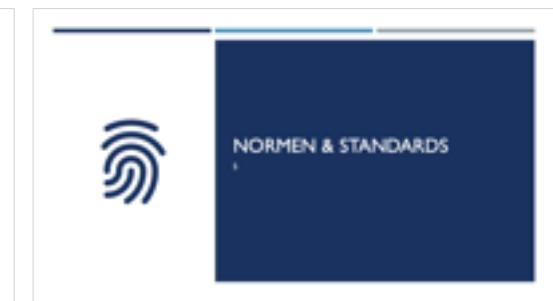
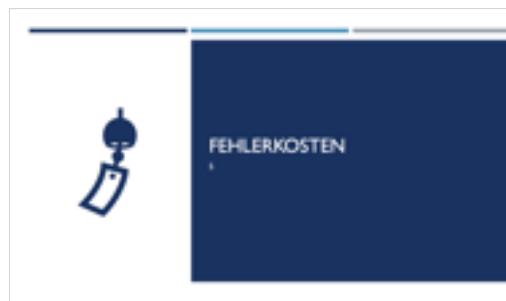
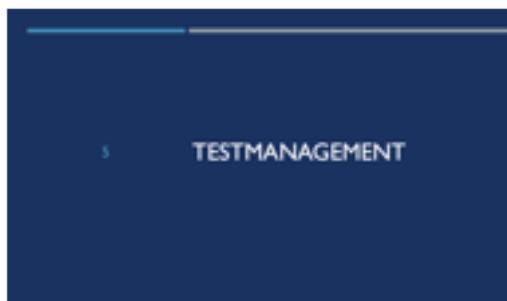
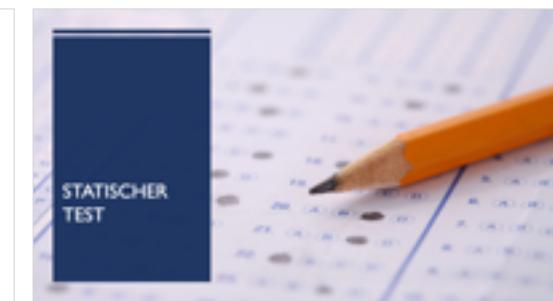


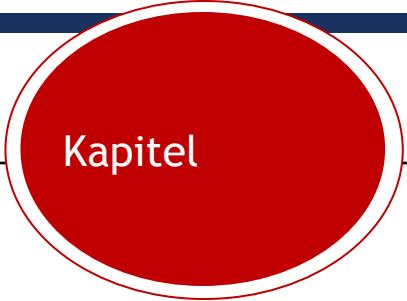


TESTEN





EINLEITUNG



Kapitel

EINLEITUNG



Certified Tester

Einstieg → Auswirkungen von Software-Fehlern

Anhang → Buchempfehlungen, Zeitschriften,
Organisationen, Tagungen

ZIELE DER LEHRVERANSTALTUNG

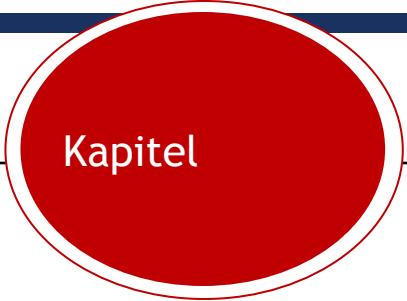


- Grundlagenvermittlung im Bereich Prüfen und Testen von Software
- Erklärung der Begriffe, Aufgaben und Tätigkeiten, Methoden und Testentwurfsverfahren
- Inhalt der Lehrveranstaltung deckt einen international festgelegten Lehrstoff für Weiterbildungseinrichtungen ab
- Vorbereitung für die Prüfung
Certified Tester - Foundation Level
 - International anerkanntes Zertifikat
 - Anerkannte professionelle Spezialisierung
 - Branchenübergreifend
(kommerzielle Software, Automotive, Web, Entertainment, ...)



ISTQB® - FOUNDATION LEVEL 2018

Grundlagen des Testens	Testen im Software Lebenszyklus	Statischer Test	Testverfahren	Testmanagement	Werkzeugunterstützung für das Testen
Was ist Testen?	Software Entwicklungs-Lebenszyklusmodelle	Grundlagen Statischer Test	Kategorien von Testverfahren	Testorganisation	Überlegungen zu Testwerkzeugen
Warum ist Testen notwendig?	Teststufen	Reviewprozess	Black-box Testverfahren	Testplanung und -schätzung	Effektive Nutzung von Werkzeugen
7 Grundsätze des Testens	Testtypen		White-box Testverfahren	Testüberwachung und -steuerung	
Testprozess	Wartungstest		Erfahrungsbasierte Testverfahren	Konfigurationsmanagement	
Psychologie des Testens				Risiken und Testen	
				Fehlermanagement	



EINLEITUNG



Certified Tester

Einstieg → Auswirkungen von Software-Fehlern

Anhang → Buchempfehlungen, Zeitschriften,
Organisationen, Tagungen



EINSTIEG

WAS IST EIN SOFTWARE-FEHLER?



9.9.1945

15:45

DER ERSTE BUG!

Admiral "Amazing Grace" Hopper<https://www.youtube.com/watch?v=S6sh8CxwWx8>

Motte im Rechner Mark II verursacht Fehler
in Relay Nr. 70, Panel F.

9th September 1947:
Moth 'bug' discovered inside a Harvard computer

Mr. Grace Murray
Hopper beseitigt
und dokumentiert
Fehler:
»First actual case
of bug being found.«

»offen«-sichtlicher
Fehler!
Beseitigung ist
einfach!

9/9

0800 Auton started
1000 .. stopped - auton ✓ { 1.2700 9.037 847 025
1300 (032) MP - MC 1.98264000 9.037 846 995 correct
033 PRO 2 2.130476415
correct 2.130676415
Relays 6-2 in 033 failed special speed test
in relay 10.000 test.
Relays changed
1100 Started Cosine Tape (Sine check)
1525 Started Multi Adder Test.

1545  Relay #70 Panel F
(moth) in relay.

16100 Auton startd.
1700 closed down.

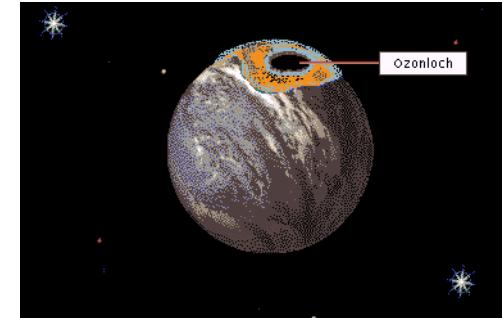
First actual case of bug being found.

<https://de.wikipedia.org/wiki/Programmfehler#/media/File:H96566k.jpg>



AUSWIRKUNGEN VON SOFTWARE-FEHLERN

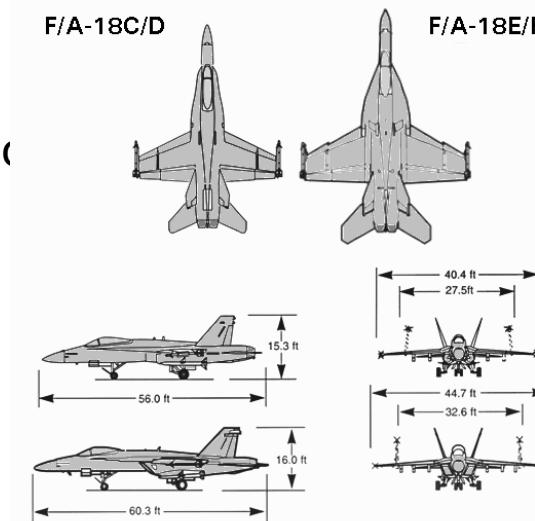
- NASA - Erdbeobachtungssatelliten 1979-1985
 - Ozonloch 7 Jahre (!) lang nicht erkannt
 - Ursache: Softwarefehler – Veränderung der Ozonschicht als Sensordrift durch automatische Nullpunkt Korrektur »herausgemittelt«
- ESA, Kourou, Franz. Guyana, 4. Juni 1996
 - Selbstzerstörung der Ariane 5 beim Jungfernflug 39 Sekunden nach dem Start
 - Ursache: Softwarefehler – Lageregelungssoftware aus Ariane 4 ohne Test gegen Start-Trajektorie der Ariane 5 wiederverwendet, dadurch Konvertierungsfehler
- Bemannte NASA-Raumkapsel Gemini V
 - Verfehlte ihren Landeplatz um ca. 160 Kilometer
 - Ursache: Softwarefehler – Rotation der Erde um die Sonne nicht berücksichtigt!





F-18 AM ÄQUATOR: DOKUMENTATIONSFEHLER

- US Air Force, Programm zur Raketensteuerung
 - Effizienz bei Flug einer Rakete über Äquator:
 - Koordinaten nicht neu berechnet, sondern nur Vorzeichen geändert
 - Effekt: Rakete drehte sich bei Äquator-Überflug um eigene Achse
 - störte niemanden!
 - Software in Autopilot des Jägers F-18 übernahm
 - Äquator-Überflug: Maschine dreht sich auf den Kopf!
 - Glück gehabt: bereits im Simulator bemerkt





T-MOBILE 2009: VIELE BETROFFENE ...



- 21. April 2009: Software-Fehler legt T-Mobile-Netz lahm
 - ab 16 Uhr Handys von Millionen T-Mobile-Kunden nicht erreichbar
 - Automatische Ansage:

"Dieser Anschluss ist aus technischen Gründen vorübergehend nicht erreichbar. Bitte rufen sie später wieder an."

- 19 Uhr: Techniker starten System neu
- 22 Uhr: Netz funktionierte wieder flächendeckend
- Ursache: **Softwarefehler** im Home Location Register (HLR). Dort sind Telefonnummern den SIM-Karten zugeordnet.



<http://www.spiegel.de/video/video-61903.html>



SOFTWAREFEHLER IN SYSTEMEN (FAHRZEUGEN)

Alles über Audi A4

Übersicht Tests News Erlkönige Gebrauchtwagen Usermeinungen Videos Neuwagen



Audi ruft 850.000 Autos zurück
Gefährlicher Airbag-Fehler im Audi A4

In diesem Artikel

1. Artikel

2. Bildergalerie

<http://www.auto-motor-und-sport.de/news/audi-ruft-850000-autos-zurueck-gefaehrlicher-airbag-fehler-im-audi-a4-743967.html>, abgerufen am 16.04.2015

Alles über Opel Insignia

Übersicht Tests News Erlkönige Gebrauchtwagen Usermeinungen Videos Neuwagen



Opel Insignia Rückruf
61.000 Fahrzeuge mit Softwarefehler

In diesem Artikel

1. Artikel

2. Bildergalerie

<http://www.auto-motor-und-sport.de/news/opel-insignia-rueckruf-61-000-fahrzeuge-mit-softwarefehler-7483237.html>, abgerufen am 16.04.2015



SOFTWAREFEHLER BEHINDERN UNS WEITERHIN ...

Fehler in der Software

Zehntausenden Studenten droht Bafög-Verspätung

Seit dem ersten August gibt es höhere Bafög-Sätze und Freibeträge. Doch eine gängige Bearbeitungs-Software kennt diese Neuerungen nicht. Deshalb könnten viele Studenten zum Semesterstart erst einmal ganz ohne Geld dastehen.

24.08.2016

f Teilen Twitter x Teilen E-mailen



<http://www.faz.net/aktuell/beruf-chance/campus/fehler-in-der-software-zehntausenden-studenten-droht-bafoga-verspaetung-14403775.html>, abgerufen am 12.03.2017

haft > Softwarefehler: Deutsche-Bank-Kunden kommen nur eingeschränkt an Geld

Softwarefehler

UPDATE 03.06.2016 18:20

Deutsche-Bank-Kunden kommen nur eingeschränkt an Geld

Nachdem bei der Deutschen Bank online fehlerhafte Doppelbuchungen aufgetreten sind, können Kunden vielfach keine Terminals nutzen. VON SARAH KRAMER



<http://www.tagesspiegel.de/wirtschaft/softwarefehler-deutsche-bank-kunden-kommen-nur-eingeschraenkt-an-geld/13685606.html>, abgerufen am 12.03.2017



... UND GEFÄHRDEN UNS

MEDIZINPRODUKTE

0

von VERÖFFENTLICHT 12. JANUAR 2017 · BEARBEITET 12. JANUAR 2017

Roche informiert: Softwarefehler in der Diabetes-Management-App Accu-Chek Connect

[Accu-Chek](#) [Diabetes](#) [Roche Diabetes Care Deutschland GmbH](#)

Die Roche Diabetes Care Deutschland GmbH informiert über einen Softwarefehler in der Diabetes-Management-App Accu-Chek Connect.

Wie das Unternehmen mitteilt, wurde ein Softwarefehler in der Diabetes-Management-App Accu-Chek Connect der Versionen 1.2.0 und 1.2.2 (iOS und Android) entdeckt.



Symbolabbildung

<https://www.produktwarnung.eu/2017/01/12/roche-informiert-softwarefehler-in-der-diabetes-management-app-accu-chek-connect/4749>, abgerufen am 12.03.2017



Modelle der Marken Chevrolet, Buick und Cadillac sind von dem Rückruf betroffen.

Freitag, 09. September 2016

Tödlicher Softwarefehler bei Airbags GM ruft 4,3 Millionen Autos zurück

Wiedereinmal machen die Airbags einem großen Autobauer Ärger. Mehr als vier Millionen Fahrzeuge muss GM in den USA zurückrufen. Der Fehler soll bereits einem Menschen das Leben gekostet haben.



Wegen Software-Problemen beordert die Opel-Mutter General Motors (GM) weltweit knapp 4,3 Millionen Fahrzeuge in die Werkstätten zurück. Bei Unfällen seien wegen des Defekts in seltenen Fällen die Airbags auf dem Fahrer- und Beifahrersitz nicht ausgelöst worden, teilte der in Detroit ansässige Konzern mit.

<http://www.n-tv.de/wirtschaft/GM-ruft-4-3-Millionen-Autos-zurueck-article18607516.html>,
abgerufen am 12.03.2017



WARUM TESTEN VON SOFTWARE IN AUTOMOTIVE?

Self-Driving Uber Car kills Pedestrian in Ariona



<https://www.youtube.com/watch?v=iWGhXof45zl>

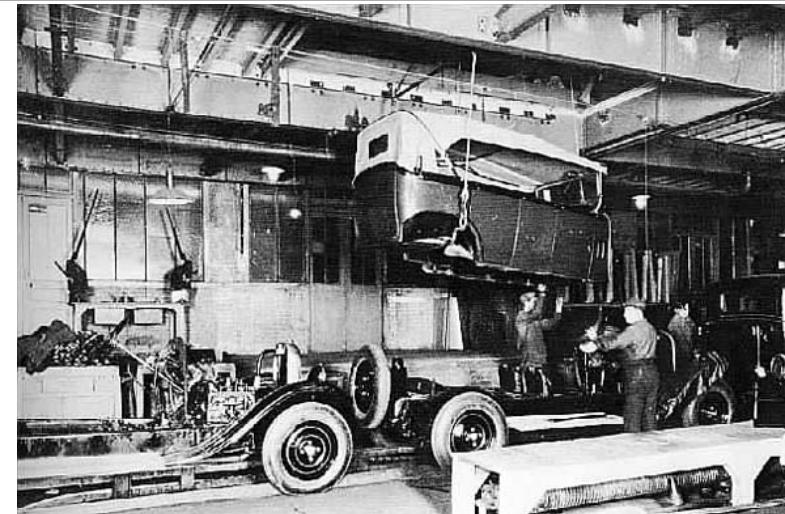


Quelle: <https://www.nytimes.com/2018/03/19/technology/uber-driverless-fatalty.html>, abgerufen 18.05.2018



WIE TESTEN ANDERE?

- Automobil-Industrie
 - Eingangstest der Komponenten (Komponententest)
 - Laufende Zwischenkontrollen am Fließband (Integrationstest)
 - realitätsnaher Einsatztest (Systemtest)
 - Probefahrt des Kunden (Abnahmetest)
 - Renneinsatz (Performanztest, Lasttest)
(Stabilität, Zuverlässigkeit, Robustheit)
 - Crashtest (Stresstest)





AUSTESTEN?

- einfaches Programm soll getestet werden
 - drei ganzzahlige Eingabewerte
- Datentyp der Eingabewerte: 16 Bit Integerzahlen
 - 2^{16} unterschiedliche Werte möglich
- drei unabhängige Eingabewerten:
 - $2^{16} * 2^{16} * 2^{16} = 2^{48}$ Kombinationen.
- Jede dieser Kombinationen ist zu testen.
- Wie lange dauert es bei 100.000 Testfälle pro Sekunde?
•



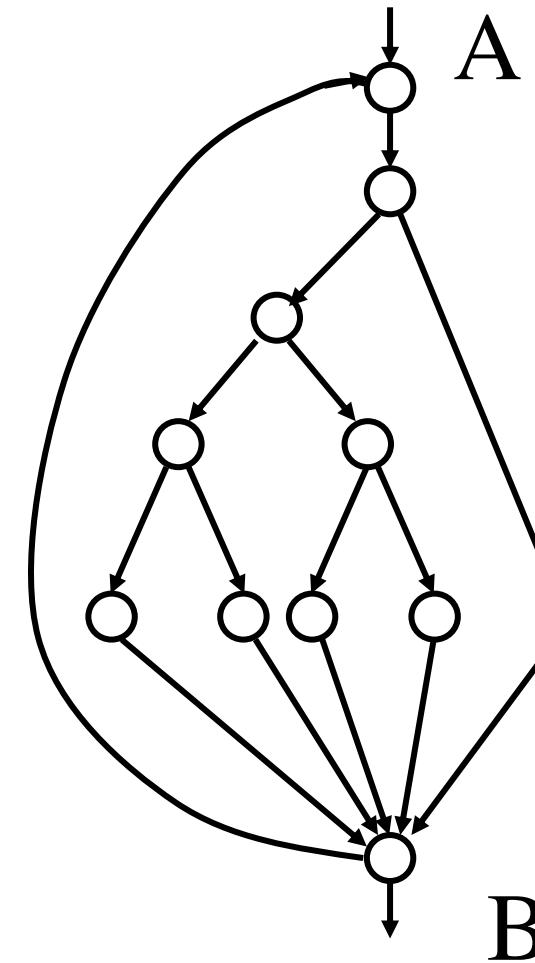
AUSTESTEN?

- einfaches Programm soll getestet werden
 - drei ganzzahlige Eingabewerte
- Datentyp der Eingabewerte: 16 Bit Integerzahlen
 - 2^{16} unterschiedliche Werte möglich
- drei unabhängige Eingabewerten:
 - $2^{16} * 2^{16} * 2^{16} = 2^{48}$ Kombinationen.
- Jede dieser Kombinationen ist zu testen.
- Wie lange dauert es bei 100.000 Testfälle pro Sekunde?
- Es sind 281.474.976.710.656 Testfälle
Dauer: ca. 90 Jahre



AUSTESTEN?

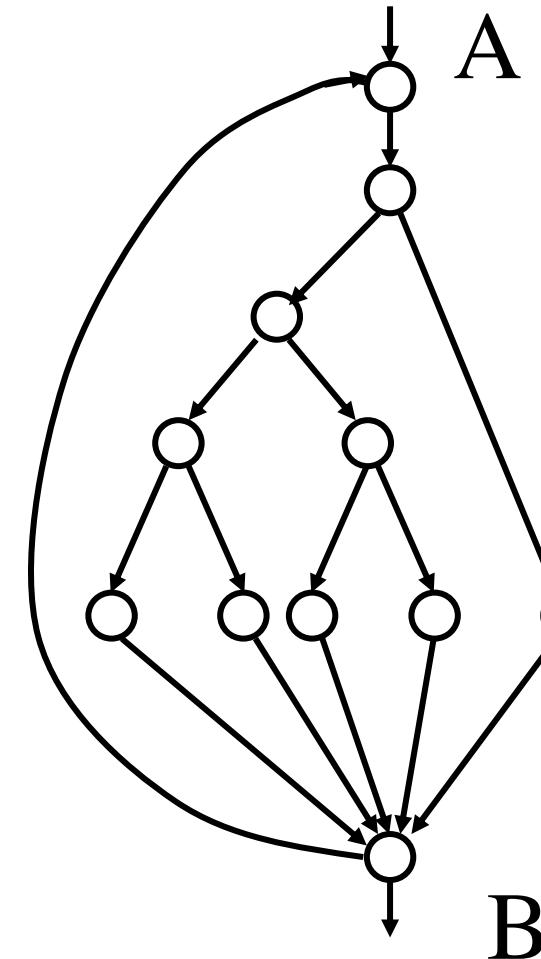
- Ein einfaches Programm soll getestet werden, das aus vier Verzweigungen (IF-Anweisungen) und einer umfassenden Schleife besteht und somit fünf mögliche Wege im Schleifenrumpf enthält.
- Unter der Annahme, dass die Verzweigungen voneinander unabhängig sind und bei einer Beschränkung der Schleifendurchläufe auf maximal 20, ergibt sich folgende Rechnung:
$$5^1 + 5^2 + \dots + 5^{18} + 5^{19} + 5^{20}$$
- Es sollen alle Pfade mit maximal 20 Schleifendurchläufen getestet werden.
- Wie lange dauert das Austesten bei 100.000 Testfällen pro Sekunde?
•





AUSTESTEN?

- Ein einfaches Programm soll getestet werden, das aus vier Verzweigungen (IF-Anweisungen) und einer umfassenden Schleife besteht und somit fünf mögliche Wege im Schleifenrumpf enthält.
- Unter der Annahme, dass die Verzweigungen voneinander unabhängig sind und bei einer Beschränkung der Schleifendurchläufe auf maximal 20, ergibt sich folgende Rechnung:
$$5^1 + 5^2 + \dots + 5^{18} + 5^{19} + 5^{20}$$
- Es sollen alle Pfade mit maximal 20 Schleifendurchläufen getestet werden.
- Wie lange dauert das Austesten bei 100.000 Testfällen pro Sekunde?
- Es sind 119.209.289.550.780 Testfälle
Dauer: ca. 38 Jahre

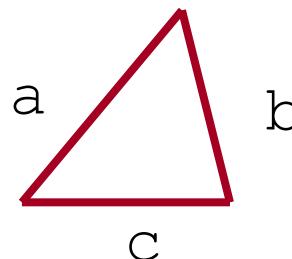




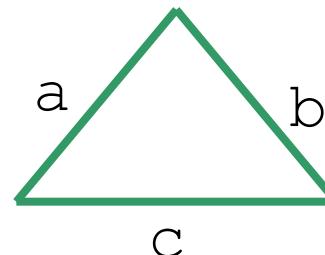
ERSTE ÜBUNG (ZUM WARM WERDEN!)

Ein Programm ist zu testen, das 3 ganzzahlige positive Werte einliest und als Längen eines Dreiecks interpretiert.

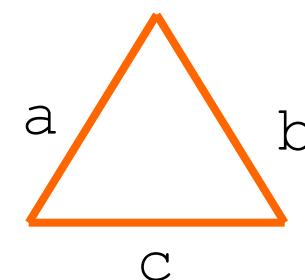
Das Programm gibt eine Meldung aus, ob es sich um ein **ungleichseitiges**, **gleichschenkliges** oder **gleichseitiges** Dreieck handelt.



$a \neq b \neq c$



$a = b \neq c$
 $a \neq b = c$
 $a = c \neq b$



$a = b = c$



IHRE VORSCHLÄGE BITTE

- Eine schnelle Antwort auf die Fragen:
 - Wie viele Tests werden benötigt?
 - Mit welchen Testdaten würden Sie das Programmstück testen?



MÖGLICHE TESTFÄLLE (1 VON 3)

Testfälle bestehen aus Testdaten und dem Soll-Ergebnis

1. 2,3,4 - zulässiges ungleichseitiges Dreieck
2. 2,2,2 - zulässiges gleichseitiges Dreieck
3. 2,2,1 - zulässiges gleichschenkliges Dreieck
- 4./5. 1,2,2 / 2,1,2 - Permutationen für gleichschenklige Dreiecke
6. 1,0,3 - kein Dreieck, eine Seitenangabe = 0
- 7./8. 0,1,3 / 1,3,0 - Permutationen
9. 5,-5,9 - kein Dreieck, eine Seitenangabe < 0
- 10./11. -5,5,9 / 5,9,-5 - Permutationen



MÖGLICHE TESTFÄLLE (2 VON 3)

12. 1,2,3 - kein Dreieck
Summe der beiden kürzeren Seiten = 3. Seitenlänge
- 13./14. 2,3,1 / 3,1,2 - Permutationen
15. 1,2,4 - kein Dreieck
Summe der beiden kürzeren Seiten < 3. Seitenlänge
- 16./17. 2,4,1 / 4,1,2 - Permutationen
- 18./19. 0,0,0 - kein Dreieck oder Fehlermeldung
alle Seiten = 0, zusätzlich 2 Seiten = 0 - Permutationen?
- 20.-22. Max_int, Max_int, Max_int - zulässiges gleichseitiges Dreieck
korrekte Dreiecksbestimmung beim Test mit maximalen Werten,
zusätzliche Tests mit 2 oder 1 maximalem Wert



MÖGLICHE TESTFÄLLE (3 VON 3)

- 23.-25. 1,1,4.4567 - Fehlermeldung »nicht ganzzahlige Werte«
Permutationen? - zusätzlich mit 2 oder 3 Werten
- 26.-28. 1,1,& - Fehlermeldung »Eingabe von Buchstaben oder Sonderzeichen«
Permutationen? - zusätzlich mit 2 oder 3 Werten
- 29./30. 1,2,3,4 / 2,3 - Fehlermeldung »falsche Anzahl von Werten«
(wenn Eingabe möglich)
31. Max_int/2 + 1, Max_int/2 + 1, Max_int/2 + 10 - zulässiges gleichschenkliges Dreieck
(Überlauf oder richtige Berechnung?
Bei $a \leq b \leq c$; Prüfung der Dreiecksbedingung mit $a+b > c$, führt $a+b$ zum Überlauf, s.a. Testfall 20)

Wie viele Tests hatten Sie überlegt?

in Anlehnung an
Glenford J. Myers:
Methodisches Testen von Programmen.
7. Auflage 2001

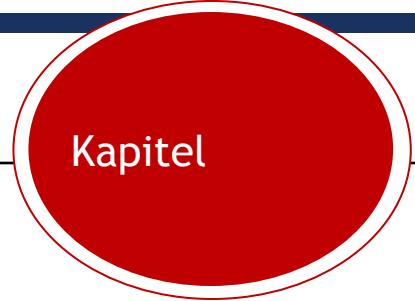


SPITZ-, STUMPF- UND RECHTWINKLIGE DREIECKE

- Werden spitz-, stumpf- und rechtwinklige Dreiecke zusätzlich berücksichtigt, ergeben sich insgesamt 47 Testfälle.

Resümee: Einfaches Problem, aber aufwendiger Test !

E. H. Riedemann:
Testmethoden für sequentielle und
nebenläufige Software-Systeme.
Teubner Verlag, Stuttgart 1997



EINLEITUNG



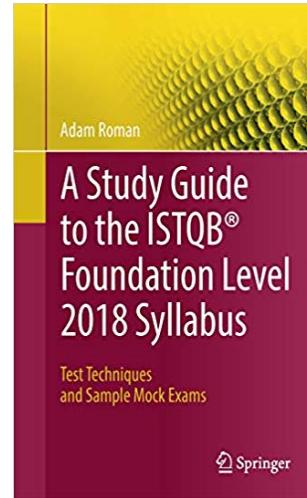
Certified Tester

Einstieg → Auswirkungen von Software-Fehlern

Anhang → Buchempfehlungen, Zeitschriften,
Organisationen, Tagungen

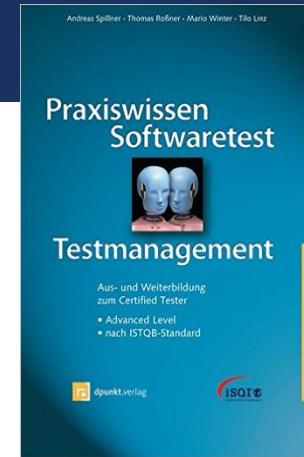
BÜCHER ZUR ERGÄNZUNG (LEHRPLAN 2018)

- Oberbörsch, Klaus: Softwaretesten nach ISTQB Standard; Kompakt+ (Lehrplan 2018); 3. Februar 2019; [Testen nach ISTQB Standard | Kompakt+: Auf Grundlage des aktuellen Lehrplans \(Syllabus\) von 2018 | ISBN: 9781727117424](#)
- Roman, Adam: **A Study Guide to the ISTQB® Foundation Level 2018 Syllabus: Test Techniques and Sample Mock Exams**; Springer, Auflage: 1st ed. 2018 (13. November 2018); **ISBN-10:** 3319987399



BÜCHER ZUR VERTIEFUNG

- Spillner, Andreas; Roßner, Thomas; Winter, Mario; Linz, Tilo:
Praxiswissen Softwaretest – Testmanagement
Aus- und Weiterbildung zum Certified Tester – Advanced Level
nach ISTQB-Standard.
dpunkt verlag, Heidelberg, Auflage: 4., überarb. u. erw. Aufl. (29. Mai 2014)



Ausführliche Beschreibung des Testmanagements – Vorbereitung auf die Certified Tester – Advanced Level – Testmanager Prüfung

- Liggesmeyer, Peter: **Software-Qualität**
Spektrum Akademischer Verlag; Auflage: 2 (18. Juni 2009)



Aktualisierung des Buches „Modultest und Modulverifikation: State of the Art“ von 1990, ergänzt um Kapitel zum Test von eingebetteten und objektorientierten Systemen.

BÜCHER ZUR VERTIEFUNG

- Westphal, Frank

Testgetriebene Entwicklung mit JUnit & FIT
Wie Software änderbar bleibt
dpunkt verlag, Auflage: 1 (10. Dezember 2012)

Inhalt: Unit Tests mit JUnit,
Testgetriebene Programmierung,
Refactoring, Häufige Integration,
Testfälle schreiben von A bis Z,
Isoliertes Testen durch Stub- und Mock-Objekte,
Akzeptanztests mit FIT (Framework for Integrated Test)



- Pol, Martin; Koomen, Tim; Spillner, Andreas:

Management und Optimierung
des Testprozesses.

dpunkt verlag, 2002 (2. Auflage)

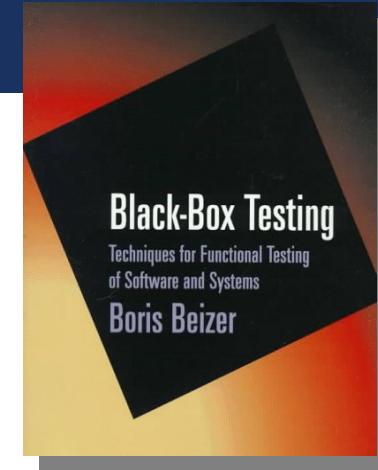
Ausführliche Beschreibung von TMap (Test Management Approach) und TPI (Test Process Improvement).
Gut geeignet zur Analyse des bestehenden Testprozesses und zur schrittweisen Verbesserung.



BÜCHER ZUR VERTIEFUNG

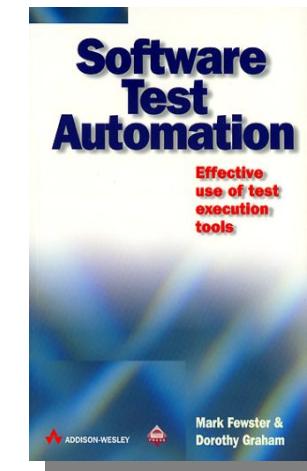
- Beizer, Boris:
Black-Box Testing –
Techniques for Functional Testing
of Software and Systems.
John Wiley & Sons, 1995

Ausführliche Beschreibung der Black-Box Verfahren.



- Fewster, Mark; Graham, Dorothy:
Software Test Automation: effective
use of test execution tools.
Addison Wesley, 1999

Ausführliche Beschreibung zur Struktur und
zum Aufbau der Testautomatisierung, mit guten
und schlechten Beispielen.



BÜCHER ZUR VERTIEFUNG

- Myers, Glenford J.:
Methodisches Testen von Programmen
Oldenbourg, 2005 (8. Auflage)

Klassischer Überblick. Für den Einstieg geeignet,
besonders zur Methodik der Testfallerstellung.
(Erstauflage „The Art of Software Testing“, 1979)



- Beizer, Boris
Software Testing Techniques
1990 (2. Auflage)

Ausführliche Behandlung vor allem
der White-Box-Testtechniken. Besonders
geeignet für frühe Testphase, weniger
für Systemtests.



BÜCHER ZUR VERTIEFUNG

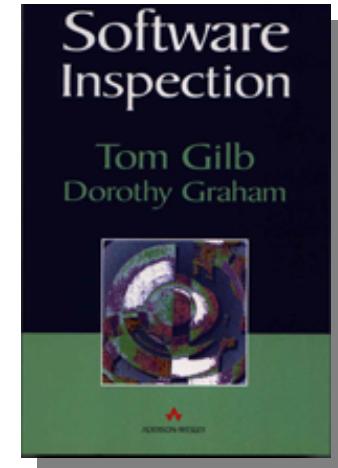
- Frühauf, Karol; Ludewig, Jochen;
Sandmayr, Helmut
Software-Prüfung
Vdf Hochschulverlag, 2006

Beschreibung von Reviews und
anderen Test- und Prüfverfahren.



- Gilb, Tom; Graham, Dorothy
Software Inspection
Addison-Wesley, 1993

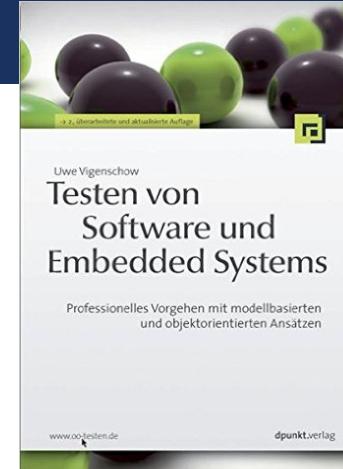
Ausführliche Beschreibung von Inspektionen.



BÜCHER ZUR VERTIEFUNG

- Vigenschow, Uwe
Testen von Software und Embedded Systems;
dpunkt Verlag, Auflage: 2., überarb. u. akt. Aufl. 2010

Beschreibung zum systematischen Vorgehen mit modellbasierten
und objektorientierten Ansätzen.



- Link, Johannes
Softwaretests mit JUnit.
Techniken der testgetriebenen Entwicklung
dpunkt Verlag; Auflage: 2 (10. Dezember 2012)

Ausführliche Beschreibung des Test-First-Ansatz,
allerdings ohne systematische Herleitung der Testfälle



BÜCHER ZUR VERTIEFUNG

- Sneed, Harry M.; Winter, Mario
Testen objektorientierter Software
Das Praxishandbuch für den
Test objektorientierter Client/Server Systeme
Carl Hanser Verlag, München, 2001

Guter Einstieg in den Test von OO-Software
Leider z.Zt. vergriffen



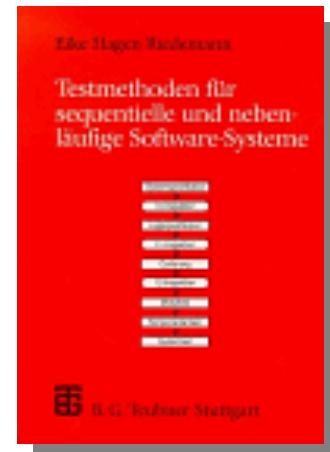
- Riedemann, Eike Hagen
Testmethoden für sequentielle und
nebenläufige Software-Systeme.
Teubner, Stuttgart, 512 S., 1997

Leider vergriffen
Gute Grundlagenbeschreibung mit Schwerpunkten
auf dem Test von nebenläufiger Software

vollständig als PDF-Dateien herunterladbar:
<http://ls10-www.cs.uni-dortmund.de/~riedemann/Homepage/PRUEF.html>

siehe auch:
<http://www-dssz.informatik.tu-cottbus.de/information/testen/zusammenfassung-testmethoden.pdf>

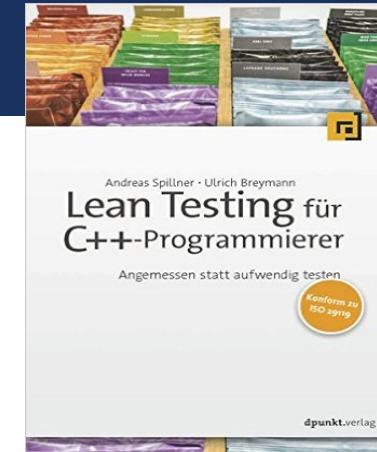
<http://www.abebooks.co.uk/9783519022749/Testmethoden-sequentielle-neben%C3%A4ufige-Software-Systeme-XLeit%C3%A4den-3519022745/plp>



BÜCHER ZUR ERGÄNZUNG

- Spillner, Andreas; Breymann, Ulrich:
Lean Testing für C++-Programmierer; Angemessen statt aufwendig testen; dpunkt.verlag GmbH; Auflage: I (2. Juni 2016);
ISBN-10: 3864903084.

Ausführliche Beschreibung des Testens auf Komponentenebene.



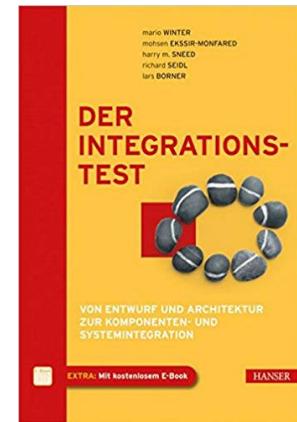
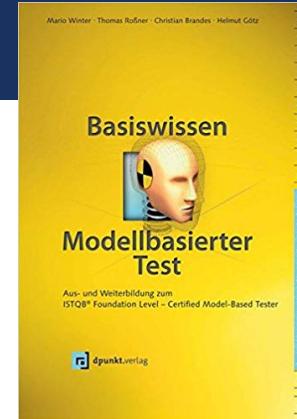
- Daigl, Matthias; Glunz, Rolf: **ISO 29119: Die Softwaretest-Normen verstehen und anwenden**; dpunkt.verlag GmbH; Auflage: I., Aufl. (28. Januar 2016).

Guter Zugang zur Softwaretestnorm ISO 29119



BÜCHER ZUR ERGÄNZUNG

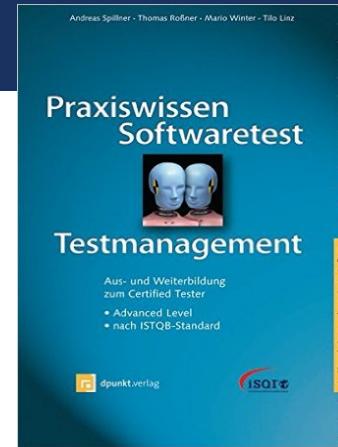
- Winter, M.; Roßner, T.; Brandes, C.; Götz, H.:
Basiswissen Modellbasierter Test; 2., akt. und erw. Aufl.,
dpunkt Verlag, Heidelberg, 2016.
- Winter, M.; Ekssir-Monfared,M.; Sneed, H.M.; Seidl, R.; Borner, L.:
Der Integrationstest – Von Entwurf und Architektur zur Komponenten- und Systemintegration; Carl Hanser Verlag,
München 2013.



BÜCHER ZUR ERGÄNZUNG

- Spillner, Andreas; Roßner, T.; Winter, M.; Linz, T.: **Praxiswissen Softwaretest – Testmanagement;** 4. überarb. u. erw. Aufl. (29. Mai 2014); dpunkt verlag.

Aus- und Weiterbildung zum Certified Tester – Advanced Level nach ISTQB Standard.



- Bath, Graham; McKay, Judy:: **Praxiswissen Softwaretest - Test Analyst und Technical Test Analyst:** Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB-Standard; 3., überarb. Auflage Februar 2015, dpunkt verlag.

Aus- und Weiterbildung zum Certified Tester - Advanced Level nach ISTQB Standard



ZEITSCHRIFTEN ZUM THEMENGEBIET

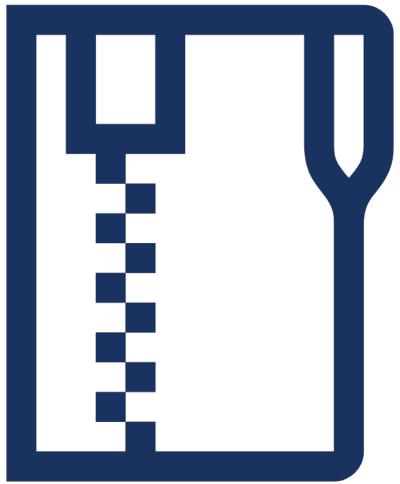
- Springer, Software Quality Journal:
<http://www.springer.com/computer/swe/journal/11219>
- ASQF, SQ-Magazin: <https://www.asqf.de/sq-magazin/>
- The Journal of Software Testing, Verification and Reliability,
John Wiley & Sons Ltd.
[http://onlinelibrary.wiley.com/journal/10.1002/\(ISSN\)1099-1689](http://onlinelibrary.wiley.com/journal/10.1002/(ISSN)1099-1689)
- Better Software
SQE Publications
<http://www.stickyminds.com/BetterSoftware/magazine.asp>
- Professional Tester, Test Publishing Ltd.
<http://www.professionaltester.com/>
- Softwaretechnik-Trends
(Mitteilungen der GI Fachgruppe TAV u.a.)
<http://pi.informatik.uni-siegen.de/stt/>



ORGANISATIONEN

- Gesellschaft für Informatik e.V. -
Fachgruppe TAV -
Test, Analyse und Verifikation von Software
(regelmäßige Treffen der Fachgruppe)
<http://fg-tav.gi.de/>
- ASQF - Arbeitskreis Software Qualität und Fortbildung e.V.
(Fachgruppentreffen und weitere Veranstaltungen)
<https://www.asqf.de/fachgruppen/software-test/>
- GTB - German Testing Board e.V.
(ISTQB Certified Tester Programm)
<http://www.german-testing-board.info>





GRUNDLAGEN DES SOFTWARETESTENS

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 / berechtigte 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*)

- Ein Ereignis in welchem eine Komponente oder ein System eine geforderte Funktion nicht im spezifizierten Rahmen ausführt. [ISO 24765, Glossar 3.2]
 - Wirkung eines Fehlerzustands, die nach »außen« sichtbar auftritt.
 - Abweichung zwischen erwartetem Ergebnis und Istergebnis.
 - 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

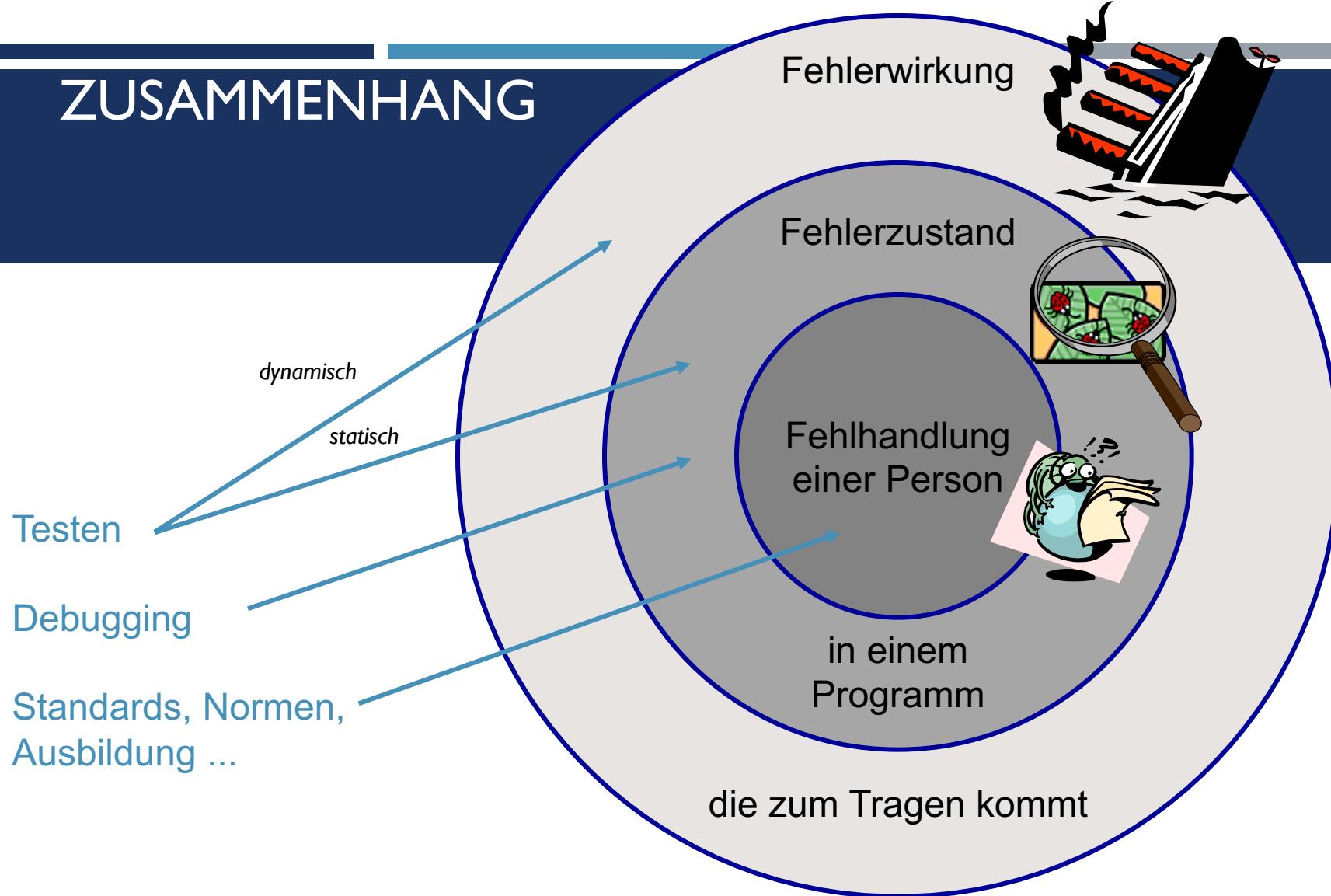
FEHLHANDLUNG - DEFINITION



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

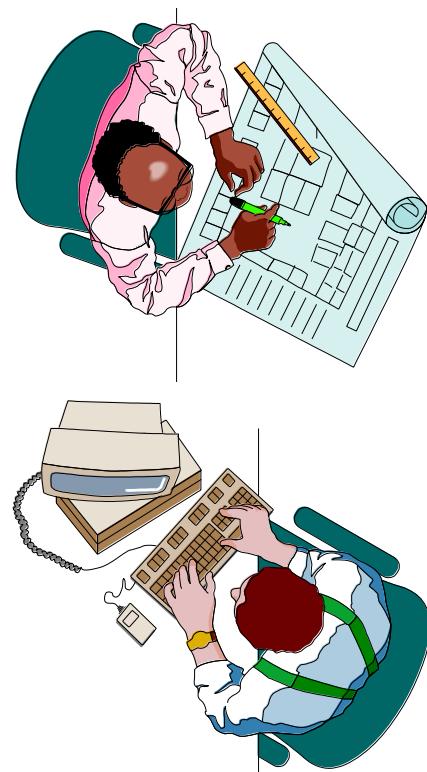
eines Softwareprodukts und dazugehöriger Arbeitsergebnisse befassen.

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



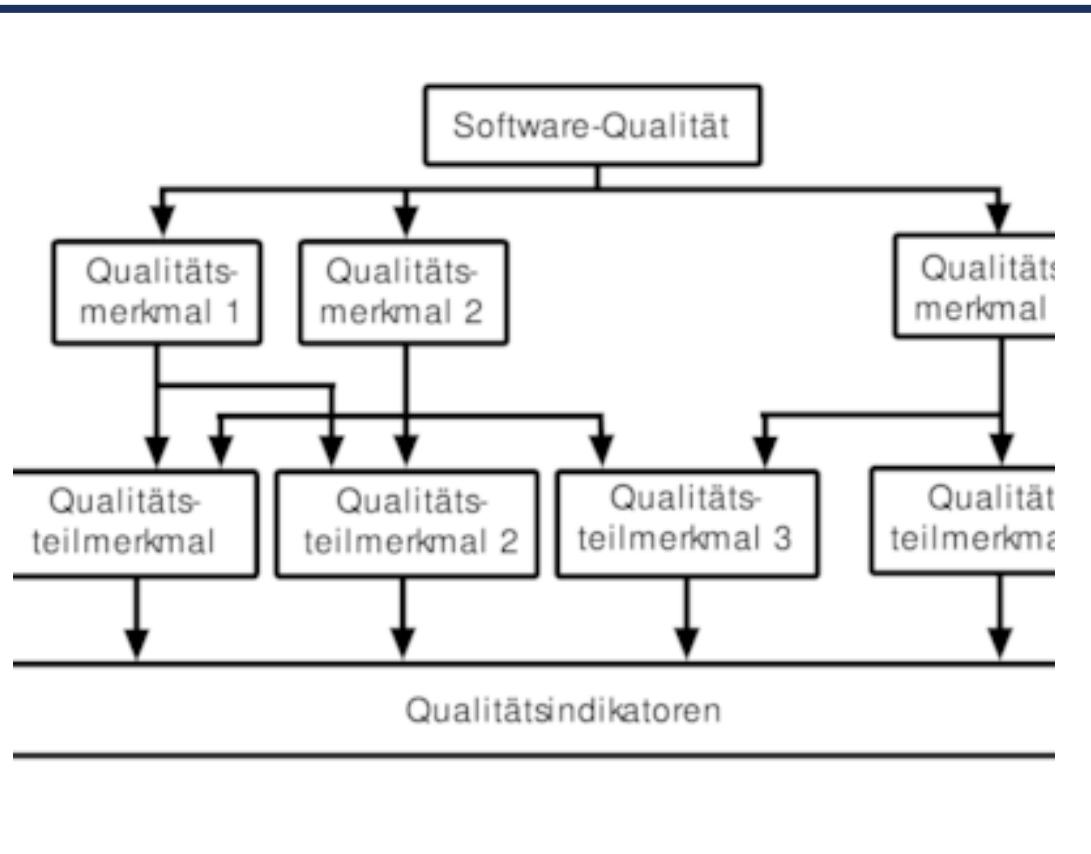
Validierung

- 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:

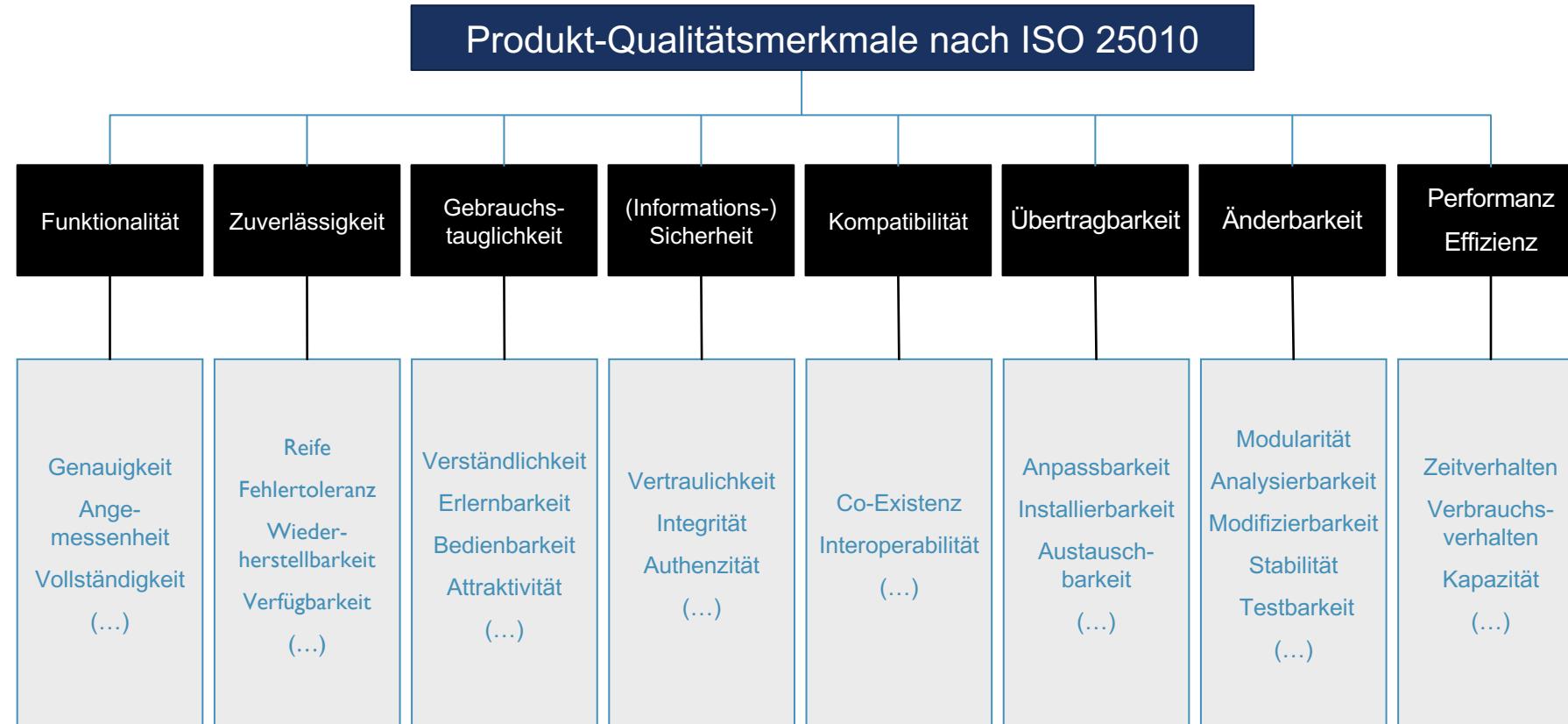
- Der Grad, in dem ein System, eine Komponente oder ein Prozess die Kundenerwartungen und -bedürfnisse erfüllt. [IEEE 610, Glossar 3.2]
- Der Grad, in dem ein Satz inhärenter Merkmale Anforderungen erfüllt. [ISO 9000:2000]

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



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

GRUNDLAGEN DES SOFTWARETEST ENS



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 (I)

- 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

GRUNDLAGEN DES SOFTWARETEST ENS



Begriffe und Motivation

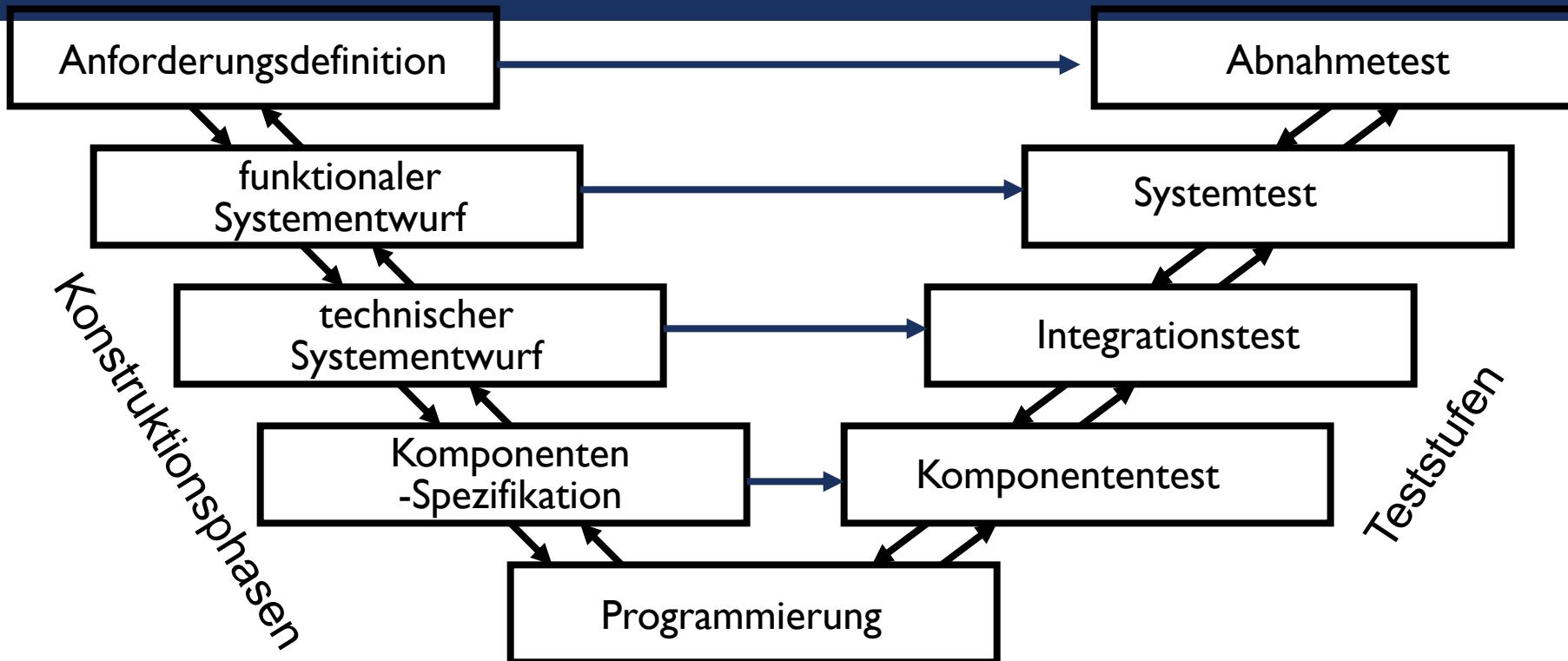
Grundsätze des Softwaretestens

Testprozess

Testfälle, Sollwerte und Testorakel

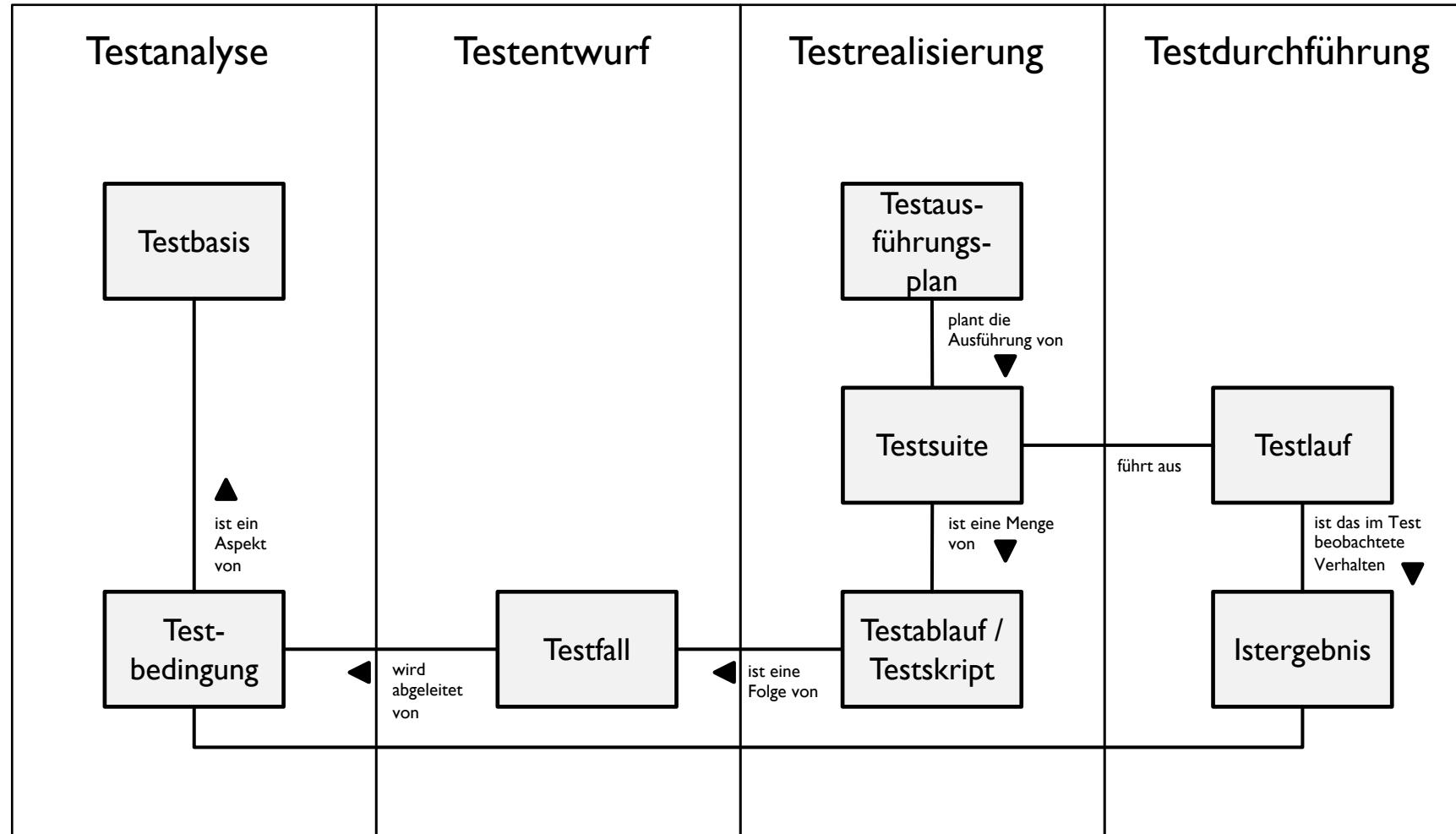
Psychologie des Testens

ALLGEMEINES V-MODELL



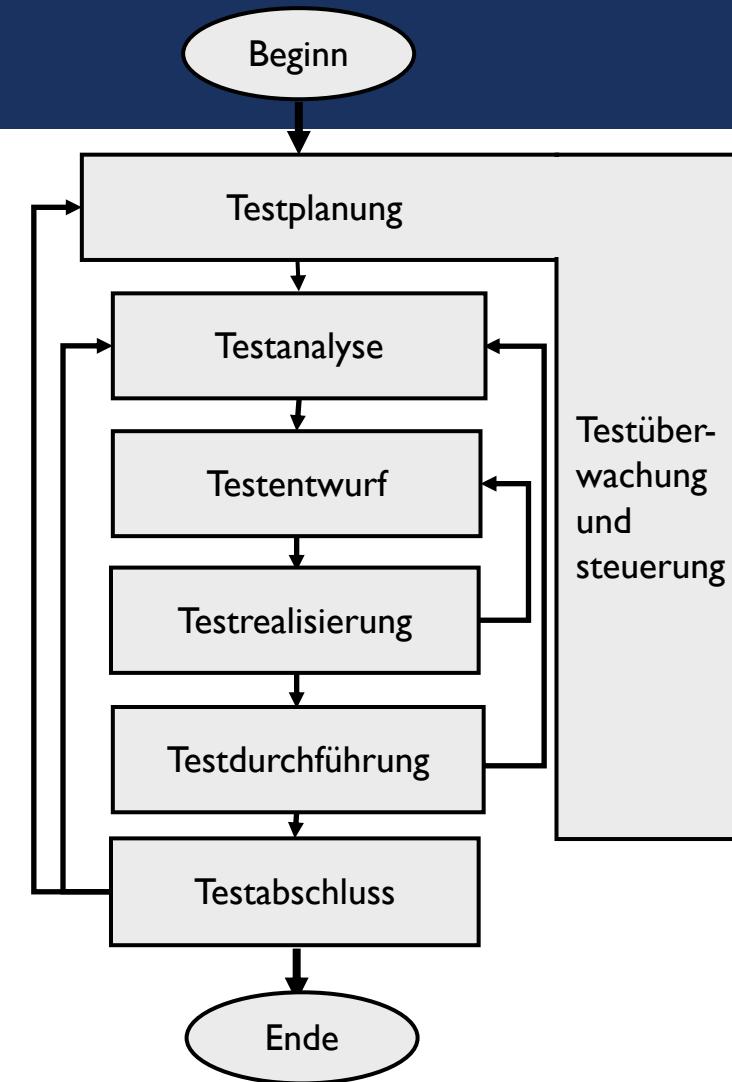
Legende
→ Testfälle basieren auf den entsprechenden Dokumenten

RÜCKVERFOLGBARKEIT

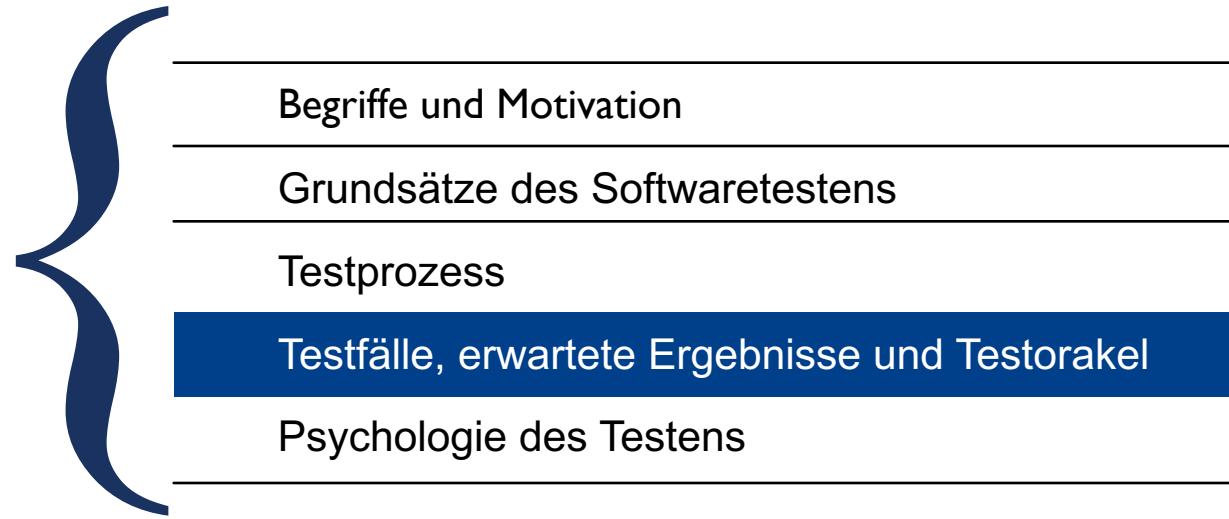


AKTIVITÄTEN DES TESTPROZESSES

- 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.



GRUNDLAGEN DES SOFTWARETEST ENS



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 ≤ 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 (Firmenzugehörigkeit)	$x \leq 3$	$3 < x \leq 5$	$5 < x \leq 8$	$x > 8$
Erwartetes Ergebnis (Gratifikation in %)	0	50	75	100

Konkreter Testfall	1	2	3	4
Eingabewert x (Firmenzugehörigkeit)	2	4	7	12
Erwartetes Ergebnis (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 <= 3 Jahre“ nicht spezifiziert wurde?