

Berühmt berüchtigte Softwarefehler:

USS Yorktown

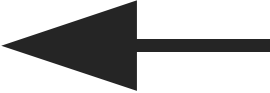
Referent: René Lotz

Seminarleiter: Bernhard Beckert

SS 2003

Universität Koblenz-Landau

Übersicht

- Einleitung 
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”

Einleitung (Smart Ship Concept)

Prinzip:

“Kriegsschiffe werden mit Computertechnik ausgerüstet, um die Matrosen zu unterstützen.”

Vorteile:

- kleinere Crew (Computer statt Matrosen)
- Kampfvorteile durch bessere/präzisere Technik
- geringere laufende Kosten

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”



Allgemeine Informationen über die USS Yorktown

Typ: Guided Missile Cruiser

Breite: 16,8 m

Länge: 173 m

Tiefgang: 10,2 m

Baujahr: 17.01.1983

Besatzung: 24 Offiziere, 340 Matrosen

Wasserverdrängung: ca. 9600 Tonnen bei voller Ladung

Einsatzgebiet:

Unterstützung und Schutz von Flugzeugträgern



Computer-Technik

- 27 Terminals (dual Pentium Pro 200 MHz)
- Glasfasernetzwerk
- Schiff kann von jedem Terminal aus gesteuert werden
- ein Hauptspeicher
- Betriebssystem: Windows NT 4.0

Aufgabenbereiche der Computertechnik


- Brückenfunktionen
 - Überwachung
 - Steuerung
- Schadenskontrolle
 - Sensorabfragen
- Wartung

Unter anderem ist das System auch für die Antriebssteuerung verantwortlich.

Vorteile der Computer-Technik

- weniger Bedarf an Matrosen
 - ca. 10% der Arbeitsplätze eingespart
 - erhebliche Vereinfachung der Überwachungsfunktionen auf der Brücke (nur 3 Personen statt 13)
- geringere Kosten
 - Einsparung hauptsächlich bei den laufenden Kosten
 - ca. 2,8 Millionen \$ pro Jahr Einsparung

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler 
- Schuldfrage
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”

Was ist passiert?

21.09.1997:

- USS Yorktown fährt ein Routine-Übungs-Manöver
- aufgrund eines Software-Problems fällt das Antriebssystem aus
- USS Yorktown schwimmt 2 Stunden und 45 Minuten ohne Antrieb hilflos auf offener See
- danach erreicht sie angeblich aus eigener Kraft die Naval Base in Norfolk, Va. und muss dort 2 Tage repariert werden
- Folgen wären enorm gewesen, wenn dies bei einem echten Einsatz geschehen wäre

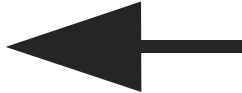
Wie ist es passiert? (Szenario des Fehlers)

- ein Überwachungsprogramm zeigt ein Ventil als geöffnet an, obwohl dieses geschlossen ist
- ein Offizier versucht den Fehler zu beheben
 - ändern von Daten direkt in der Datenbank
 - dies ist nicht vorgesehen, aber durchaus üblich
 - Änderungen werden schriftlich festgehalten
- Eingabe einer "0" an einer bestimmten Stelle
- Software verwendet diesen Eintrag als Divisor

Wie ist es passiert? (Szenario des Fehlers) (II)

- angeblich mehrfache “divide by zero”-Error
- Buffer Overflow des temporären Speichers
- Daten im Hauptspeicher des Antriebssystems werden überschrieben
- das Netzwerk bricht zusammen
- Antriebssystem fällt komplett aus

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage 
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”

verschiedene Sichten der Schuldfrage

Wer ist Schuld?

- NAVY (offiziell): menschliches Versagen
- Windows NT 4.0
- Applikation

Meinung der NAVY: menschliches Versagen

Offizielle Vertreter der NAVY bezeichnen die Ursache des Vorfalls als “menschliches Versagen”.

- Fehlerbehebung durch direkte Änderung der Werte in der Datenbank
 - ändern der Werte von der Software nicht vorgesehen
 - Fehlerbehebung an Bord durch “trial and error”-Prinzip

Meinung der NAVY: menschliches Versagen (II)

dagegen spricht allerdings:

- beliebige Eingaben sollten bei der Entwicklung der Software beachtet werden
- gewisse menschliche Fehlleistungen müssen berücksichtigt werden
- anscheinend keine andere Möglichkeit Fehler zu beheben, ausser der Änderung der Werte direkt in der Datenbank

Meinung der NAVY: menschliches Versagen (III)

dagegen spricht auch:

- besonders auf Kriegsschiffen muss mit Streßsituationen gerechnet werden
- Computersysteme müssen gegen jegliche Bedienfehler, die unter Stress entstehen können abgesichert sein

Windows NT 4.0

Einige Kritiker sind der Meinung, dass Windows NT den Fehler hätte vermeiden müssen.

Zitat:

"Egal welches Betriebssystem, welchen Computer, welche Anwendung ich benutze - ich sollte eine Null eingeben können, ohne dass der Computer abstürzt."

GilYoung, Netzwerkingenieur

Windows NT 4.0 (II)

Zitat:

"Unix ist das bessere System für die Kontrolle von Ausrüstung und Maschinen, NT hingegen für Datentransfer. NT ist nicht ganz ausgereift, es gab **einige Ausfälle** wegen des Systems."

Ron Redman,
stellvertretender technischer Leiter bei der US Marine

Windows NT 4.0 (III)

Microsoft:

- übernimmt keine Verantwortung
- Systemadministratoren und Programmierer der Yorktown seien Schuld
- Software sollte Werte prüfen und Fehler behandeln
- Betriebssystem kann nur Fehlercodes weitergeben, nicht behandeln
- Ein Fehler sollte sich nicht im gesamten Netzwerk ausbreiten

Applikation

- Eingabewerte sollten auf Korrektheit geprüft werden
 - Einträge, welche nicht vom User verändert werden sollen, sollten vor Zugriffen geschützt werden
 - Überlauf des temporären Speichers in den Speicherbereich anderer Systeme sind zu verhindern
 - Trennung der Systemkomponenten nicht ausreichend
 - Versäumnisse bei der Wartung der Software
- Die Haupt-Schuld liegt bei der Software

Applikation Schuldfrage

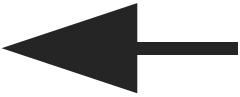
Wichtige Faktoren für die Fehler in der Software:

- zu wenig Software-Entwickler für
 - ↳ Planung
 - ↳ Implementierung und vor allem
 - ↳ **Testen** bzw. **Verifizieren**
 - ↳ **Warten**

der Software

- ✚ Die Schuld liegt eindeutig beim **Management**, welches die Entwicklung unterschätzt hat.

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage
- Fehlervermeidung 
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”

Wie könnten solche Fehler vermieden werden?

- Planung:
 - vollständige Trennung der Systeme
 - Überlauf von einem Speicher in Speicher eines anderen Systems verhindern
- Implementierung:
 - Überprüfen der Einträge auf Korrektheit
 - ggf. Fehlerbehandlung

Wie könnten solche Fehler vermieden werden?

- gründliches Testen
 - sehr (zeit-) aufwendig
- alle möglichen Eingabewerte durchtesten
- dadurch werden solche möglichen Fehler sofort erkannt
- Zusätzliche Fehlerbehandlung kann integriert werden

Wie könnten solche Fehler vermieden werden?

- verifizieren von Programmteilen
 - extrem aufwendig
 - benötigt viele hochqualifizierte Personen
- die sicherste Methode jegliches Fehlverhalten der Software zu vermeiden
- für kritische System-Teile auf Kriegsschiffen unbedingt notwendig

Wie könnten solche Fehler vermieden werden?

- warten der Software
 - ständig ablaufender Prozess
 - erhöht laufende Kosten
- Fehlerbehebung durch direktes schreiben in die Datenbank verhindern
- Alternativen zur Fehlerbehandlung zur Verfügung stellen
- Test der Eingabewerte (akzeptieren des direkten Schreibens in der Datenbank)

Wie wird versucht solche Fehler zu vermeiden?

NAVY:

- Personal wird angewiesen an bekannten Stellen keine "0" einzugeben
- Aufzeichnungen über die Änderungen
- Schiff kann bei einem Fehler schnell wieder funktionsbereit gemacht werden

Bewertung:

Besonders bei einem Kriegsschiff ist dies **keine** Lösung!

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept”



Wieso wurde der Fehler nicht vermieden?

Schuld liegt beim Management.


Ursachen:

- Unterschätzung des Aufwands der Systementwicklung
- Fehleinschätzung der Risiken
- Termindruck
- Kostendruck durch Regierung

Folgen:

- Personalmangel bei der Entwicklung
- erhöhtes Risiko von Fehlfunktionen

Übersicht

- Einleitung
- Allgemeine Informationen über die USS Yorktown
- Der Software-Fehler
- Schuldfrage
- Fehlervermeidung
- Wieso wurde der Fehler nicht vermieden?
- “Smart Ship Concept” 

Smart Ship Concept

Vorteile des Konzepts sind offensichtlich:

- weniger Crew nötig
- geringere laufende Kosten für
 - Personal
 - Wartung
- normalerweise geringeres Risiko von Fehlern durch “menschliches Versagen”

Smart Ship Concept (Beispiel)

US Zerstörer USS McFaul

- Besatzung: 350 Soldaten

für 2008 ist ein Nachfolger angekündigt:

“Smart Ship” USS Zumwalt

- Besatzung: 90 Soldaten !
- auf Windows 2000 basierendes Betriebssystem

Smart Ship Concept (Ausblick)

Die NAVY will auch in Zukunft viele Schiffe nach diesem Konzept bauen.

- nur höchstens ein Drittel der bisherigen Besatzung
 - ↳ laufende Kosten für Personal, Wartung, ... reduzieren
- Einsatz von Windows 2000 als Betriebssystem
 - ↳ Beschaffungskosten reduzieren

Allerdings erhöhen sich die laufenden Kosten aufgrund der nötigen Software-Wartung.

Smart Ship Concept (kritisch)

Kritiker bemängeln, dass weiterhin standardisierte Software (wie Windows NT / 2000) in Kriegsschiffen verwendet werden soll.

Gründe dafür sind:

- zu wenig verifiziert
- zu wenig getestet
- Windows NT/2000 nicht uneingeschränkt echtzeitfähig
- zu hohes Risiko von Fehlfunktionen
- evtl. gefährliche Folgen von Fehlfunktionen

Smart Ship Concept (mögliche Risiken)

Auf einem Kriegsschiff könnten beliebige Fehlfunktionen verheerende Folgen haben.

z.B.

- in einem Kampfeinsatz nicht manövrierfähig
- Unkontrollierte Steuerung der Ventile,...
- Beeinflussung taktischer Systeme
 - z.B. durch Überschreiben von Daten in deren Speicherbereich
 - könnte zu ungewollten Raketenabschüssen,... führen

Smart Ship Concept (Fazit)

Vorteile:

- Einsparung von Besatzung
- Einsparung von Kosten

Nachteile:

- großes Risiko bei
 - ↳ leichtfertiger Entwicklung
 - ↳ Verwendung von “Standard-Software”