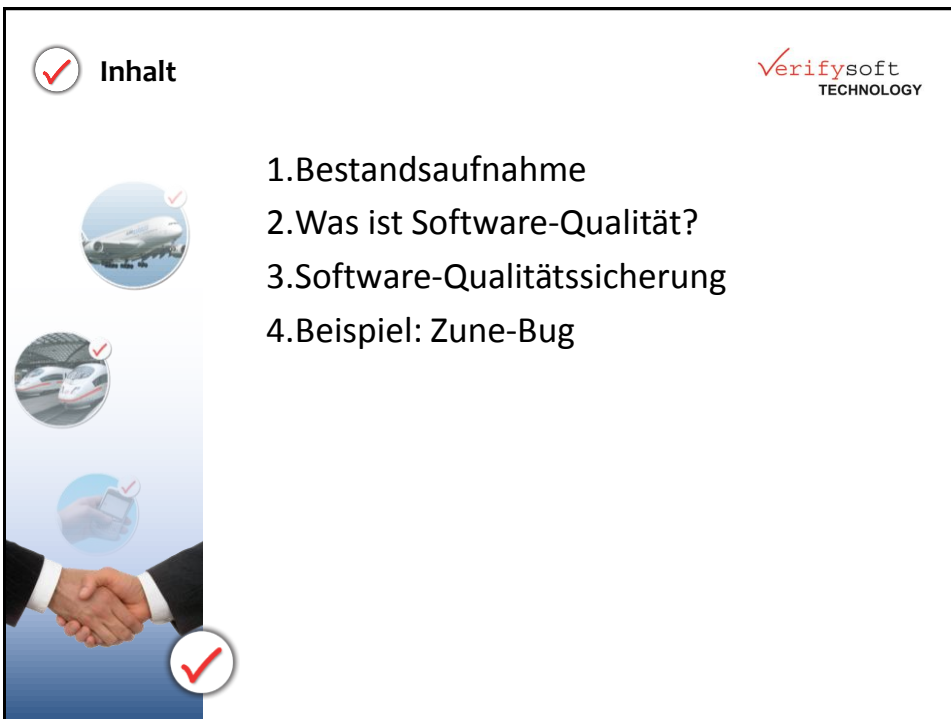




Verifysoft
TECHNOLOGY

Softwarequalitätssicherung

Prof. Dr. Daniel Fischer



✓ Inhalt

Verifysoft
TECHNOLOGY

1. Bestandsaufnahme
2. Was ist Software-Qualität?
3. Software-Qualitätssicherung
4. Beispiel: Zune-Bug



1. Bestandsaufnahme

Verifysoft
TECHNOLOGY

Olympia 2012: Software-Fehler schuld an Drama um Heidler-Wurf

...laut Olympia-Organisatoren einem Zufall und einem Software-Fehler geschuldet: "Es ist passiert, was bisher vermutlich noch nie passiert ist", sagte der Sprecher des Internationalen Olympischen Komitees (IOC), Mark Adams.

Heidlers fünfter Versuch sei mit 77,13 Metern auf den Zentimeter exakt so weit gewesen, wie der vierte Versuch der späteren Siegerin Tatjana Lysenko aus Russland. Das elektronische Computer-Messsystem habe die Annahme von Heidlers Weite daher verweigert. Die Anlage ist so programmiert, dass sie bei zwei exakt gleichen Messergebnissen hintereinander automatisch einen Fehler der Kampfrichter zugrunde legt.



Foto: Fotolia

<http://www.spiegel.de/sport/sonst/olympia-2012-software-fehler-schuld-an-heidler-drama-im-hammerwurf-a-849533.html>



1. Bestandsaufnahme

Verifysoft
TECHNOLOGY

F16 Kampfjet (1978)

Wiederverwendung von bestehender Software.

Eine **bestehende** Raketensteuerung wurde übernommen. Ein Fehler in dieser Software war für eine unbemannte Rakete unerheblich. Um Speicher zu sparen wurde beim Überflug über den Äquator ein Vorzeichen bei den Koordinaten geändert. Die Folge war, dass sich die Rakete dann auch um 180 Grad um die eigene Achse drehte. Dies wäre auch beim F18 Kampfjet so passiert.

Der Fehler wurde im **Simulator** entdeckt.

Gerade im militärischen und öffentlichen Bereich werden viele Softwarefehler publik gemacht.



Foto: Fotolia



1. Bestandsaufnahme

Verifysoft
TECHNOLOGY

Blackberry

Millionen Blackberry-Nutzer stundenlang ohne Mail-Zugriff
(11.10.2011)

Viele Nutzer konnten an diesem Tag nicht ihr Emailsystm auf dem Blackberry nutzen. Serverprobleme von RIM (Research in Motion – Hersteller des Blackberry) waren die Ursache.

Eine **verspätete Kommunikation** des Problems verärgerte zudem die Anwender!



Foto: Fotolia



1. Bestandsaufnahme

Verifysoft
TECHNOLOGY

USS Yorktown (21.09.1997)

Bei einem Manöver fiel für knapp drei Stunden die Steuerung aus. Das Schiff war manövrierunfähig.

Als Testschiff des Smart Ship Projects wurden ab 1996 27 Workstations mit Windows NT 4.0 eingerichtet und mit Glasfaserkabel vernetzt. Die Mannschaftszahl konnte verringert werden und auf der Brücke taten nur drei statt 13 Mann ihren Dienst.



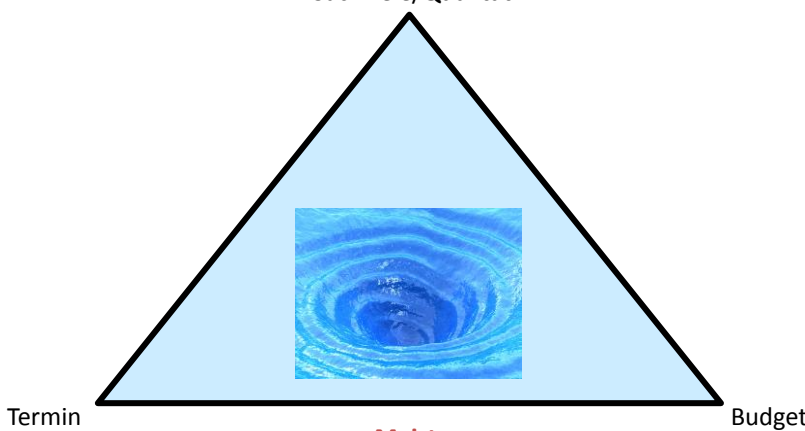
Foto: Fotolia

Ursache: Ein **Ventil wurde als offen angezeigt**, obwohl es geschlossen war. Ein Offizier änderte daraufhin einen Eintrag in einer Datenbank manuell auf 0, um den Fehler zu korrigieren. Die Software überprüft diese Änderungen nicht und später kam es dann zu einem „Divide by zero“-Error. Schuldzuweisungen: **Offizier – Betriebssystem – Anwendungsprogramm.**

✓ **1. Bestandsaufnahme** Verifysoft
TECHNOLOGY

Bermuda-Dreieck des Projektmanagements

Sachziele/Qualität



**Meist:
Take Two**

Termin Budget

Foto: Fotolia

✓ **2. Was ist Software-Qualität?** Verifysoft
TECHNOLOGY

Definition

Unter **Softwarequalität** versteht man die Gesamtheit der Merkmale und Merkmalswerte eines Softwareprodukts, die sich auf dessen Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen. (Balzert, Softwaretechnik, 1998)

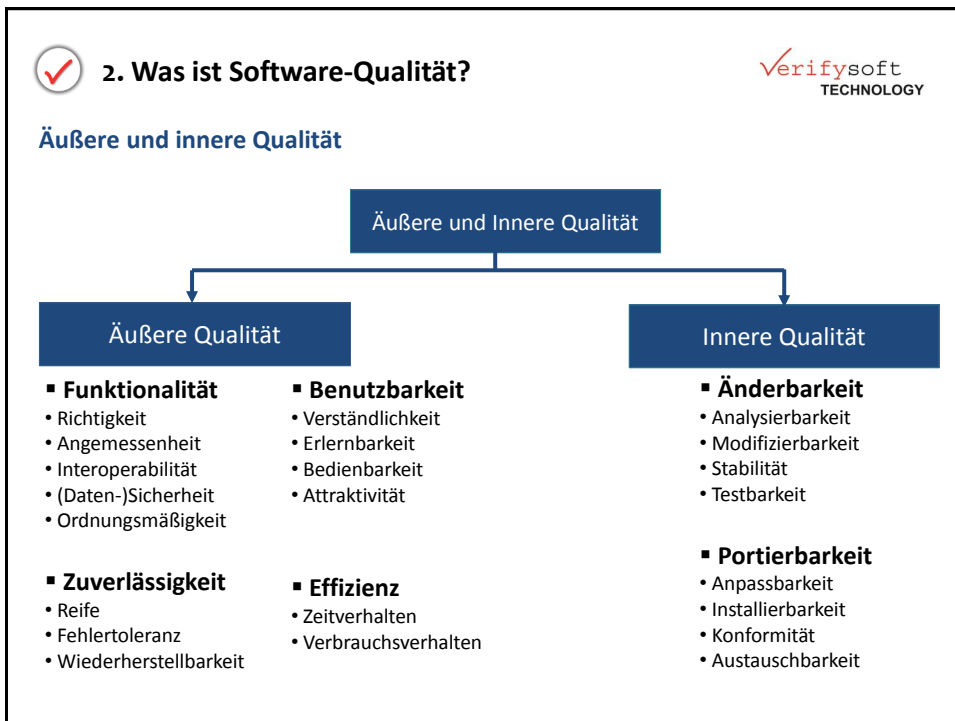
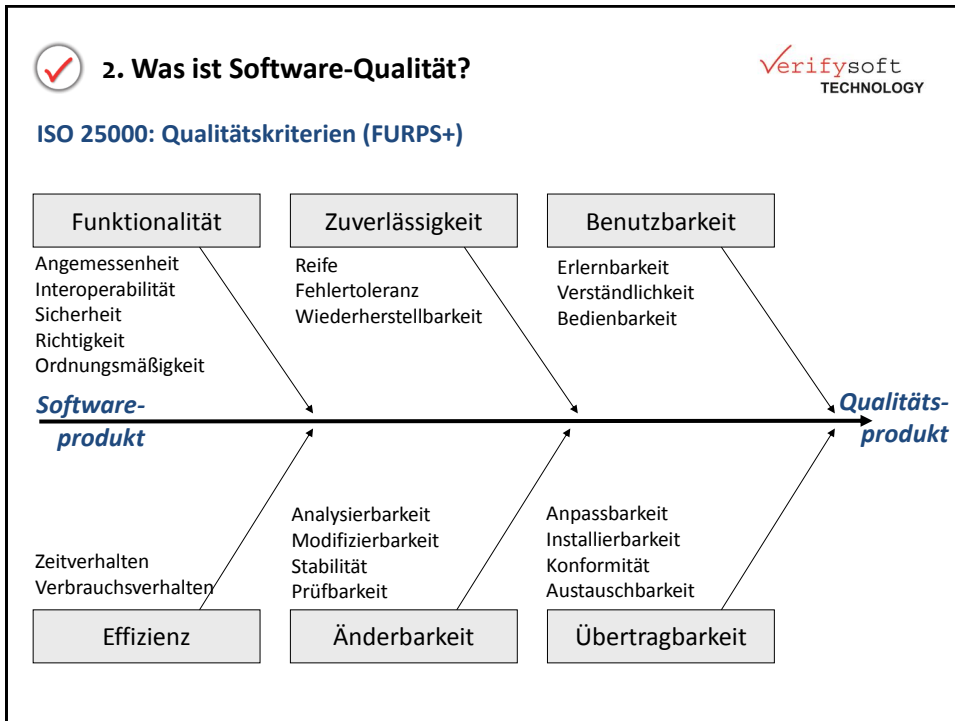
Softwarequalität

```

graph TD
    A[Softwarequalität] --> B[Gebrauchsqualität]
    A --> C[Äußere und Innere Qualität]
    
```

Gebrauchsqualität	Qualitätsmerkmale	Äußere und Innere Qualität
<ul style="list-style-type: none"> ▪ Effektivität ▪ Produktivität ▪ Sicherheit ▪ Zufriedenheit 		<ul style="list-style-type: none"> ▪ Funktionalität ▪ Zuverlässigkeit ▪ Benutzbarkeit ▪ Effizienz ▪ Änderbarkeit ▪ Portierbarkeit

ISO/IEC 9126 abgelöst seit 2005 durch ISO/IEC 25000

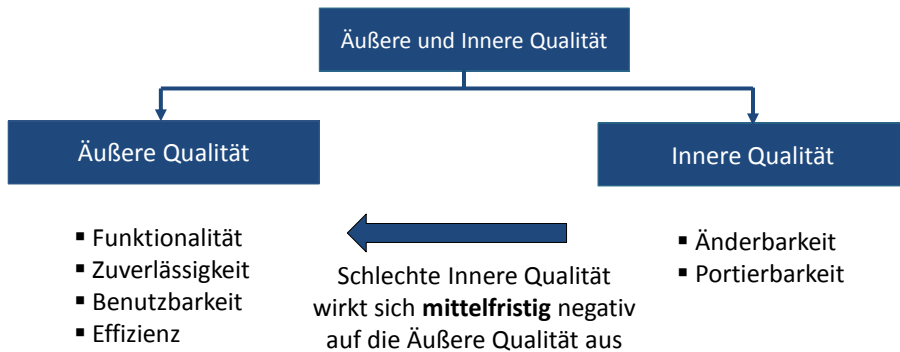




2. Was ist Software-Qualität?

Verifysoft
TECHNOLOGY

Äußere und innere Qualität



Ca. 80% der Zeit im Softwarelebenszyklus befindet man sich in der Phase Support und Wartung. Hier macht sich eine schlechte innere Qualität dann bemerkbar!

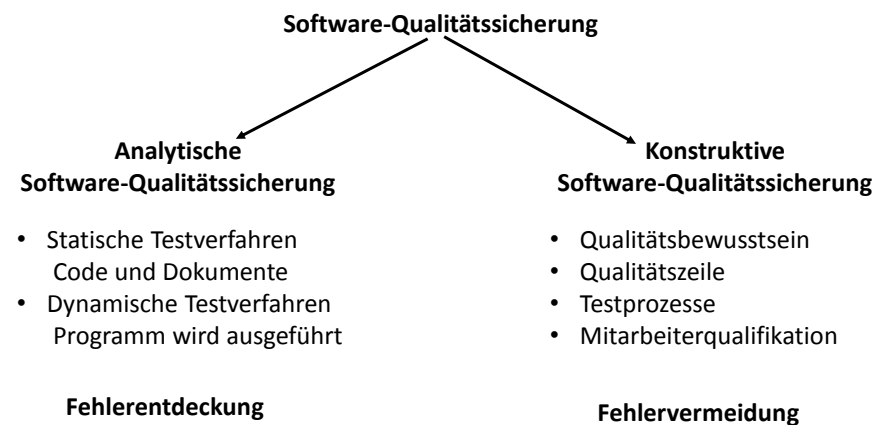
- Bis ca. 25 % aller Bugfixes führen zu neuen z.T. schwerwiegenden Bugs
- Ca. 39 % aller Concurrency Bugfixes sind fehlerhaft!

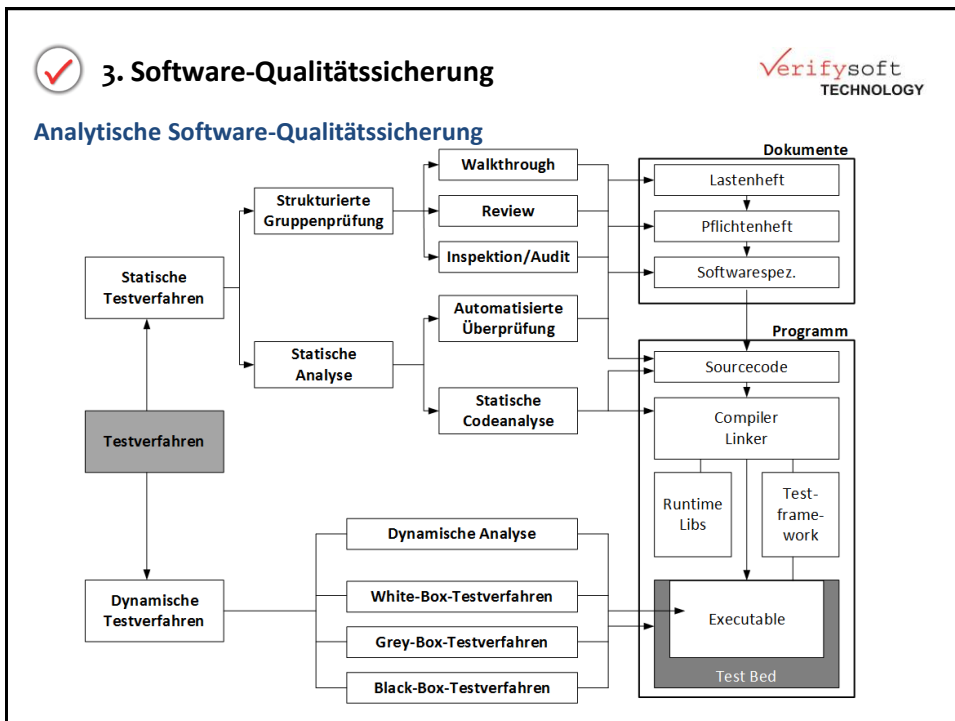
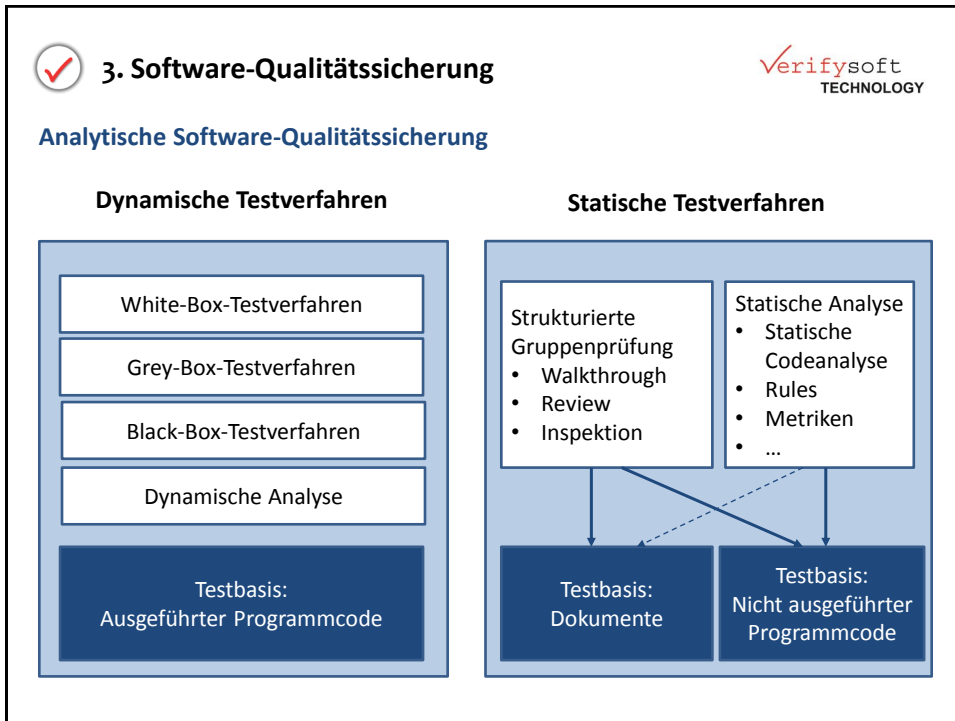


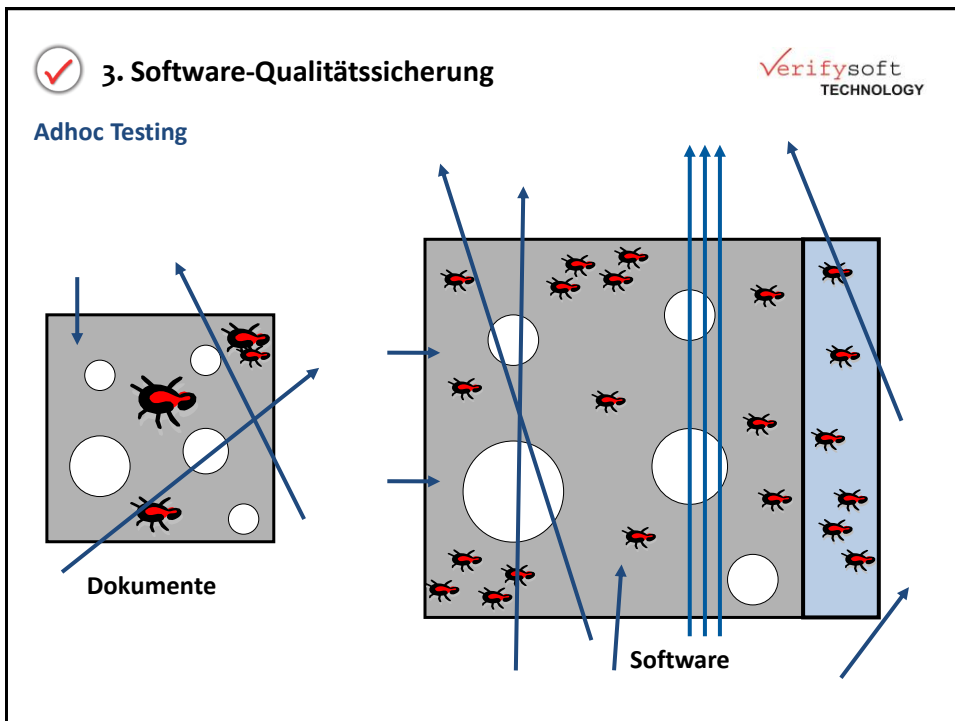
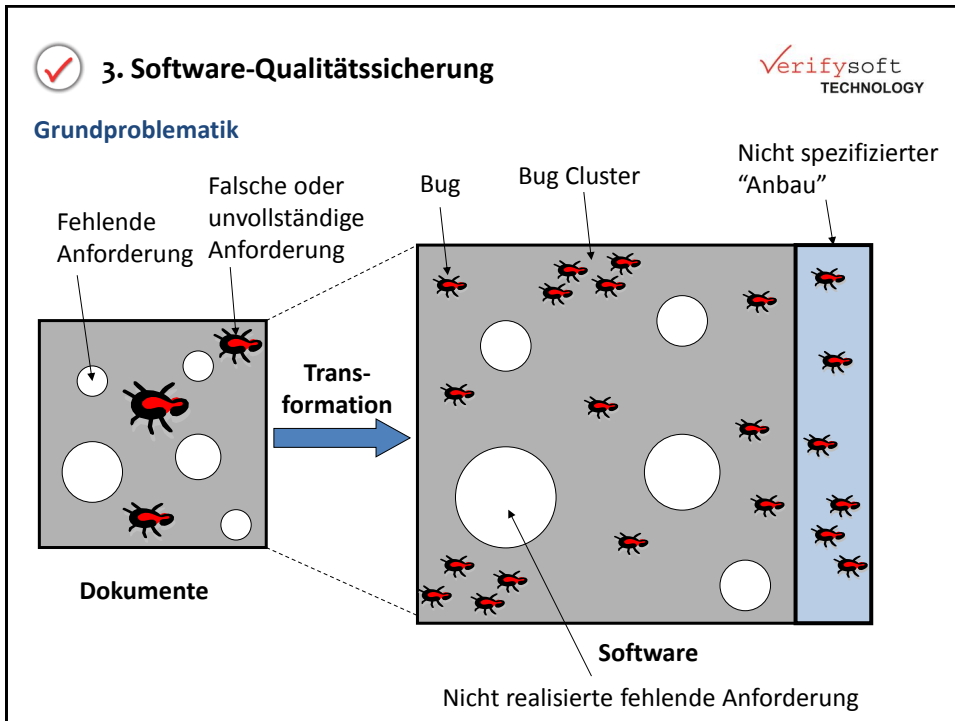
3. Software-Qualitätssicherung

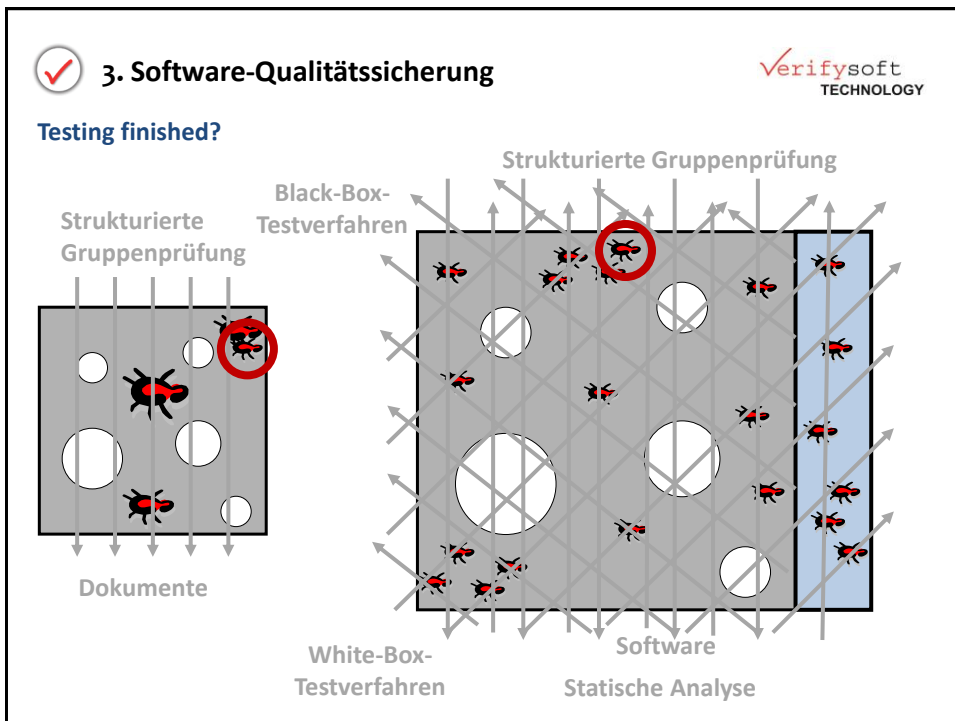
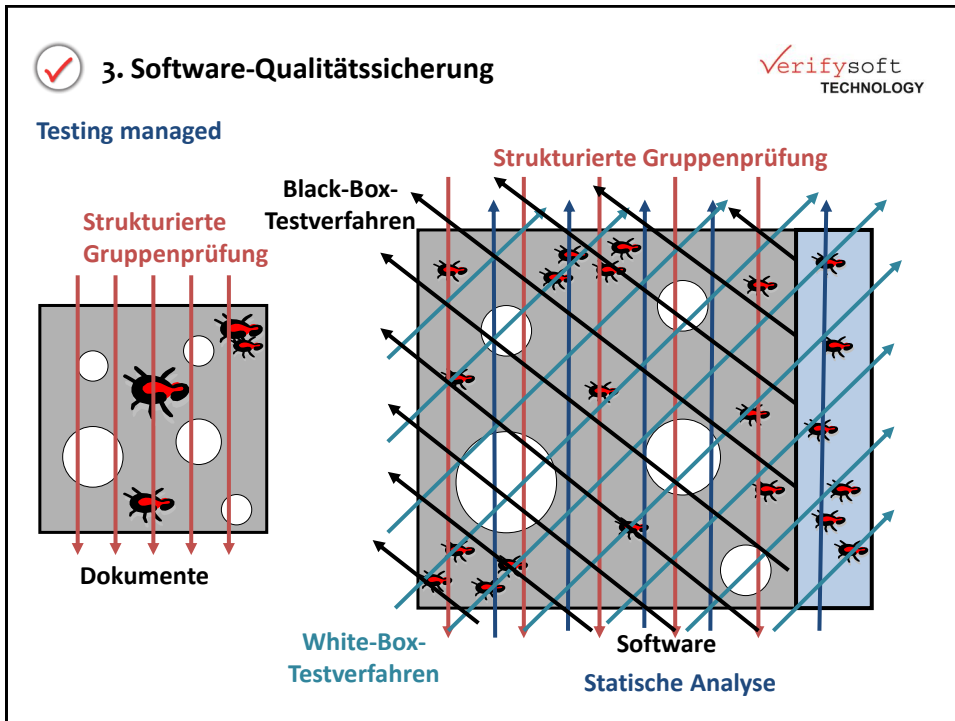
Verifysoft
TECHNOLOGY

Analytisch versus konstruktiv











4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

Microsoft MP3-Player



Foto: Fotolia

„Modelle der ersten Generation mit 30 GB-Festplatte stürzten am 31. Dezember 2008 sämtlich ab und ließen sich nicht neu starten. Der Hersteller gab ein Problem mit dem Schaltjahr 2008 als Grund für die Fehlfunktion an. Dieser Fehler erregte weltweite mediale Aufmerksamkeit.“

<http://de.wikipedia.org/wiki/Zune#Kritik>

The Zune's **real-time clock** stores the time in terms of days and seconds since January 1st, 1980. When the Zune's clock is accessed, the **driver** turns the number of days into years/months/days and the number of seconds into hours/minutes/seconds. Likewise, when the clock is set, the driver does the opposite.

The Zune frontend first accesses the clock toward the end of the boot sequence. Doing this triggers the code that reads the clock and converts it to a date and time.
<http://makezine.com/2008/12/31/cause-of-zune-leapyear-problem-free/>



4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

```
static void SetYearAndDayOfYear (struct RtcTime *time)
{
int days = time->daysSince1980;
int year = STARTING_YEAR;
while (days >365)
{
if (IsLeapYear(year))
{
if (days > 366)
{
days -= 366;
year += 1;
}
}
else
{
days -= 365;
year += 1;
}
}
time->dayOfYear = days;
time->year = year;
}
```

Ursprünglicher Fehler
Markteinführung war 2005

Am 31.12.2008 „stürzten“
alle Zune ab!
time->daysSince1980 ist
dann 10593

IsLeapYear() ist korrekt

```
int IsLeapYear(int year)
{
return !(year % 100 ?
year % 4 : year % 400);
}
```



4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

```
static void SetYearAndDayOfYear3 (struct RtcTime *time)
{
int days = time->daysSince1980;
int year = STARTING_YEAR;

while (days > 365)
{
if (IsLeapYear(year))
{
if (days > 366)
{
days -= 366;
year += 1;
}
else
{
if (days == 366)
{
days -= 366;
}
}
}
else
{
days -= 365;
year += 1;
}
}
time->dayOfYear = days;
time->year = year;
}
```

**Vorgeschlagener
fehlerhafter Bugfix**
dayOfYear war dann 0



4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

```
static void SetYearAndDayOfYear1 (struct RtcTime *time)
{
int days = time->daysSince1980;
int year = STARTING_YEAR;
while (days >365)
{
if (IsLeapYear(year))
{
if (days > 366)
{
days -= 366;
year += 1;
}
else
{
break;
}
}
else
{
days -= 365;
year += 1;
}
}
time->dayOfYear = days;
time->year = year;
}
```

Bugfix 1
Nicht strukturiert
programmiert



4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

```

static void SetYearAndDayOfYear2 (struct RtcTime *time)
{
int days = time->daysSince1980;
int year = STARTING_YEAR;
int daysInYear;

while(1)
{
    daysInYear = IsLeapYear(year) ? 366 : 365;

    if(days <= daysInYear)
        break;

    days -= daysInYear;
    year += 1;
}

time->dayOfYear = days;
time->year = year;
}

```

Bugfix 2

Nicht strukturiert
programmiert und
Endlos-while



4. Beispiel Zune-Bug

Verifysoft
TECHNOLOGY

```

static void SetYearAndDayOfYear4 (struct RtcTime *time)
{
int days = time->daysSince1980;
int year = STARTING_YEAR;
int daysInYear;

daysInYear = IsLeapYear(year) ? 366 : 365;

while(days > daysInYear)
{
    days -= daysInYear;
    year += 1;

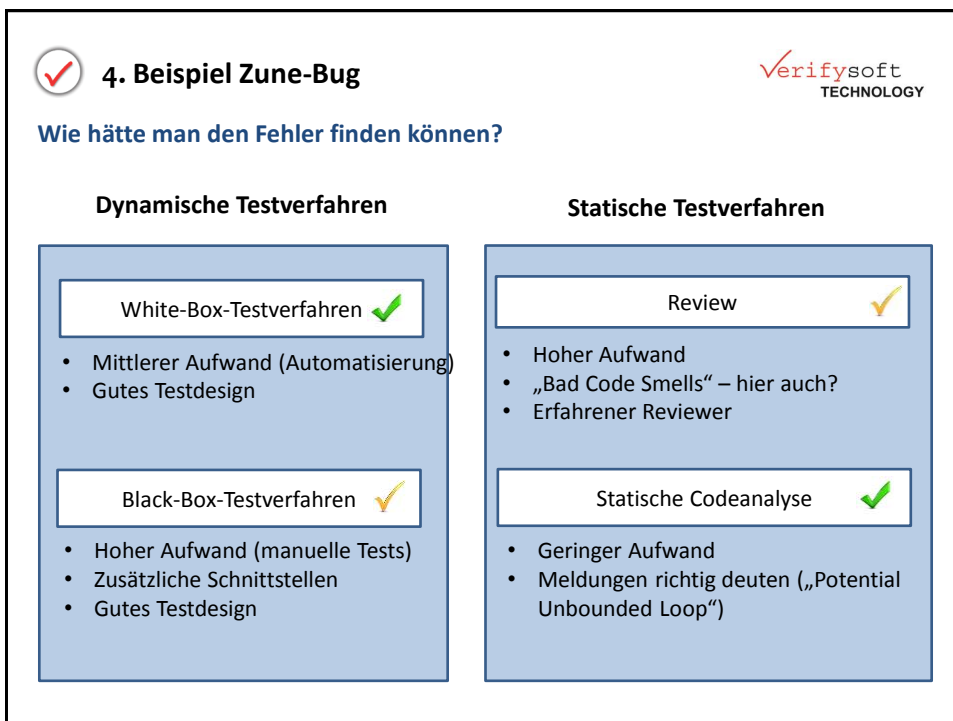
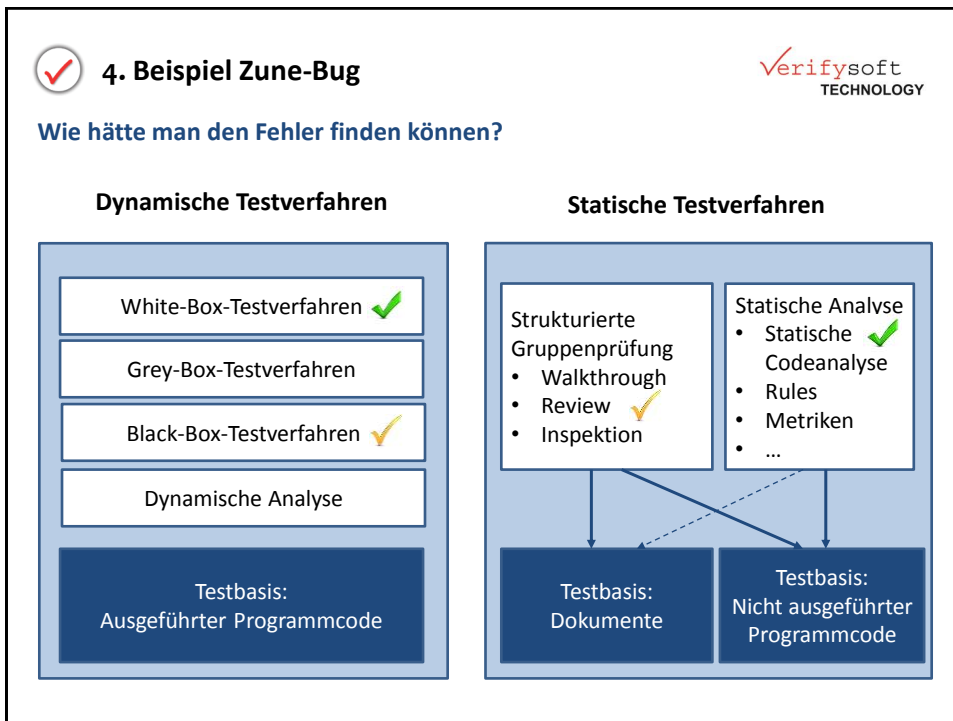
    daysInYear = IsLeapYear(year) ? 366 : 365;
}

time->dayOfYear = days;
time->year = year;
}

```

Bugfix 3

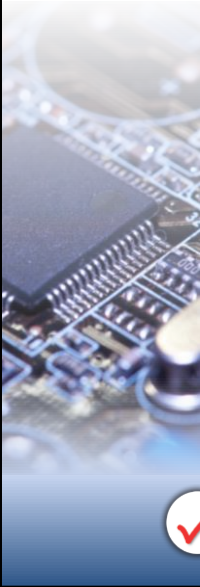
Strukturiert





Essentials!

Verifysoft
TECHNOLOGY

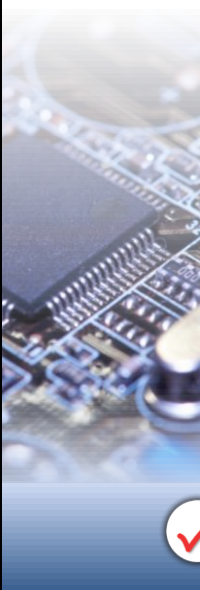


- Irren ist menschlich! Software wird immer komplexer und kann nie fehlerfrei sein.
- Beantwortung der Frage: „Was kostet uns ein Software-Bug?“
- Schaffung der Rolle des Testmanagers! Dieser koordiniert die Testverfahren und Beschaffung von Tools.
- Testverfahren sollten ausbalanciert eingesetzt werden!



Fragen?

Verifysoft
TECHNOLOGY



Gerne stehe ich Ihnen noch bei Fragen
jetzt oder in den Pausen zur Verfügung!