer Prototyp werkzeuggestützt erzeugen
    - Ergebnisse des Prototypen dienen als Testorakel für den eigentlichen Test
  - ***Back-to-back-Test***
    - Software mehrfach und parallel von unabhängigen Entwicklungsgruppen erstellen lassen
    - Die Ergebnisse aller Programmversionen werden verglichen (bei gleichen Eingaben)
    - unterschiedliche Ergebnisse: Fehlerzustand in einer Version
    - Nur Fehlerzustände unentdeckt, die in allen Versionen die gleichen Fehlerwirkungen haben



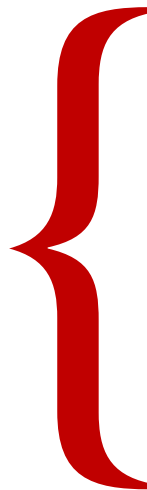
## Weitere Testorakel

- Benutzungshandbuch verwenden
- (Getestete!) Vorgängerversionen, Programme mit ähnlicher Funktionalität
- Nicht immer lassen sich exakte Werte vorhersagen oder ermitteln  
Toleranzbereich festlegen, Plausibilität prüfen
- Erfahrung ist wichtig
- System selbst (Nutzungsprofil erstellen) -> schlechteste Möglichkeit



# Kapitel 1

## Grundlagen des Softwaretestens



Begriffe und Motivation

Grundsätze des Softwaretestens

Testprozess

Testfälle, erwartete Ergebnisse und Testorakel

Psychologie des Testens

# Psychologie des Testens

- Errare humanum est
  - aber wer gibt es schon gerne zu?

- Entwicklung = konstruktiv  
Test = destruktiv ?

»Testen ist eine extrem kreative  
und intellektuell herausfordernde Aufgabe«

Myers, Glenford J.:  
Methodisches Testen von Programmen  
Oldenbourg, 2001 (7. Auflage)

- Kann der Entwickler sein eigenes Programm testen?  
Was meinen Sie?

# Psychologie des Testens - Entwicklertest



- Blindheit gegenüber den eigenen Fehlern
  - Aufgabenstellung falsch verstanden?
  - Folge: grundsätzlicher Entwurfsfehler auch im Test vorhanden
- Kein Einarbeitungsaufwand
  - der Entwickler kennt sein Testobjekt

# Psychologie des Testens - Unabhängiges Testteam

- Unvoreingenommenheit
  - nicht das eigene Produkt
  - Verständnis des Entwicklers anders als Verständnis des Testers
- Einarbeitung notwendig
  - Tester eignet sich das benötigte Wissen über das Testobjekt an
  - Kostet Zeit
- Test-Know-how
  - Hat der Tester
  - Hat der Entwickler nicht (zwangsläufig)
    - bzw. er müsste es sich erst aneignen
    - bzw. er hat es auf der Hochschule gelernt

# Psychologie des Testens - mögliche Abstufungen

- Tests werden durchgeführt von ...
  - dem Entwickler selbst
  - einem Kollegen des Entwicklers im gleichen Projekt
  - Personen anderer Abteilungen
  - Personen anderer Organisationen
- Bei allen Möglichkeiten können Werkzeuge eingesetzt werden.
- Die Aufteilung ist produkt- und projektabhängig.
- Wichtig ist die richtige Mischung und Ausgewogenheit zwischen »unabhängigen Tests« und Entwicklertests.

# Psychologie des Testens - Mitteilung von Fehlern

- Mitteilung von gefundenen Fehlerwirkungen oder Abweichungen:
  - neutrale, objektive und konstruktive Art und Weise
  - ungestörte, offene Kommunikation.
- Anderen Menschen Fehlhandlungen nachweisen erfordert viel Fingerspitzengefühl
- Reproduzierbarkeit ist wichtig!
  - Dokumentation der Testumgebung
  - Unterschiede zur Entwicklungsumgebung
- Eindeutige Anforderungen, präzise Spezifikation
  - »It's not a bug, it's a feature!«
- Gegenseitiges Verständnis
  - Miteinander!
  - kein Gegeneinander!



# Zusammenfassung

- **Fehlerwirkung - Nichterfüllung einer Anforderung!**
  - Beschreibung dieses Sachverhalts als Fehlerwirkung (*failure*)
  - Ursache: Fehlerzustand (*defect*) in der Software
  - die durch eine Fehlhandlung (*error*) einer Person verursacht wurde
- **Tests sind wichtige Maßnahmen zur Sicherung der Qualitätsmerkmale**
  - Funktionalität, Zuverlässigkeit, Gebrauchstauglichkeit, Informationssicherheit, Kompatibilität, Übertragbarkeit, Änderbarkeit und Effizienz [nach ISO 25010]
- **Austesten ist nicht möglich, Tests sind immer Stichproben, Fehlerwirkungen daher mit einer gewissen Wahrscheinlichkeit nachweisen**
- **Intensität und Umfang der Tests abhängig vom erwarteten Risiko bei fehlerhaftem Verhalten des Programms**
- **Priorisierung der Tests ist vorzunehmen**



# Zusammenfassung

- **Der Testprozess besteht aus den Hauptaktivitäten**
  - Testplanung
  - Testüberwachung und -steuerung
  - Testanalyse
  - Testentwurf
  - Testrealisierung
  - Testdurchführung
  - Testabschluss
- **Testfall: Eingabewert; erwartetes Ergebnis; Vorbedingungen, unter denen der Testfall ablaufen muss, und Nachbedingungen, die nach Ablauf erfüllt sein müssen.**
- **Testfalldurchführung: Testobjekt zeigt Ist-Ergebnis**
  - erwartetes und Ist-Ergebnis verschieden? ggfs. Fehlerwirkung
  - prüfen, ob ein Fehler im erwarteten Ergebnis, im Testfall, in der Spezifikation oder im Testobjekt liegt.
- **Ein Testorakel bestimmt die Soll-Werte für jeden Testfall vor Testdurchführung.**
- **Menschen machen Fehler – aber sie geben es nur ungern zu!**





## Folgende Fragen sollten Sie jetzt beantworten können

1. Definieren Sie die Begriffe Fehlerwirkung, Fehlerzustand, Fehlhandlung.
2. Was ist eine Fehlermaskierung?
3. Erläutern Sie den Unterschied zwischen Testen und Debugging.
4. Definieren Sie die Begriffe Verifizierung und Validierung.
5. Erläutern Sie, warum jeder Test eine stichprobenartige Prüfung ist.
6. Nennen Sie die Hauptmerkmale der Softwarequalität nach ISO.
7. Definieren Sie den Begriff Zuverlässigkeit eines Systems.
8. Erläutern Sie die Hauptaktivitäten des Testprozesses.
9. Charakterisieren Sie die zugehörigen Arbeitsergebnisse des Testprozesses.
10. Erläutern Sie die bidirektionale Rückverfolgbarkeit.
11. Was ist ein Testorakel?
12. Warum sollte ein Entwickler nicht seine eigenen Programme testen?



# Muster-Prüfungsfragen

Testen Sie Ihr Wissen...



# Frage 1

## 1. Welche der folgenden Antworten beschreibt eine Testbedingung? [K1]

|    |                                                                                                                                                                               |                          |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| a) | Ein Attribut einer Komponente oder eines Systems, spezifiziert oder abgeleitet aus der Anforderungsspezifikation.                                                             | <input type="checkbox"/> |
| b) | Ein Aspekt der Testbasis, der für die Erreichung bestimmter Testziele relevant ist.                                                                                           | <input type="checkbox"/> |
| c) | Der Grad, zu dem eine Komponente oder ein System Funktionen zur Verfügung stellt, welche unter festgelegten Bedingungen explizit genannte und implizite Bedürfnisse erfüllen. | <input type="checkbox"/> |
| d) | Der Anteil aller einfachen Bedingungsergebnisse, die von einer Testsuite ausgeführt wurden und unabhängig voneinander einen Entscheidungsausgang beeinflussen.                | <input type="checkbox"/> |



## Frage 1 – Lösung

### 1. Welche der folgenden Antworten beschreibt eine Testbedingung? [K1]

|    |                                                                                                                                                                               |                                     |
|----|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| a) | Ein Attribut einer Komponente oder eines Systems, spezifiziert oder abgeleitet aus der Anforderungsspezifikation.                                                             | <input type="checkbox"/>            |
| b) | Ein Aspekt der Testbasis, der für die Erreichung bestimmter Testziele relevant ist.                                                                                           | <input checked="" type="checkbox"/> |
| c) | Der Grad, zu dem eine Komponente oder ein System Funktionen zur Verfügung stellt, welche unter festgelegten Bedingungen explizit genannte und implizite Bedürfnisse erfüllen. | <input type="checkbox"/>            |
| d) | Der Anteil aller einfachen Bedingungsergebnisse, die von einer Testsuite ausgeführt wurden und unabhängig voneinander einen Entscheidungsausgang beeinflussen.                | <input type="checkbox"/>            |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 2

### 2. Welche der folgenden Aussagen beschreibt ein gültiges Ziel des Testens? [K1]

|    |                                                                                                                                        |                          |
|----|----------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| a) | Der Test soll möglichst spät starten, damit die Entwicklung genug Zeit hatte, ein gutes Produkt zu erstellen.                          | <input type="checkbox"/> |
| b) | Es sollen so viele Fehlerwirkungen wie möglich aufgedeckt werden, so dass die Fehlerursachen lokalisiert und korrigiert werden können. | <input type="checkbox"/> |
| c) | Es soll nachgewiesen werden, dass alle möglichen Fehlerzustände identifiziert wurden.                                                  | <input type="checkbox"/> |
| d) | Es soll nachgewiesen werden, dass alle verbleibenden Fehlerzustände keine Fehlerwirkungen verursachen werden.                          | <input type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 2 – Lösung

### 2. Welche der folgenden Aussagen beschreibt ein gültiges Ziel des Testens? [K1]

|    |                                                                                                                                        |                                     |
|----|----------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| a) | Der Test soll möglichst spät starten, damit die Entwicklung genug Zeit hatte, ein gutes Produkt zu erstellen.                          | <input type="checkbox"/>            |
| b) | Es sollen so viele Fehlerwirkungen wie möglich aufgedeckt werden, so dass die Fehlerursachen lokalisiert und korrigiert werden können. | <input checked="" type="checkbox"/> |
| c) | Es soll nachgewiesen werden, dass alle möglichen Fehlerzustände identifiziert wurden.                                                  | <input type="checkbox"/>            |
| d) | Es soll nachgewiesen werden, dass alle verbleibenden Fehlerzustände keine Fehlerwirkungen verursachen werden.                          | <input type="checkbox"/>            |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 3

### 3. Welche der folgenden Aussagen beschreibt den Unterschied zwischen Testen und Debuggen zutreffend? [K2]

|    |                                                                                                                                                                                                          |                          |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------|
| a) | Testen identifiziert die Ursache von Fehlerzuständen. Debuggen analysiert die Fehlerzustände und schlägt Präventionsmaßnahmen vor.                                                                       | <input type="checkbox"/> |
| b) | Dynamische Tests zeigen Fehlerwirkungen auf, die durch Fehlerzustände verursacht wurden. Debugging ist eine Entwicklungsaktivität, die die Ursache von Fehlerwirkungen findet, analysiert und beseitigt. | <input type="checkbox"/> |
| c) | Testen entfernt Fehlerzustände. Debugging identifiziert die Ursachen von Fehlerwirkungen.                                                                                                                | <input type="checkbox"/> |
| d) | Dynamische Tests verhindern die Ursache von Fehlerwirkungen. Debugging entfernt die Fehlerwirkungen.                                                                                                     | <input type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 3 – Lösung

### 3. Welche der folgenden Aussagen beschreibt den Unterschied zwischen Testen und Debuggen zutreffend? [K2]

|    |                                                                                                                                                                                                          |                                     |
|----|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------|
| a) | Testen identifiziert die Ursache von Fehlerzuständen. Debuggen analysiert die Fehlerzustände und schlägt Präventionsmaßnahmen vor.                                                                       | <input type="checkbox"/>            |
| b) | Dynamische Tests zeigen Fehlerwirkungen auf, die durch Fehlerzustände verursacht wurden. Debugging ist eine Entwicklungsaktivität, die die Ursache von Fehlerwirkungen findet, analysiert und beseitigt. | <input checked="" type="checkbox"/> |
| c) | Testen entfernt Fehlerzustände. Debugging identifiziert die Ursachen von Fehlerwirkungen.                                                                                                                | <input type="checkbox"/>            |
| d) | Dynamische Tests verhindern die Ursache von Fehlerwirkungen. Debugging entfernt die Fehlerwirkungen.                                                                                                     | <input type="checkbox"/>            |





## Frage 4

4. **Nachfolgend finden Sie eine Liste von Problemen, die während des Testens oder im Betrieb beobachtet werden können.**

**Welches Problem ist eine Fehlerwirkung? [K2]**

|    |                                                                                      |                          |
|----|--------------------------------------------------------------------------------------|--------------------------|
| a) | Das Produkt stürzte ab, als der Benutzer eine Option in einer Dialogbox auswählte.   | <input type="checkbox"/> |
| b) | Eine kompilierte Quellcodedatei wurde in der falschen Version zum Build hinzugefügt. | <input type="checkbox"/> |
| c) | Der Berechnungsalgorithmus verwendet die falschen Eingangsvariablen.                 | <input type="checkbox"/> |
| d) | Der Entwickler hat die Anforderungen an den Algorithmus falsch interpretiert.        | <input type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 4 – Lösung

4. **Nachfolgend finden Sie eine Liste von Problemen, die während des Testens oder im Betrieb beobachtet werden können.**

**Welches Problem ist eine Fehlerwirkung? [K2]**

|    |                                                                                      |                                     |
|----|--------------------------------------------------------------------------------------|-------------------------------------|
| a) | Das Produkt stürzte ab, als der Benutzer eine Option in einer Dialogbox auswählte.   | <input checked="" type="checkbox"/> |
| b) | Eine kompilierte Quellcodedatei wurde in der falschen Version zum Build hinzugefügt. | <input type="checkbox"/>            |
| c) | Der Berechnungsalgorithmus verwendet die falschen Eingangsvariablen.                 | <input type="checkbox"/>            |
| d) | Der Entwickler hat die Anforderungen an den Algorithmus falsch interpretiert.        | <input type="checkbox"/>            |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 5

5. Herr Schmidt hat über einen Zeitraum von 5 Jahren Software-Applikationen auf mobilen Geräten einem Test unterzogen. Er hat sich einen großen Erfahrungsschatz im Testen von mobilen Applikationen angeeignet und erzielt in kürzer Zeit bessere Ergebnisse als andere. Über einen längeren Zeitraum hat Herr Schmidt die existierenden automatisierten Testfälle nicht modifiziert und auch keine neuen Testfälle mehr erstellt. Dies führt dazu, dass durch Ausführung der Tests immer weniger Fehler gefunden werden. Welchen Grundsatz des Testens hat Herr Schmidt nicht beachtet? [K2]

|    |                                        |                          |
|----|----------------------------------------|--------------------------|
| a) | Testen ist abhängig vom Umfeld         | <input type="checkbox"/> |
| b) | Vollständiges Testen ist nicht möglich | <input type="checkbox"/> |
| c) | Wiederholungen haben keine Wirksamkeit | <input type="checkbox"/> |
| d) | Häufung von Fehlerzuständen            | <input type="checkbox"/> |



## Frage 5 – Lösung

5. Herr Schmidt hat über einen Zeitraum von 5 Jahren Software-Applikationen auf mobilen Geräten einem Test unterzogen. Er hat sich einen großen Erfahrungsschatz im Testen von mobilen Applikationen angeeignet und erzielt in kürzer Zeit bessere Ergebnisse als andere. Über einen längeren Zeitraum hat Herr Schmidt die existierenden automatisierten Testfälle nicht modifiziert und auch keine neuen Testfälle mehr erstellt. Dies führt dazu, dass durch Ausführung der Tests immer weniger Fehler gefunden werden. Welchen Grundsatz des Testens hat Herr Schmidt nicht beachtet? [K2]

|    |                                        |                                     |
|----|----------------------------------------|-------------------------------------|
| a) | Testen ist abhängig vom Umfeld         | <input type="checkbox"/>            |
| b) | Vollständiges Testen ist nicht möglich | <input type="checkbox"/>            |
| c) | Wiederholungen haben keine Wirksamkeit | <input checked="" type="checkbox"/> |
| d) | Häufung von Fehlerzuständen            | <input type="checkbox"/>            |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 6

**6. Inwiefern kann das Testen ein Bestandteil der Qualitätssicherung sein?**  
**[K2]**

|    |                                                                                  |                          |
|----|----------------------------------------------------------------------------------|--------------------------|
| a) | Es stellt sicher, dass Anforderungen detailliert genug sind.                     | <input type="checkbox"/> |
| b) | Es reduziert die Höhe des Risikos in Bezug auf die Qualität des Systems.         | <input type="checkbox"/> |
| c) | Es stellt sicher, dass in der Organisation Standards befolgt werden.             | <input type="checkbox"/> |
| d) | Es misst die Softwarequalität im Hinblick auf die Anzahl ausgeführter Testfälle. | <input type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 6 – Lösung

**6. Inwiefern kann das Testen ein Bestandteil der Qualitätssicherung sein?**  
**[K2]**

|    |                                                                                  |                                     |
|----|----------------------------------------------------------------------------------|-------------------------------------|
| a) | Es stellt sicher, dass Anforderungen detailliert genug sind.                     | <input type="checkbox"/>            |
| b) | Es reduziert die Höhe des Risikos in Bezug auf die Qualität des Systems.         | <input checked="" type="checkbox"/> |
| c) | Es stellt sicher, dass in der Organisation Standards befolgt werden.             | <input type="checkbox"/>            |
| d) | Es misst die Softwarequalität im Hinblick auf die Anzahl ausgeführter Testfälle. | <input type="checkbox"/>            |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 7

**7. Welche der folgenden Aktivitäten ist Teil der Hauptaktivität „Testanalyse“ im Testprozess? [K1]**

|    |                                                               |                          |
|----|---------------------------------------------------------------|--------------------------|
| a) | Identifikation der erforderlichen Infrastruktur und Werkzeuge | <input type="checkbox"/> |
| b) | Erstellen von Testsuiten basierend auf den Testskripten       | <input type="checkbox"/> |
| c) | Analyse der „Lessons learned“ zur Prozessverbesserung         | <input type="checkbox"/> |
| d) | Review der Testbasis hinsichtlich Testbarkeit                 | <input type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;



## Frage 7 – Lösung

**7. Welche der folgenden Aktivitäten ist Teil der Hauptaktivität „Testanalyse“ im Testprozess? [K1]**

|    |                                                               |                                     |
|----|---------------------------------------------------------------|-------------------------------------|
| a) | Identifikation der erforderlichen Infrastruktur und Werkzeuge | <input type="checkbox"/>            |
| b) | Erstellen von Testsuiten basierend auf den Testskripten       | <input type="checkbox"/>            |
| c) | Analyse der „Lessons learned“ zur Prozessverbesserung         | <input type="checkbox"/>            |
| d) | Review der Testbasis hinsichtlich Testbarkeit                 | <input checked="" type="checkbox"/> |

Quelle: ISTQB Foundation Level Sample Paper; SET A; 2018; Deutschsprachige Fassung/ Lokalisierung; German Testing Board;