

Vorlesung "Anwendungssysteme"

Risiken bei Informatiksystemen

Prof. Dr. Lutz Prechelt
Freie Universität Berlin, Institut für Informatik
<http://www.inf.fu-berlin.de/inst/ag-se/>

- Begriffsdefinitionen
- Arten von Gefahren/Risiken
 - Technik
 - Menschen
 - Ereignisse/Kopplungen
- Beispiele
 - von bedrohlich über kurios bis hin zu schrullig
- Methodik für Sicherheit
- Probleme dabei

- Bisher haben wir eine sehr breite Sicht auf die Auswirkungen von Informatiksystemen benutzt
 - Alle Arten von Wirkungen waren von Interesse
- Diese Einheit wird jetzt spezieller:
 - Wir interessieren uns für Bedrohungen der Sicherheit
 - Insbesondere Gefahren für Leib und Leben
 - aber auch schwächere Arten (z.B. plötzliche Freiheitseinschränkungen)



- Unfall (*accident*):
 - Unerwünschtes, ungeplantes (aber nicht zwingend ganz unerwartetes) Ereignis, das zu einem Schaden führt
- Bedrohung, Gefahr (*threat, hazard*):
 - Eine mögliche Einwirkung auf das System, die zusammen mit anderen zu einem Unfall führen kann
- Sicherheit (*safety*):
 - Zustand des Geschütztseins vor Unfällen
 - d.h. alle Gefahren werden vermieden oder beherrscht
- Risiko (*risk*):
 - Eine im Prinzip quantitative Größe. Produkt aus Eintrittswahrscheinlichkeit eines Unfalls und Schadenshöhe
 - Solche Berechnungen sind aber meist sehr dubios
 - Schwächer: Eine Möglichkeit, dass (oder wie) ein System unerwünschtes Verhalten zeigen kann
 - ...was man aber gelegentlich bewusst in Kauf nimmt

- Versagen (*failure*):
 - Eine untolerierbare Abweichung eines Systems von seiner Spezifikation
 - Kann (direkt oder indirekt) zu Unfall führen, muss es aber nicht
- Zuverlässigkeit (*reliability*):
 - Qualitativ: das Funktionieren eines Systems gemäß der Spezifikation unter allen vorgesehenen(!) Bedingungen
 - Quantitativ: Wahrscheinlichkeit des Nichtversagens
- Robustheit (*robustness*):
 - Grad der Fähigkeit eines Systems, auch unter *unvorhergesehenen* Bedingungen Unfälle auszuschließen

- Fehler (*error*):
 - Ein Ereignis beim Bau eines Systems, das zu einem Mangel führt oder führen kann
- Mangel, Defekt (*fault, defect*):
 - Eine strukturelle Unzulänglichkeit in einem System, die zu Versagen führt oder führen kann
- Schutz (*security*):
 - Die Widerstandsfähigkeit eines Systems gegen absichtliche Angriffe. Das System ist sicher (geschützt, *secure*), wenn die Angriffe ohne Unfall überstanden werden
 - Bei Informatiksystemen insbesondere: Informationssicherheit
 - Teilaspekt von Sicherheit

Arten von Risiken

Es gibt viele Arten, **Risiken** zu **klassifizieren**:

- Nach Art der Abweichung:
 - Ein System zeigt eine erwünschte u. erwartete Eigenschaft nicht
 - Ein System zeigt eine unerwünschte Eigenschaft
- Nach Wahrscheinlichkeit/Häufigkeit des Eintretens:
 - z.B.: sehr gering, gering, erheblich
- Nach Höhe des Schadens:
 - z.B.: vernachlässigbar, gering, erheblich, hoch, untragbar
- Nach Art der Ursachen:
 - menschliches Versagen, technisches Versagen, beides
 - eine Ursache, mehrere Ursachen
- und andere

Die wichtigsten Arten von Bedrohungen

1. Software enthält Defekte
2. Hardware und technische Einrichtungen versagen
3. Menschen sind
 - fehlbar (insbesondere unter Stress)
 - unvorsichtig (insbesondere wenn sie sich sicher fühlen)
 - oftmals ignorant
 - manchmal böswillig
4. Auch unwahrscheinliche Ereignisse treten gelegentlich ein
 - auch mehrere zugleich
5. Scheinbar unverbundene Ereignisse sind tatsächlich oftmals eng gekoppelt

Beispiele folgen

- Achtung: Häufig passen sie auch in andere Kategorien (d.h. die Einordnung ist oft mehrdeutig)

Software enthält Defekte

- Bekanntlich gibt es in umfangreicher Software meist auch nach gründlichem Test noch mehrere Entwurfs- oder Programmierungsfehler
- Viele davon lassen sich nur extrem schwer mit Tests entdecken
- Man kann sie mit Durchsichten entdecken
- Aber nur, wenn man weiß, wonach man sucht (d.h. die Anforderungen sind bekannt)
 - Das ist nicht immer der Fall
- Beispiele folgen

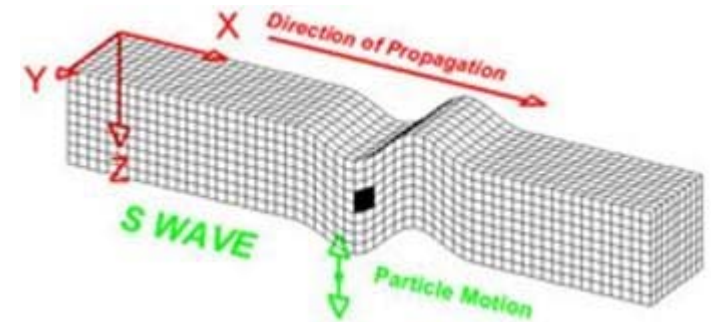
Softwaredefekte: Simulations-SW für Space-Shuttle-Missionen

- Zur Vorbereitung auf die zweite Mission des Space-Shuttle übten die Astronauten mit einem **Simulator**
- Sie entschieden sich unterwegs, die Mission abubrechen, überlegten es sich dann aber anders und brachen das Abbrechen ab
 - Die Simulation funktionierte korrekt weiter
- Nach der nächsten Erdumrundung brachen Sie erneut ab
- Das Programm geriet in eine Endlosschleife
- Die Entwerfer des Simulators hatten die Möglichkeit zweier Abbrüche nie bedacht
 - Software Engineering Notes (SEN) 8(3), 1983



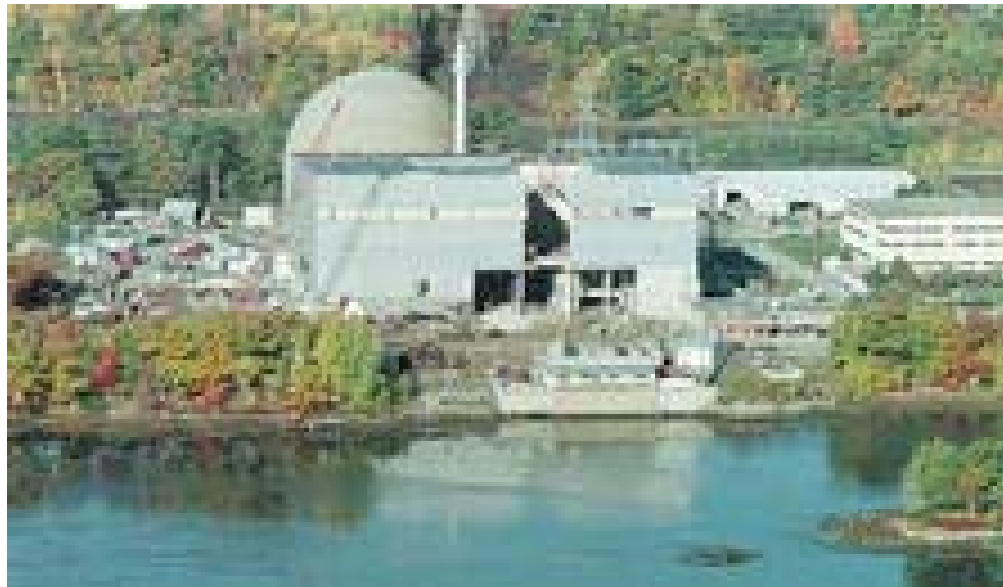
Softwaredefekte: Erdbeben-Simulation 1979

- In einem Programm zur Simulation der Wirkung von Erdbeben auf Gebäude wurde ein kleiner Fehler entdeckt:
 - An einer Stelle hätte man $\sum_k (|x_k|)$ berechnen müssen, es wurde aber $\sum_k (x_k)$ berechnet
 - Wurde nur entdeckt, weil ähnliche Läufe unerklärlich unterschiedliche Resultate ergaben
- Das Programm war verwendet worden, um die Erdbebenfestigkeit von Reaktorgebäuden für Kernkraftwerke sicherzustellen
 - Laut Gesetzgebung müssen diese den stärksten je in der Region erlebten Beben stand halten
- 5 Kernkraftwerke wurden deshalb dauerhaft abgeschaltet:
 - in Maine, New York, Pennsylvania, Virginia, Virginia



Softwaredefekte: Erdbeben-Simulation 1979 (2)

- Interessante Technikfolgenbewertungs-Frage hier:
 - Darf und will man sich für kritische Entscheidungen auf eine Simulation verlassen, die man nicht experimentell überprüfen kann?



Hardware und technische Einrichtungen versagen oder gehen kaputt

- Auch die zuverlässigsten technischen Systeme versagen zumindest hin und wieder oder gehen kaputt
 - z.B. können selbst intakte Computer in der Tat irren, wenn kosmische Strahlung den Inhalt einer Speicherzelle ändert

- Beispiele folgen

Technik versagt: Telefonausfall New York 1991

- **Notstromgenerator** für eine Vermittlungsstelle sprang nicht an
- System lief 6 Stunden lang auf Notbatterie, bis diese leer war
- Die vorgesehenen Signale, um Personal zu Hilfe zu holen, sprangen nicht an
 - weil sie außer Betrieb genommen worden waren:
 - sie hatten zuvor mehrfach wegen Bauarbeiten falschen Alarm gegeben
- Hätten auch nicht zwingend geholfen, denn beide zuständigen Notfallbearbeiter waren abwesend
 - sie besuchten einen Kurs über Stromversorgungs-Notfälle
- Resultat:
 - ca. 5 Mio. ausgefallene Telefongespräche,
 - 4 Stunden Ausfall aller drei New Yorker Flughäfen



Technik versagt: Geldautomaten-Versagen 1993

- Am 13. März 1993 versagten ca. 5000 Geldautomaten in allen Gegenden der USA
- Grund: Das **Dach** des zuständigen Rechenzentrums war eingestürzt
 - Grund: zu hohe Schneelast auf dem Dach
- Das Ersatz-Rechenzentrum war nicht verfügbar, weil es wegen des Terroristenangriffs auf das World Trade Center (einen Monat zuvor) bereits belegt war



- Menschen **machen Fehler**, insbesondere unter Stress.
 - Fast jedes noch so gut gestaltete System kann durch Bedienfehler in unerwünschte Zustände geraten
 - Entscheidend ist dann, ob es zusätzliche Sicherungen gibt, die das höchstwahrscheinlich bemerken und zu reparieren erlauben
- Menschen sind oft **unvorsichtig**
 - Insbesondere, wenn sie sich zu sicher fühlen,
 - was z.B. durch gute Sicherheitsvorkehrungen passieren kann...
- Beispiele folgen

Menschen sind fehlbar: Frontaler Zugcrash Berlin 1993

- Während Bauarbeiten zur Elektrifizierung der Strecke war Wochentags nur einspuriger Betrieb bei Wannsee
- Am Karfreitag konnten beide Spuren benutzt werden
 - Der **Fahrdienstleiter** stellte den Schalter jedoch versehentlich trotzdem auf "einspurig" ein
 - Der Computer setzte korrekt ein Signal auf "Halt"
 - Der **Fahrdienstleiter** hielt dies für ein Versagen. Er stellte das Zusatzsignal (extra für Bauarbeiten!) auf "Weiterfahren", weil ein Zug durch sollte
 - Er übersah einen entgegenkommenden, nicht fahrplanmäßigen Zug
 - Er fragte nirgends wegen des "Versagens" zurück
- Aber: Es gab "nur" 3 Tote und 20 Verletzte
 - Denn beide Züge fuhren (wegen einspurig) nur Tempo 30



Menschen sind fehlbar: Beinahe-Unfall Space Shuttle 1986

- Übermüdete **Techniker** interpretierten die Anzeige eines Ventilversagens falsch
- Sie ließen deshalb eine Sauerstoffleitung offen
 - Dadurch flossen 5 Minuten vor dem Start 68.000 Liter flüssiger Sauerstoff aus dem externen Tank ab
- Im Startfall hätte das Shuttle seine Umlaufbahn nicht erreichen können
- Aber: Der Countdown wurde bei -31 sec abgebrochen, nachdem ein Temperaturfühler durch den Abfluss unerlaubt niedrige Werte angezeigt hatte



Menschen sind unvorsichtig: Untergang der Titanic

- Paradebeispiel hierfür ist der Untergang der Titanic (1912)
 - er kam nur zustande durch extremen Leichtsinn
 - dieser wurde möglich, weil das Schiff als unsinkbar galt:
 - 4 von 16 Abteilungen durften vollaufen, ohne dass das Schiff sinken würde. So ein Fall war noch nie vorgekommen
 - Das späte Ausweichen vor dem Eisberg führte zum Aufschlitzen von 5 Abteilungen
 - Man hatte keine Sicherheitsübungen gemacht und 50% zu wenig Boote an Bord
- Oft ist die Wirkung des Leichtsinns aber subtiler



Menschen sind unvorsichtig: Osprey-Schwenkrotor-Flugzeug

- Die Konstruktion enthält eine computerunterstützte Steuerung
- Diese benutzt z.B. Rollraten-Sensoren
- Zur Ausfallsicherheit gibt es diese Sensoren dreifach
- Ein Prototyp stürzte ab, weil 2 der 3 Sensoren verpolt angeschlossen waren
 - Bei 2 weiteren Prototypen entdeckte man daraufhin je 1 verpolten Sensor
 - Arglosigkeit des **Entwerfers** (k. Warnung)
 - Schlamperei der **Monteure**



Menschen sind unvorsichtig: Chernobyl 1986

- Aufgrund einer Kernschmelze im Reaktor Chernobyl 4 traten große Mengen Radioaktivität aus
 - Mehrere Tausend Tote, ca. 0,5 Millionen Verseuchte
- Der Unfall geschah während der Durchführung von Experimenten zum Thema "Wie weit reicht Notstrom zum Kühlen aus?"
 - Für die Experimente hatten die **Techniker** mehrere Sicherheitssysteme abgeschaltet: Notkühlsystem, Leistungsregelg., automatische Notabschaltung
- Die **Konstruktion des Reaktors** war inhärent gefährlich und schwer zu kontrollieren
 - z.B. Grafit (brennbar!) für die Moderatorstäbe



Menschen sind oftmals ignorant

- Es fehlt ihnen an Wissen
 - selbst an entscheidenden Stellen
- ohne dass sie das merken

- Daraus resultieren sowohl Entwurfsfehler als auch Benutzungsfehler
 - Die Entwurfsfehler sind eine besonders interessante Kategorie

- Beispiele folgen

Ignoranz: Abkühlautomatik im Stahlwerk 1993

- Deutsches Stahlwerk benutzt Computersteuerung für den Abkühlprozess des frischen Stahls und zum Start der Weiterverarbeitung
- Die **Programmierer** hatten die Zeitmessung auf die Normalzeituhr der PTB abgebildet (amtliche Uhrzeit)
 - Physikalisch-technische Bundesanstalt in Braunschweig
- Umschaltung auf Sommerzeit verkürzte die Kühlzeit um 1 Stunde
- Riesige Sachschäden am Stahlwerk
- 2 Ignoranz-Fehler:
 - Eigentlich kommt es auf die Stahl-Temperatur an, nicht auf Zeit
 - Uhrzeitunterschied ist nicht immer ein Indikator für die abgelaufene Zeit



Menschen sind oftmals ignorant: U-Bahn ohne Fahrer 1993

- Eine U-Bahn-Tür klemmte und schloss nicht richtig.
- Der **Fahrer** stieg aus, um das zu beheben.
- Sobald die Tür schloss, fuhr der Zug los – ohne den Fahrer
- Der **Fahrer** hatte den Knopf zum Losfahren mit Klebeband in "gedrückt"-Stellung festgeklebt
 - Bahn stoppte automatisch und fuhr automatisch los, sobald (aber erst wenn) die Türen wieder geschlossen waren
 - Fahrer hatte die Betriebsvorschrift verletzt, niemals die Führerkabine eines solchen Zugs zu verlassen
 - und nicht an die Konsequenzen gedacht
- **Entwerfer** hatte die Faulheit des Fahrers vorherzusehen versäumt

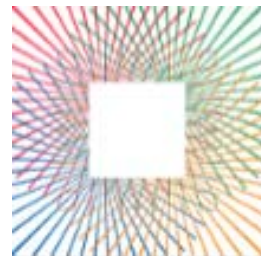


Auch unwahrscheinliche Ereignisse treten gelegentlich ein

- Manchmal funktionieren Systeme für lange Zeit ganz sicher, bis ein sehr seltenes Ereignis eintritt
- Deshalb kann Testen als Sicherheitsgarantie nicht ausreichend sein
- Beispiele folgen

Unwahrscheinliche Ereignisse: Kontoüberziehung Bank of New York

- In der Buchungssoftware lief unbemerkt ein Zähler über
 - Dadurch **konnte** die Bank einlaufende ***Gutschriften nicht verbuchen***
 - Die Federal Reserve Bank buchte hingegen die Belastungen von Zahlungsausgängen normal weiter
- Es entstand ein Soll von 32 Milliarden Dollar
- Die BoNY musste sich für einen Tag 24 Milliarden Dollar leihen
 - Die Zinskosten dafür betrugen 5 Millionen Dollar



The **BANK**
of **NEW YORK**



In the 1990s The Bank of New York moved its headquarters to One Wall Street.

Unwahrscheinliche Ereignisse: McMuffin

- McDonalds hatte ein seit Jahren korrekt funktionierendes Arbeitszeit-Erfassungssystem
- Plötzlich **liefen dessen Uhren weitaus zu schnell**
 - Die Angestellten bekamen 2 bis 4 Stunden zu viel Arbeitszeit zugerechnet
 - und erhielten entsprechend höhere Lohnzahlungen
- Ursache waren die Zeitgeber in den neuen Toastern für das neue Produkt McMuffin
 - Diese erzeugten Impulse im Stromnetz, die die Uhren beeinflussten
 - Zugleich kam es zusätzlich zu Phantombestellungen in den Kassen, die das Inventar- und Kassensystem ebenfalls durcheinander brachten



Unwahrscheinliche Ereignisse: "Your 'Cancel' was canceled"

- Mitarbeiter von Mizuho Securities wollte verkaufen 1 Aktie von J-Com für 610.000 Yen (~6.000 USD)
- Er vertauschte zwei Felder und gab ein:
verkaufe 610.000 Aktien von J-Com für je 1 Yen
 - Transaktionssystem der Tokyo Stock Exchange, 8. Dezember 2005, 9:27 Uhr
 - Warnung "Beyond price limit" ignoriert
- 9:29-9:35 Uhr:
Der Auftrag lässt sich nicht löschen
 - Subtiler Fehler im Ordersystem; *seit 5 Jahren unentdeckt!*
- ab 9:35 Uhr: Mizuho kauft von sich selber die Aktien zurück
 - Es gab vom J-Com überhaupt nur 14.500 Aktien
 - Mizuho konnte 510.000 zurückkaufen, 100.000 nicht
 - zahlte dafür bis zu je 912.000 Yen
- Gesamtverlust: ~40 Milliarden Yen (~400 Mio. US-Dollar)



Scheinbar unverbundene Ereignisse sind tatsächlich oftmals eng gekoppelt

- Auch wenn ein System entwurfsseitig scheinbar keine einzelnen Versagenspunkte enthält (sondern überall Redundanz hat), kommen Versagen durch einzelnes Ereignis vor
- Denn manchmal sind die mehreren nötigen Ereignisse gar nicht wirklich verschieden
- Beispiele folgen

Gekoppelte Ereignisse: ARPAnet-Versagen 1986

- Das ARPAnet (direkter Vorläufer des Internet) war stets für Fehlertoleranz entworfen
- Deshalb war z.B. die Region Neuengland an den Rest des ARPAnet nicht nur über 1 Leitung, sondern über 7 getrennte Leitungen angebunden
- Alle diese 7 Leitungen wurden am 12.12.1986 gleichzeitig bei Baggerarbeiten durchtrennt
- Denn sie liefen **alle über das selbe Glasfaserkabel**
 - SEN 12(1), 1987



Bsp: Chicago 1990

- Was muss passieren, damit in der ganzen Region Chicago gleichzeitig (z.B.)
 - der Flugverkehr lahm gelegt wird und
 - Geldautomaten kein Geld mehr ausspucken?
- Krieg? Erdbeben? Unwetter?
- Nein: Man muss nur ein Bäumchen pflanzen:
 - Die Gartenbaufirma durchtrennte dabei ein **Telefonkabel, über das 150.000 Leitungen angeschlossen waren**
 - Der Tower von O'Hare Airport hatte keine Verbindung mehr zur zuständigen zentralen Fluglotsenstelle
 - Viele Geldautomaten konnten die Abbuchungen nicht mehr prüfen und stellten den Betrieb ein
 - SEN 16(1), 1991

Ende der Beispiele

- Im Prinzip ist die Rolle von Computern und Informatik bei Sicherheitsproblemen keine besondere
- Aber Computer haben die Entwicklung höchst komplizierter und riskanter Systeme extrem verstärkt:
 - Größere solche Systeme als je zuvor
 - Stärkere Wechselwirkungen zwischen ihnen
 - Mehr und schnellere Veränderungen an ihnen
 - Manchmal ein naives Vertrauen in ihre Verlässlichkeit

Hierarchische Sicht von Unfällen

Zum Verstehen eines Unfalls sollte man drei Ebenen unterscheiden:

1. Mechanismen:

- konkreter Hergang beim Unfall. Rein beschreibend.

2. Bedingungen:

- Zustand des Systems und seiner Umgebung bei Beginn des Unfalls. (Zum Verstehen des Hergangs.)

3. Urgründe (*root causes*):

- Allgemeine Bedingungen im Umfeld des Systems, die zu den konkreten Bedingungen bei Unfallbeginn geführt haben. (Zum Vermeiden ähnlicher Unfälle in der Zukunft.)

- Urgrund-Analyse wird leider selten gemacht, weil Sie fast immer einige Beteiligte beschämt.

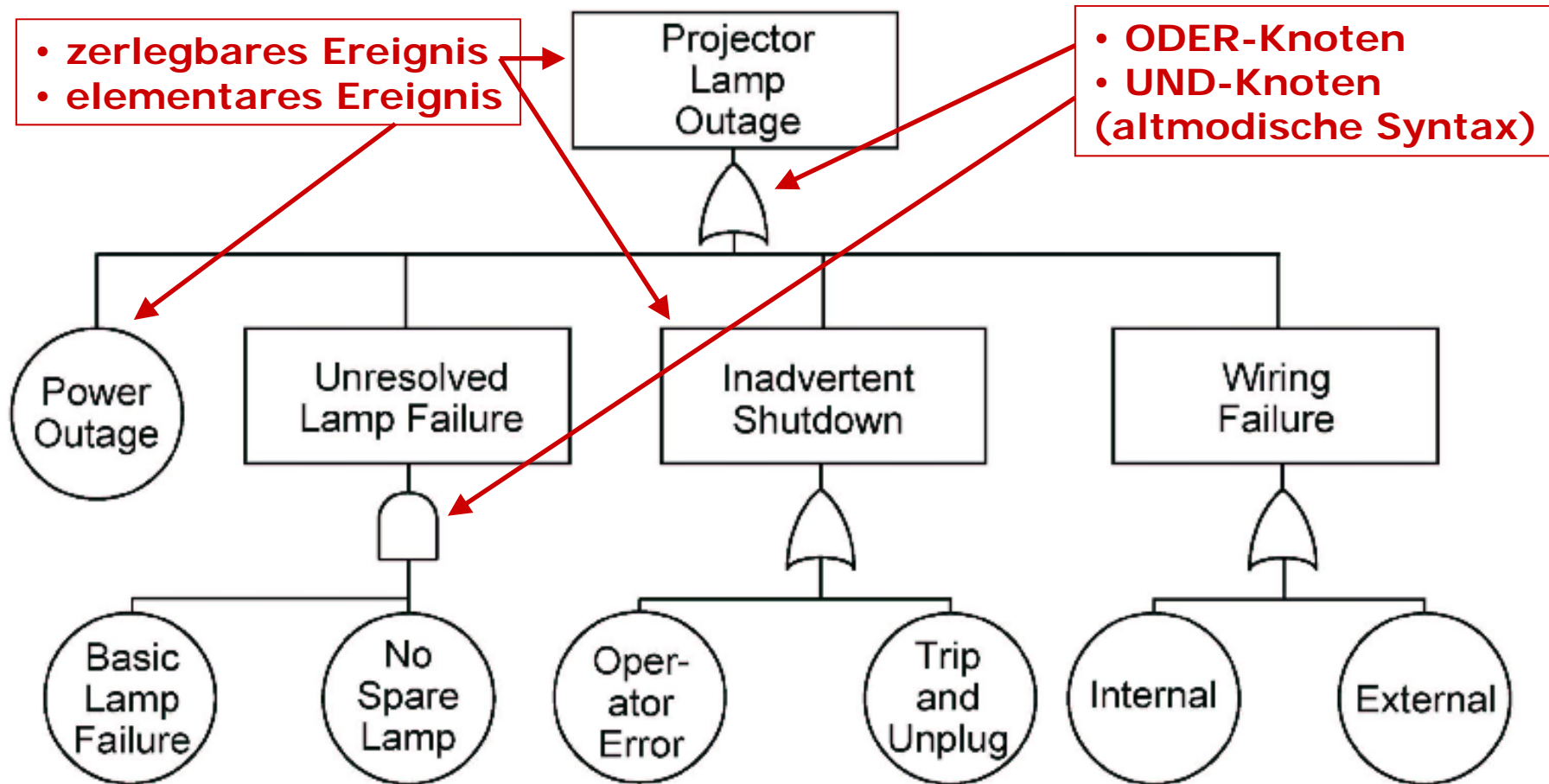
Was man aus den Beispielen lernen kann

1. Vollständigkeit von Anforderungen ist kritisch
2. Menschliches Versagen ist allgegenwärtig
 - auch da, wo man es nicht dulden will
3. Technisches Versagen ist häufiger als uns lieb ist
 - und kann auch indirekt von Mensch, Tier oder Natur verursacht sein
4. Für schwere Unfälle gibt es meist mehr als eine Ursache
5. Wir lassen uns von Vorsichtsmaßnahmen gern einlullen
6. Je komplexer ein System wird, desto mehr Risiken hat es
 - leider steigern Gegenmaßnahmen gegen Risiken oft die Komplexität noch weiter...
 - Klarheit und Einfachheit sind hohe, aber schwierige Ziele

Vorgehen zum Bau risikoarmer Systeme

1. Gefahrenbestimmung und -analyse
 - Welche Gefahren gibt es?
2. Risikoanalyse
 - Zu jeder Gefahr: Wie hoch ist das Risiko? Kann man es ganz umgehen? Lohnen sich Vorbeugung oder Abwehr?
3. Entwurf
 - Gegenmaßnahmen erfinden und umsetzen.
Vorrangfolge beachten (siehe unten).
4. Entwurfssicherheitsprüfung
 - Ähneln Gefahrenbestimmung, aber viel konkreter.
Nötigenfalls Entwurf nachbessern und Prüfung wiederholen.
5. Bau
6. Abnahmesicherheitsprüfung
 - Noch konkreter. Bau bewirkt stets Entwurfsänderungen.

- Idee: Gefahrereignisse so lange hierarchisch in Bedingungen zerlegen, bis Gefahr verstanden ist



Vorgehen: Grundregeln

- Baue Sicherheit von vornherein ein
 - Mehr dazu nächste Stunde
- Betrachte das System als Ganzes, nicht seine Teile
 - Sicherheit ist eine Eigenschaft des Systems, nicht der Teile
 - Selbst wenn alle Teile versagensfrei sind, kann das System vielfältig unsicher sein
- Verlasse Dich nicht allein auf Erfahrungen und Standards
 - Jedes System ist anders. Analysiere es!
- Verwende qualitative statt quantitative Methoden
 - Zahlen führen nur in die Irre (falsche Prioritäten)
- Gestehe ein, dass Abwägungen nötig sind und Konflikte auftreten; perfekte Sicherheit gibt es nicht

Vorgehen: Grundregeln (2)

Beachte die Vorrangfolge beim Umgang mit Gefahren:

- z.B.: *Datenqualität: Gefahren durch Fehler in Datenbeständen*

1. Gefahr eliminieren ("intrinsische Sicherheit")

- z.B.: eine Information, die nicht gespeichert wurde, kann auch nicht falsch sein

2. Gefahr reduzieren

- z.B.: falsche Dateneingabe durch Plausibilitätsprüfungen unwahrscheinlicher machen

3. Gefahr beherrschen (passiv, aktiv)

- z.B. passiv: Schadenersatz vertraglich ausschließen
- z.B. aktiv: Datenqualität regelmäßig überwachen, Korrekturen

4. Schaden verringern

- z.B. Schadenersatzhöhe begrenzen oder potentiell falsche Daten nur für mäßig wichtige Zwecke verwenden

Risiken der Risikoanalyse

Es gibt nur drei Möglichkeiten, alle sind problematisch:

1. Risiken werden unterschätzt

- Vermeidbare Schäden beim Systembetrieb treten ein
 - Die klassische Niederlage

2. Risiken werden überschätzt

- Vermeidbare Kosten beim Systembau treten ein
 - Eine teilweise Niederlage

3. Risiken werden richtig eingeschätzt

- Gerade weil nun Unfälle vermieden werden, kann der Wert der Risikoanalyse nicht bewiesen werden
- Das kann die nächste Risikoanalyse beeinträchtigen
 - Ggf. eine indirekte Niederlage

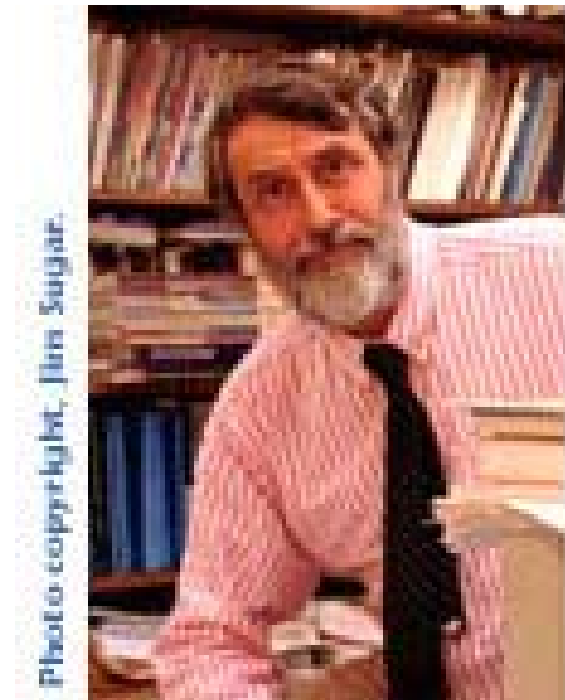
Probleme beim Entwurf sicherer Systeme

- Sicherheit kostet zusätzliches Geld
 - "Amortisation" ist unklar
- Risikobetrachtungen haben negativen "Geruch"
 - sind deshalb schwer zu vermitteln
- Sicherheitsmaßnahmen sind nicht sexy
 - Nicht so imponierend wie Funktionalität.
Schwer zu verkaufen.
- Oft wird das Vorhandensein guter Sicherheitsmaßnahmen mit vollständiger Sicherheit gleichgesetzt
 - "Titanic-Effekt"
- Oder es herrscht die Haltung vor
"Man bekommt es ja sowieso nicht sicher hin"

Quellen für viele der Beispiele in diesem Foliensatz:

- Peter G. Neumann: *Computer-related Risks*, Addison-Wesley 1995
 - ist ein "best of" der SEN RISKS-Meldungen
- ACM Sigsoft Software Engineering Notes (SEN)
 - enthält ein "best of" des RISKS Forums
 - <http://www.acm.org>
 - Volltext verfügbar aus dem FU-Netz
- The RISKS Forum
 - <http://www.risks.org>

Peter G. Neumann



- Fast alle Informatiksysteme bergen Risiken
 - nicht unbedingt gleich für Leib und Leben
- Die Risiken entstammen vielen verschiedenen Quellen
 - die meisten davon sind nicht technischer Natur
- Risiken lassen sich nur beherrschen, wenn das System im Ganzen betrachtet wird
 - und nicht nur die Software oder nur das technische System
- Unfälle entstehen fast immer durch das Zusammenwirken mehrerer Gefahren
 - Ihre Vermeidung verlangt also entsprechend vernetztes Denken beim Systementwurf
 - Einen solchen Fall werden wir sehr ausführlich in der nächsten Stunde betrachten

Danke!

Es folgen noch einige Bonusfolien

Vorgehen: Gefahrenbestimmung

- Nutze historische Daten
 - Sicherheitserfahrungen, Problem- und Unfallberichte
- Nutze öffentliche Checklisten, Standards, Vorgehensbeschreibungen
- Für physische Gefahren: Untersuche Energiequellen, Energieflüsse, gefährliche Materialien
- Untersuche die Mensch-Maschine-Schnittstelle
 - Ist sie verständlich? Für jeden? Ist sie robust?
- Untersuche den Prozess des Systembaus (insbes. SW)
 - Solide Ingenieurpraktiken?
- Untersuche alle normalen/besonderen Betriebszustände
 - insbes. Übergänge (Start, Stopp, Zusammenbruch, Reparatur, etc.)

- Untersuche denkbare ungewöhnliche Umstände des Betriebs
 - Umwelteinflüsse (Wetter, Tiere, Erdbeben, ...)
 - Infrastrukturversagen: Strom, Kühlung, ...
 - Hardwareversagen (endgültig, vorübergehend)
 - Missbenutzung: Fehlinstallation, Fehlbedienung, Gebrauch zu anderem Zweck, Missachtung von Meldungen, vorsätzlicher Missbrauch
- Beachte die langfristige Perspektive
 - Änderungen am System; Änderungen am technischen, organisatorischen oder sozialen Umfeld
- Gehe schrittweise durch Gesamtprozess des Systems
 - Was kann schief gehen? Wie kann man es vermeiden?
 - Was ist zu tun, wenn der schlimmste Fall eintritt?



Relevant bei Ersatz elektromechan. Sicherheitsmechanismen durch SW-Mechanismen:

- Mythos: *"Computer sind billiger als analoge oder elektromechanische Sicherheitssysteme"*
 - Stimmt nur bei hohen Stückzahlen, weil die Entwicklung sicherer SW sehr teuer ist
- Mythos: *"Software ist leicht zu ändern"*
 - Aber sehr schwer korrekt zu ändern
- Mythos: *"Computer sind zuverlässiger als analoge oder elektromechanische Systeme"*
 - Hardwareseitig ja, aber es kommen häufig Softwaredefekte hinzu, die die Sicherheit beeinträchtigen
 - z.B. Space Shuttle: Höchste denkbare Qualitätsansprüche bei der Entwicklung. Extrem konservativer Ansatz. 1980–95 wurden dennoch 16 gefährliche Defekte aufgedeckt.

Mythen über Software und Sicherheit (2)

- Mythos: *"Erhöhung der SW-Zuverlässigkeit steigert die Sicherheit"*
 - Stimmt oft, aber nicht immer, weil viele Sicherheitsprobleme in Anforderungen oder Entwurf begründet sind und nicht von mangelnder Korrektheit herrühren
 - Stimmt tendenziell nicht, weil ein Bewusstsein hoher SW-Zuverlässigkeit die Vorsicht untergraben kann
- Mythos: *"SW-Wiederverwendung steigert die Sicherheit"*
 - kann stimmen wegen höherer Zuverlässigkeit (siehe oben)
 - kann aber auch falsch sein: Ein sicherheitskritisches System wird so entworfen, dass die Sicherheitsargumente möglichst einleuchtend werden.
Das ist bei Wiederverwendung oft nicht mehr möglich.

- Risiken müssen ernst genommen werden
 - Wo keine ausdrücklichen gesetzlichen Vorschriften bestehen (und selbst dort), ist das leider oft nicht der Fall
- Stillschweigende Annahmen müssen sichtbar gemacht werden
 - damit sie überprüft werden können
 - und dann müssen sie überprüft werden!
- Sorgfältige Implementierung ist unverzichtbar
 - denn selbst gute Fehlertoleranzmaßnahmen sind begrenzt
- Ganzheitliches Denken ist unverzichtbar
 - Nicht nur die SW, sondern das ganze System betrachten
 - Nicht nur das System, sondern auch sein Umfeld betrachten

Weitere Beispiele

Bsp: Shirley Jackson

- Eine Shirley Jackson wurde polizeilich gesucht
 - per Haftbefehl im Rechner des National Crime Information Center (NCIC)
- Eine andere Frau wurde von der **Polizei** festgenommen,
 - trotz anderem Namen: Sheila Jackson Stossier
 - trotz anderer Körpergröße: 15 cm Unterschied
 - trotz anderem Körpergewicht: 31 kg Unterschied
 - trotz der Tatsache, dass die Gesuchte bereits im Gefängnis saß
- Es wurde sogar ein Eintrag über die Inhaftierung in der Datenbank des NCIC angelegt

Bsp: Terry Dean Rogan

- US-Bürger Terry Dean Rogan verlor seine Brieftasche
 - samt Führerschein und Kreditkarten
 - der **Finder** nahm seine Identität an und verübte zwei Morde und zwei Raubüberfälle
 - Es wurde ein Haftbefehl auf Terry Dean Rogan ausgegeben
- Er wurde binnen 14 Monaten 5 mal von der Polizei festgenommen,
 - obwohl er seit dem ersten Mal beständig versuchte, die Datenbankeinträge korrigieren zu lassen
 - Ihm wurden vor Gericht \$ 55.000 Schadenersatz zugesprochen, zahlbar von der Polizei von Los Angeles



Bsp: Geld als Weihnachtsgeschenk

- Am 24.12.1987 überwies ein holländischer **Bankangestellter** sich selbst 8,4 Millionen Dollar und 6,7 Millionen Dollar auf ein Schweizer Konto
- Das Sicherheitssystem der Bank sah vor, dass Überweisungen von zwei Personen autorisiert werden müssen (4-Augen-Prinzip)
- Der Mann kannte jedoch das Passwort eines Kollegen
- Die zweite Überweisung gelang wegen eines technischen Fehlers nicht
 - nur dadurch flog das Ganze am nächsten Tag auf

