Willkommen zur Vorlesung

Methodische Grundlagen des Software-Engineering

im Sommersemester 2012

Prof. Dr. Jan Jürjens

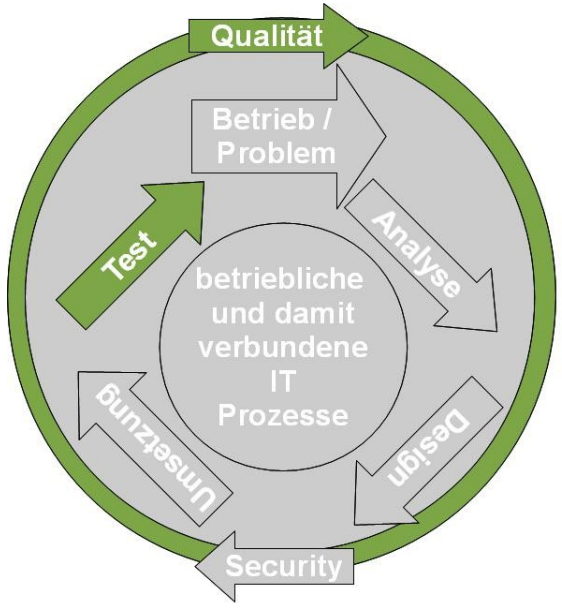
TU Dortmund, Fakultät Informatik, Lehrstuhl XIV

4.0 Testen: Einführung

Basierend auf dem Foliensatz
„Basiswissen Softwaretest - Certified Tester“
des „German Testing Board“
(nach Certified Tester Foundation Level Syllabus,
deutschsprachige Ausgabe, Version 2011)
(mit freundlicher Genehmigung)

Einordnung
Grundlagen Testen

- Geschäfts-Prozesse
- Qualitätsmanagement
- Testen
  - Einführung
  - Grundlagen Softwaretesten
  - Testen im Softwarelebenszyklus
  - Statischer Test
  - Black-Box-Test
  - White-Box-Test
  - Test-Management
  - Testwerkzeuge
  - Fuzzing
- Sicheres Software Design



Vorab Dank

- Ein großer Teil der Folien wurde übernommen aus einer gemeinsamen Hochschul-übergreifenden Lehrveranstaltung und vom GTB entsprechend des derzeitigen Syllabus 2011 aktualisiert.
- Dank gilt den Kollegen:
  - Dr. Falk Fraikin, TU Darmstadt
  - Dr. Eike Hagen Riedemann, ehemals TU Dortmund
  - Prof. Dr. Andreas Spillner, Hochschule Bremen
  - Prof. Dr. Mario Winter, FH Köln
- HS@GTB bezeichnet die Hochschul-Mitglieder des German Testing Boards (Stand 07.2011):
  - Dipl.-Inf. Timea Illes-Seifert, EnBW SIS GmBH
  - Dipl.-Inf. Horst Pohlmann, Hochschule Ostwestfalen-Lippe
  - Prof. Dr.-Ing. Ina Schieferdecker, FU Berlin
  - Prof. Dr. Mario Winter, FH Köln

German Testing Board
www.german-testing-board.info

aktuelle Informationen unter http://www.german-testing-board.info

Nach diesem Abschnitt sollten Sie...

- wissen, was Sie im weiteren Verlauf des Teils „Testen“ der Vorlesung erwartet;
- erläutern können, wie Software-Fehler Menschen, Maschinen, Unternehmen etc. Schaden zufügen können;
- begründen können, warum das vollständige „Austesten“ eines Programms in der Regel nicht möglich ist.

Ziele dieses Teils der Lehrveranstaltung

- Grundlagenvermittlung im Bereich Prüfen und Testen von Software
- Erklärung der Begriffe, Aufgaben und Tätigkeiten, Methoden und Testentwurfsverfahren
- Inhalt 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, ...)

Internationalität und Verbreitung (Stand 07.2011)

- 47 nationale Testing Boards auf allen 5 Kontinenten
- ca.174.000 Certified Tester weltweit
- ca. 25.000 GTB-autorisierte deutschsprachige Prüfungen

- aktuelle Informationen unter http://www.istqb.org bzw. http://www.german-testing-board.info

* American Software Testing Qualifications Board (ASTQB)
* Australia and New Zealand Testing Board (ANZTB)
* Austrian Testing Board (ATB)
* Bangladesh Software Testing Board (BSTB)
* Belarus Software Testing Qualifications Board (BySTQB)
* Belgium and Netherlands (BNTQB)
* Brazilian Software Testing Qualifications Board (BSTQB)
* Canadian Software Testing Board (CSTB)
* Chinese Software Testing Qualifications Board (CSTQB)
* Comité Français des Tests Logiciels (CFTL)
* Czech and Slovak Testing Board (CaSTB)
* Danish Software Testing Board (DSTB)
* Egyptian Software Testing Board (ESTB)
* Estonian Testing Board (ETB)
* Finnish Software Testing Board (FiSTB)
* **German Testing Board (GTB)**
* Gulf Software Testing Board (GSTB)
* Hispanic America Software Testing Qualifications Board (HASTQB)
* Hungarian Testing Board (HTB)
* Indian Testing Board (ITB)
* Iranian Testing Qualification Board (ITQB)
* Irish Software Testing Board (ISTB)
* Israeli Testing Certification Board (ITCB)
* Italian Software Testing Qualifications Board (ITA-STQB)
* Japan Software Testing Qualifications Board (JSTQB)
* Jordan Software Testing Qualifications Board (JOSTQB)
* Korean Testing Board (KTB)
* Latvian Software Testing Qualification Board (LSTQB)
* Luxembourg Testing Board (LTB)
* Malaysian Software Testing Qualifications Board (MSTB)
* Nigerian Software and Testing Board (NSTB)
* Norwegian Testing Board (NTB)
* Polish Testing Board (PTB)
* Russian Software Testing Qualifications Board (RSTQB)
* Saudi Arabia Testing Board (KSATB)
* Singapore Testing Qualifications Board (SGTQB)
* South African Software Testing Qualifications Board (SASTQB)
* South East European Testing Board (SEETB)
* Spanish Software Testing Qualifications Board (SSTQB)
* Sri-Lanka Testing Board
* Swedish Software Testing Board (SSTB)
* Swiss Testing Board (STB)
* Turkish Testing Board (TTB)
* Ukrainian Software Quality Board (USQB)
* UK Testing Board (UKTB)
* Vietnamese Testing Board (VTB)

Lehrbuch zum Certified Tester - Foundation Level

Andreas Spillner, Tilo Linz
Basiswissen Softwaretest
Aus- und Weiterbildung zum Certified Tester
Foundation Level nach ISTQB-Standard

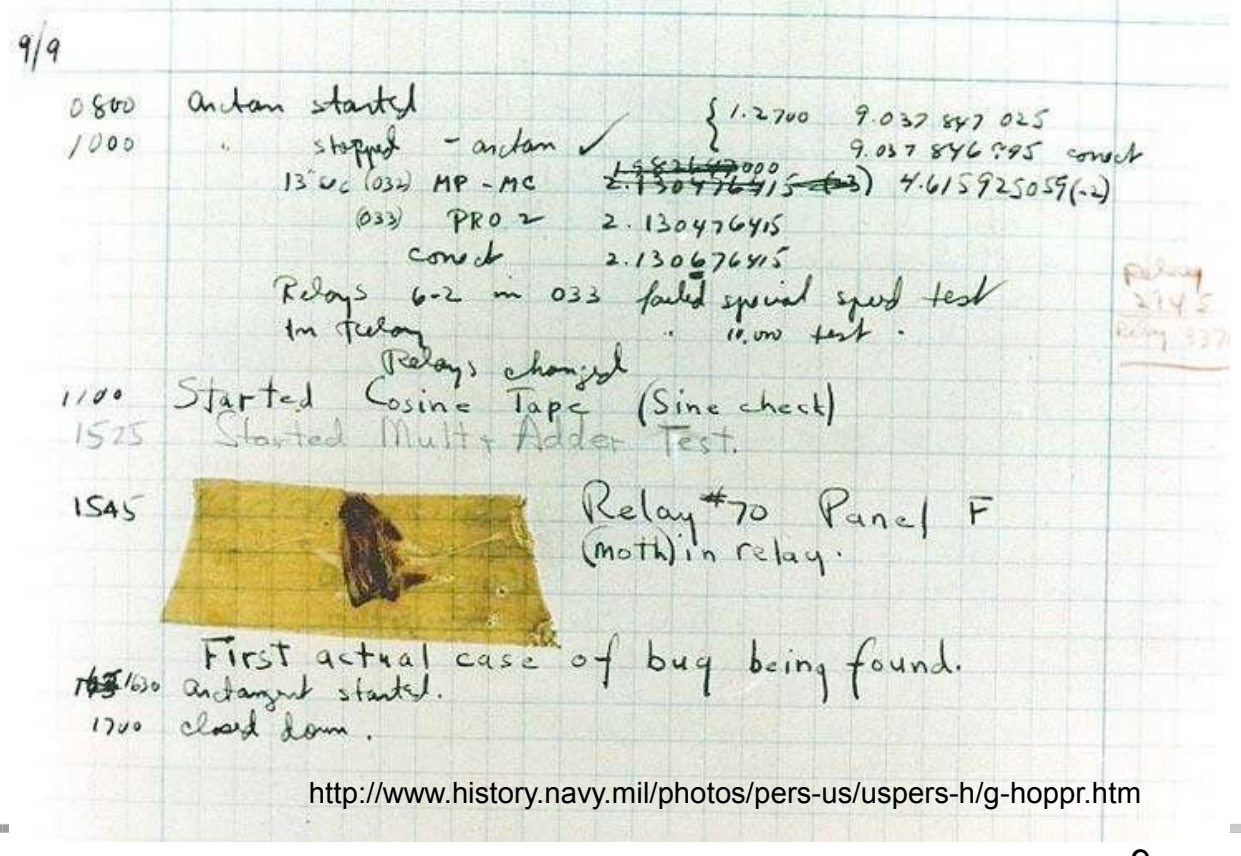
- 4., überarbeitete Auflage
- dpunkt.verlag, 2010, http://www.dpunkt.de
- 308 Seiten, Gebunden
- 39 Euro (D)
- ISBN 987-3-89864-642-0

Der erste Software-Fehler !

Eine Motte im Rechner Mark II verursacht Fehler in Relay Nr. 70, Panel F. Mrs. Grace Murray Hopper beseitigt den Fehler und dokumentiert ihn im Log-Buch.

»First actual case of bug being found.«

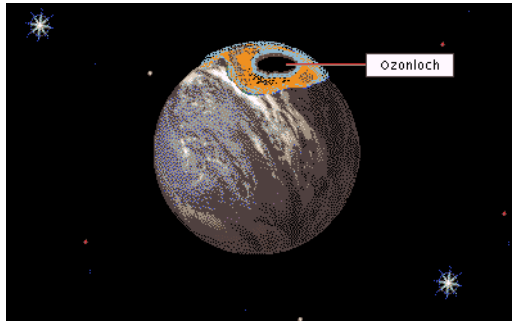
- »offen«-sichtlicher Fehler.
- Beseitigung ist einfach.



http://www.history.navy.mil/photos/pers-us/uspers-h/g-hoppr.htm

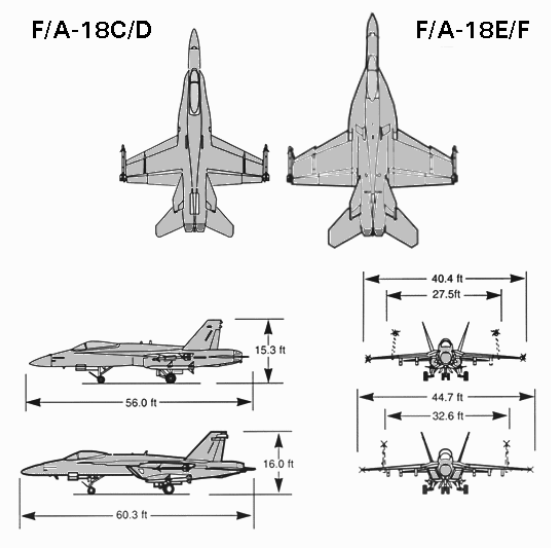
Auswirkungen von Software-Fehlern

- NASA - Erdbeobachtungssatelliten 1979-1985
  - Ozonloch 7 Jahre (!) lang nicht erkannt
  - Ursache: Softwarefehler – Veränderung der Ozonschicht als Sensordrift durch automatische Nullpunktkorrektur »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
  - Aus Speicherplatzmangel wurden beim Flug einer Rakete über den Äquator die Flugkoordinaten nicht neu berechnet, sondern nur das Vorzeichen geändert.
  - Dadurch drehte sich die Rakete beim Überflug des Äquator zwar um die eigene Achse, was allerdings niemanden störte.
  - Programm wurde unverändert in den Autopiloten des Jäger F-18 übernommen.
  - Bei Überflug des Äquator drehte sich die Maschine auf den Kopf !
  - Glück im Unglück: Man bemerkte dies bereits im Simulator ...



ALG2 LL (Harz IV) Software

September 2004: Drei Wochen Verspätung
Durch anhaltende Probleme mit der Computersoftware können die Arbeitsagenturen erst drei Wochen später als geplant die Antragsdaten für das Arbeitslosengeld II flächendeckend erfassen. Ursprünglich sollte das Programm schon ab 4. Oktober 2004 zur Verfügung stehen.
Mittlerweile haben sich die Bundesagentur für Arbeit (BA) und die Software-Entwickler der Telekom-Tochter T-Systems geeinigt: Das Programm soll stufenweise eingeführt werden. "Wir sind zurzeit in der Testphase, um Fehler zu ermitteln und zu beheben. Am 4. Oktober werden wir planmäßig starten und das System stufenweise hochfahren", sagte ein Sprecher von T-Systems im Gespräch mit **wdr.de** am Freitag (10.09.04). Das Computerprogramm werde Anfang Oktober in einigen Agenturen anlaufen, damit die BA Erfahrungen sammeln kann.

In einigen Agenturen wird das Programm getestet.

September 2005: Durch einen Fehler in der Software A2LL kommt es derzeit zu falschen Krankenkassen-Meldungen, teilt die Bundes-agentur für Arbeit (BA) mit. In mehreren hunderttausend Fällen seien Meldungen zur Krankenversicherung, also Anmeldungen, Abmeldungen, Veränderungsmitteilungen, von Arbeitslosengeld-II-Empfängern ohne Grund automatisch storniert worden.

http://www.heise.de/newsticker/meldung/62595

T-Mobile 2009: Viele Betroffene …

21. April 2009: Software-Fehler legt T-Mobile-Netz lahm

- Etwa ab 16 Uhr waren Millionen T-Mobile-Kunden mit ihren Handys nicht mehr erreichbar und konnten auch selbst niemanden mehr anrufen.
- Wer Geduld hatte und eine lange Stille nach dem Wählen der Nummer aushielt, konnte sich von einer kühlen Frauenstimme sagen lassen: **"Dieser Anschluss ist aus technischen Gründen vorübergehend nicht erreichbar. Bitte rufen sie später wieder an.“**
- T-Mobile-Sprecher Dirk Wende sagte, gegen 19 Uhr hätten Techniker das System zurückgesetzt und neu gestartet. Ab etwa 22 Uhr sollte das Netz wieder flächendeckend funktionieren.
- Ursache: **Softwarefehler** im Home Location Register (HLR)

  Dort werden die Telefonnummern den einzelnen SIM-Karten zugeordnet. Insgesamt drei Datenbanken waren betroffen und mussten nach und nach wieder verfügbar gemacht werden.

NASA - Mariner 1: Codierfehler

- Die NASA verliert die auf dem Weg zur Venus befindliche Raumsonde Mariner 1 am 22.6.1962:
  »Because of a launch-vehicle deviation from the planned flight path, Mariner R-1 was destroyed by the range safety officer after approximately 290 seconds of flight.«
- Korrekt codiert wäre gewesen:

```
DO 10 I=1,3      (Definition einer Schleife  in FORTRAN IV)
   …
10 CONTINUE
```

- Fehlerhaft codiert wurde

```
DO10I=1.3        (Zuweisung des Werts 1.3
   ...                an die Variable DO10I)
10 CONTINUE
```

NASA - Mariner 1: Codierfehler (2)

Nur ein FORTRAN-Problem? Nein!

Versehentliche nur 1-malige Ausführung mit i=4

C
```
for (i=1; i<=3; i++);
    f(i);
```

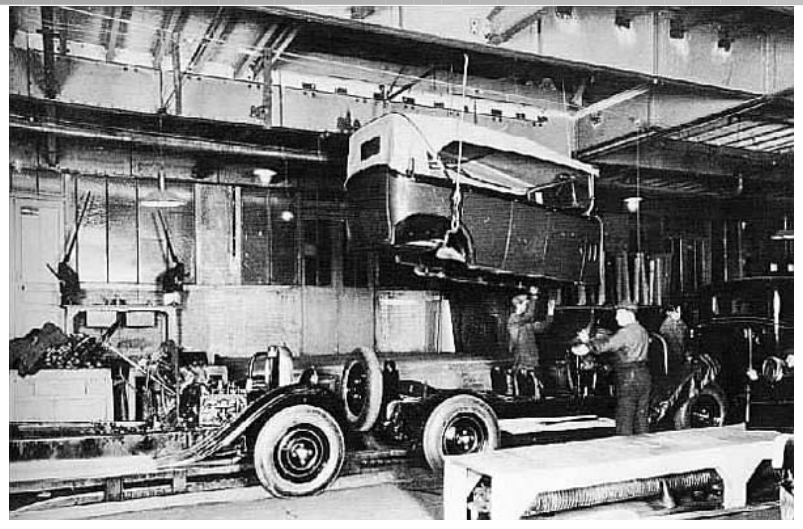
Java, C#
```
for (i=1; i<=3; i++){}
    f(i);
```

Perl
```
for ($i=1;$i<=3;$i++){}
   &f(i);
```

Ergänzung von H. Klaeren:
Probleme des Software-Engineering.
Informatik Spektrum (1994) 17: 21-28

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 ?

- Ein einfaches Programm soll getestet werden, das drei ganzzahlige Eingabewerte hat. Übrige Randbedingungen haben keinen Einfluss auf das Testobjekt.
- Jeder Eingabewert kann bei 16 Bit Integerzahlen 2^{16} unterschiedliche Werte annehmen.
- Bei drei unabhängigen Eingabewerten ergeben sich $2^{16} * 2^{16} * 2^{16} = 2^{48}$ Kombinationen.
- Jede dieser Kombinationen ist zu testen.
- Wie lange dauert es bei 100.000 Tests pro Sekunde ?

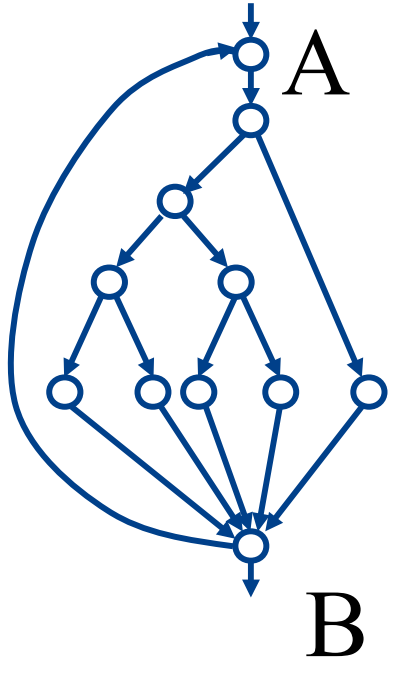
90 Minuten / 90 Stunden / 90 Tage / 90 Jahre / 90 Jahrzehnte ?

Austesten ?

- Ein einfaches Programm soll getestet werden, das drei ganzzahlige Eingabewerte hat. Übrige Randbedingungen haben keinen Einfluss auf das Testobjekt.
- Jeder Eingabewert kann bei 16 Bit Integerzahlen 2^{16} unterschiedliche Werte annehmen.
- Bei drei unabhängigen Eingabewerten ergeben sich $2^{16} * 2^{16} * 2^{16} = 2^{48}$ Kombinationen.
- Jede dieser Kombinationen ist zu testen.
- Wie lange dauert es bei 100.000 Tests pro Sekunde ?
- Es sind 281.474.976.710.656 Testfälle.
  Dauer: ca. 90 Jahre.

Austesten ?

Wir versuchen jetzt, die maximale denkbare Testmenge durch Verwendung interner Informationen über das Programm einzuschränken.



Beispiel: 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.

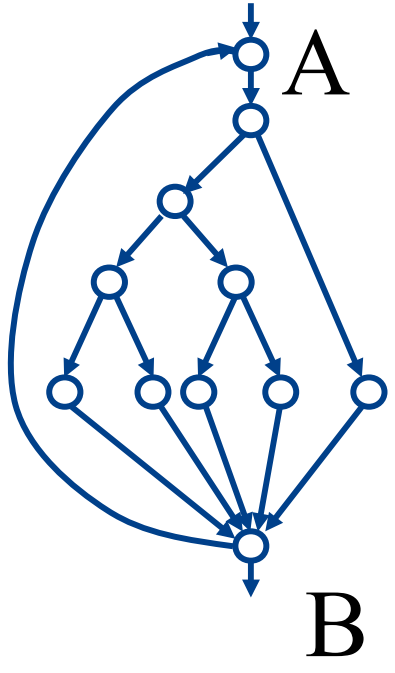
Unter der Annahme, dass die Verzweigungen voneinander unabhängig sind und bei einer Beschränkung der Schleifendurchläufe auf maximal 20, ergibt sich folgende Rechnung:

$$5^1 + 5^2 + \ldots + 5^{18} + 5^{19} + 5^{20}$$

- Wie lange dauert das Austesten bei 100.000 Tests pro Sekunde ?
- 38 Tage / 38 Wochen / 38 Monate / 38 Jahre ?

Austesten ?

Wir versuchen jetzt, die maximale denkbare Testmenge durch Verwendung interner Informationen über das Programm einzuschränken.



Beispiel: 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.

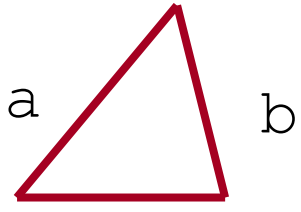
Unter der Annahme, dass die Verzweigungen voneinander unabhängig sind und bei einer Beschränkung der Schleifendurchläufe auf maximal 20, ergibt sich folgende Rechnung:

$$5^1 + 5^2 + \ldots + 5^{18} + 5^{19} + 5^{20}$$

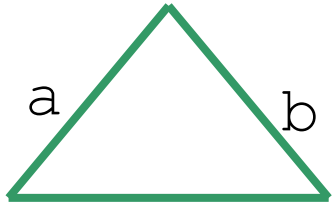
- Wie lange dauert das Austesten bei 100.000 Tests pro Sekunde ?
- Es sind 119.209.289.550.780 Testfälle.
  Dauer: ca. 38 Jahre.

Diskussion

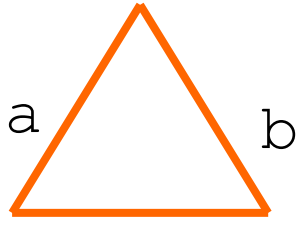
Um den Testaufwand zu begrenzen, kann man versuchen, die Fallunterscheidungen zu identifizieren, die ein Programm, das ein vorliegendes Problem implementiert, durchführen muss, um zumindest je einen Test pro Fallunterscheidung durchzuführen.
Beispiel: Ein Programm ist zu testen, das 3 ganzzahlige positive 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.



$a \neq b$ und
$b \neq c$ und
$a \neq c$



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



$a = b = c$

- Welche Anzahl Tests werden benötigt ?
- Mit welchen Testdaten würden Sie das Programmstück testen ?

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

Mögliche Testfälle

Testfälle bestehen aus Testdaten und dem Soll-Ergebnis:

Testnr.:	Testdaten (die drei Längen des Dreiecks):
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 2 weitere Testfälle mit 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
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
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<=b<=c; Prüfung der Dreiecksbedingung mit a+b>c, führt a+b zum Überlauf, s.a. Testfall 20)

Resümee: Einfaches Problem aber aufwendiger Test

Zusammenfassung

In diesem Abschnitt haben wir besprochen:

- wie Software-Fehler Menschen, Maschinen, Unternehmen etc. Schaden zufügen können;
- warum das vollständige „Austesten“ eines Programms in der Regel nicht möglich ist.