# Why Things Break --

## With Examples From Autonomous Vehicles

Phil Koopman
Department of Electrical & Computer Engineering
& Institute for Complex Engineered Systems

(based, in part, on material from Dan Siewiorek)

Carnegie Mellon

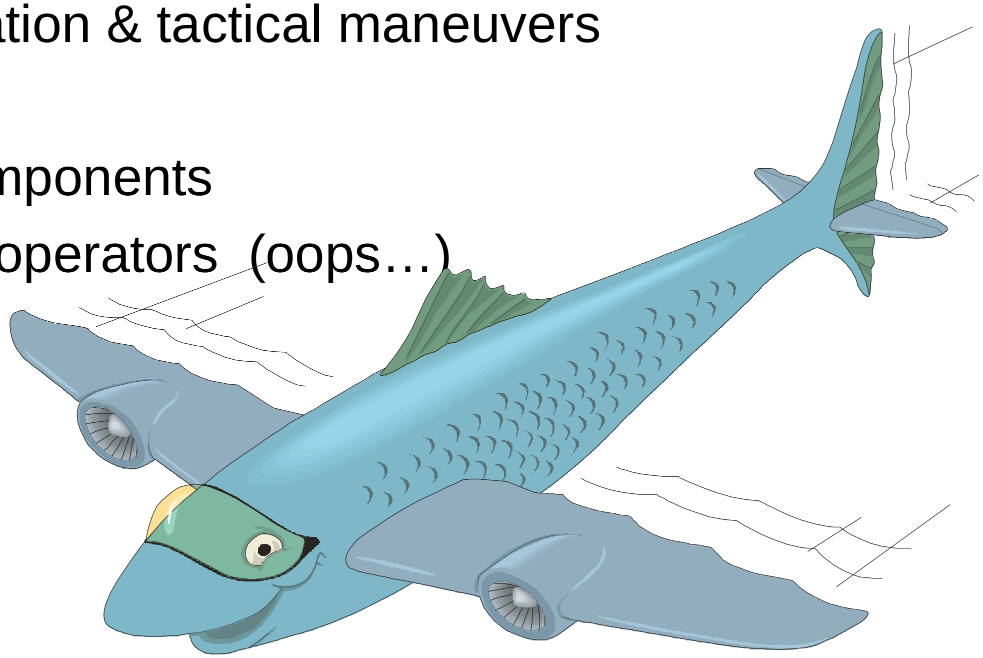
Institute for Complex Engineered Systems

# Outline

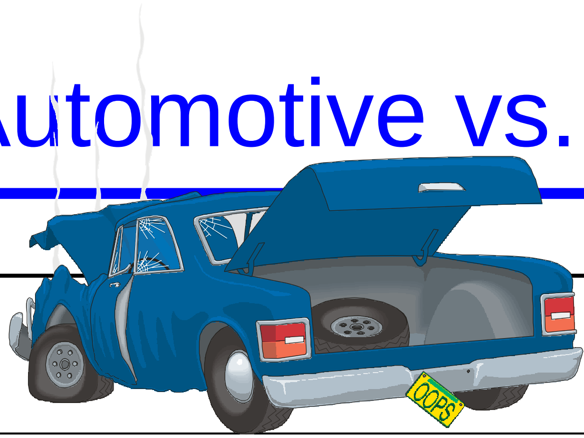
- Why aerospace approaches don’t work on automobiles
- How and why things break
- Designing systems for failure detection & recovery
  - Practical limits of fault tolerant design
  - Environment & other sources of problems
  - How to (and not to) design a highly available system
- Conclusions

# Why Not Build Cars Like Aircraft?

- We all “know” that flying is safer than driving
  - (This is only true from a certain point of view…)
- So, use commercial aircraft techniques to build automated vehicles
  - Computer-controlled navigation & tactical maneuvers
  - Redundant hardware
  - High-quality design and components
  - Highly trained professional operators  (oops…)



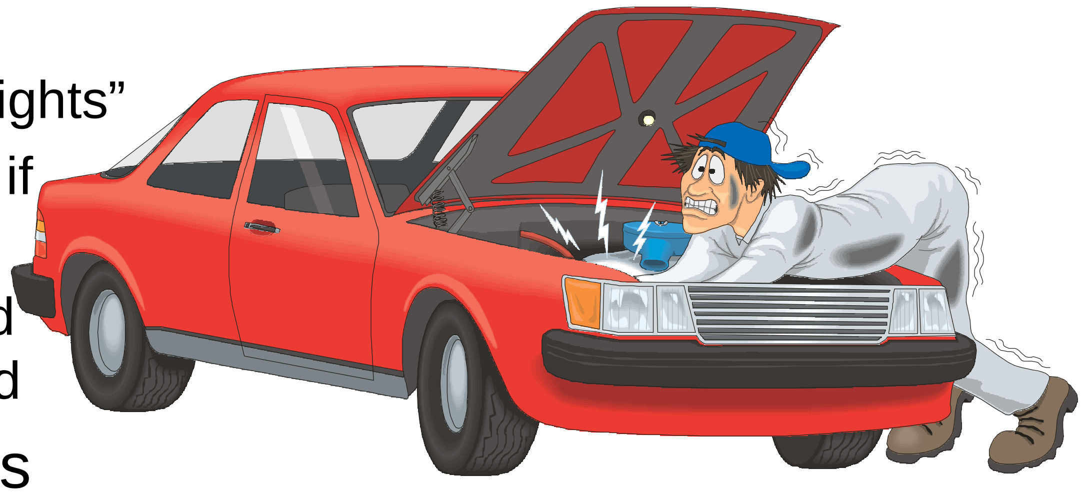
# Automotive vs. Aviation Safety




| | U.S. Automobiles | U.S. Commercial Aircraft |
|---|---|---|
| **Deployed Units** | ~100,000,000 | ~10,000 |
| **Operating hours/year** | ~30,000 Million | ~55 Million |
| **Cost per vehicle** | ~$20,000 | ~$65 Million |
| **Mortalities/year** | 42,000 | ~350 |
| **Accidents/year** | 21 Million | 170 |
| **Mortalities / Million Hours** | 0.71 | 6.4 |
| **Operator Training** | Low | High |
| **Redundancy Levels** | Brakes only | All flight-critical systems |

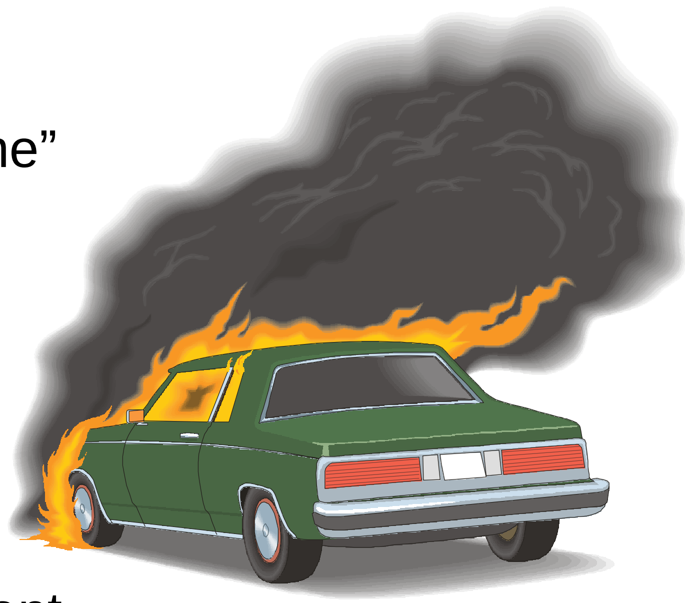
# Why Aerospace Approaches Will Fail

- Too expensive
  - Component “Pain threshold” for vehicles is at the $.05 level
- Different operating environment/reaction time
- Difficult to enforce maintenance
  - People run out of gas & engine oil; ignore “idiot lights”
  - Aircraft don’t leave gate if something is broken
  - End-of-life wearout -- old vehicles stay on the road



- Poorly trained operators
  - Yearly driver exam with road test?
  - Required simulator time for accident response?

# Definitions

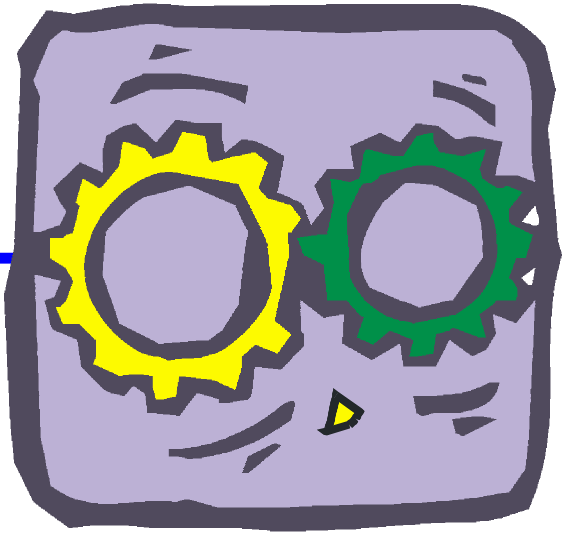


- RELIABILITY -- Aviation model
  – Survival probability for given “mission time”
  – Good when repair is difficult
- AVAILABILITY -- Automotive model
  – The fraction of time a system meets its specification
  – Good when continuous service is important
- DEPENDABILITY
  – Generalization: system does the right thing at the right time

# Fault Classification

- Duration
  - Transient -- design flaws, environmental factors, *etc.*
  - Intermittent -- recurring events
  - Permanent -- “hard” failures/replace component --10% effect
- Extent
  - Local (independent)
  - Distributed (related)
- Value
  - Determinate (stuck-at-high or -low)
  - Indeterminate (varying values)

# Generic Sources of Faults



- Mechanical -- *"wears out"*
  - Deterioration: wear, fatigue, corrosion
  - Shock: fractures, stiction, overload
- Electronic Hardware -- *"bad fabrication; wears out"*
  - Latent manufacturing defects
  - Operating environment: noise, heat, ESD, electro-migration
  - Design defects (Pentium FDIV bug)
- Software -- *"bad design"*
  - Design defects
  - "Code rot" -- accumulated run-time faults
- People
  - Takes a whole additional page...

# How Often Do Components Break?

- Failure rates often expressed in failures / million operating hours ("Lambda" $\lambda$):

| | |
|---|---|
| Military Microprocessor | **0.022** |
| Automotive Microprocessor | **0.12 (1987 data)** |
| Electric Motor | **2.17** |
| Lead/Acid battery | **16.9** |
| Oil Pump | **37.3** |
| Human: single operator best case | **100 (per Mactions)** |
| Automotive Wiring Harness (luxury) | **775** |
| Human: crisis intervention | **300,000 (per Mactions)** |

# Errors By Development Phase

| STAGE | ERROR SOURCES | ERROR DETECTION |
|---|---|---|
| Specification & design | Algorithm Design<br>Formal Specification | Simulation<br>Consistency checks |
| Prototype | Algorithm design<br>Wiring & assembly<br>Timing<br>Component Failure | Stimulus/response<br>Testing |
| Manufacture | Wiring & assembly<br>Component failure | System testing<br>Diagnostics |
| Installation | Assembly<br>Component failure | System Testing<br>Diagnostics |
| Field Operation | Component failure<br>Operator errors<br>Environmental factors | Diagnostics |

# “Mainframe”Outage Sources

| | AT&T Switching System | Bellcore Commercial | Japanese Commercial Users | Tandem 1985 | Tandem 1987 | Northern Telecom | Mainframe Users |
|---|---|---|---|---|---|---|---|
| Hardware | 0.20 | 0.26 | 0.75* | 0.18 | 0.19 | 0.19 | 0.45 |
| Software | 0.15 | 0.30 | 0.75* | 0.26 | 0.43 | 0.19 | 0.20 |
| Maintenance | -- | -- | 0.75* | 0.25 | 0.13 | -- | 0.05 |
| Operations | 0.65 | 0.44 | 0.11 | 0.17 | 0.13 | 0.33 | 0.15 |
| Environment | -- | -- | 0.13 | 0.14 | 0.12 | 0.15 | 0.15 |
| Power | -- | -- | -- | -- | -- | 0.13 | -- |

(* the sum of these sources was 0.75)

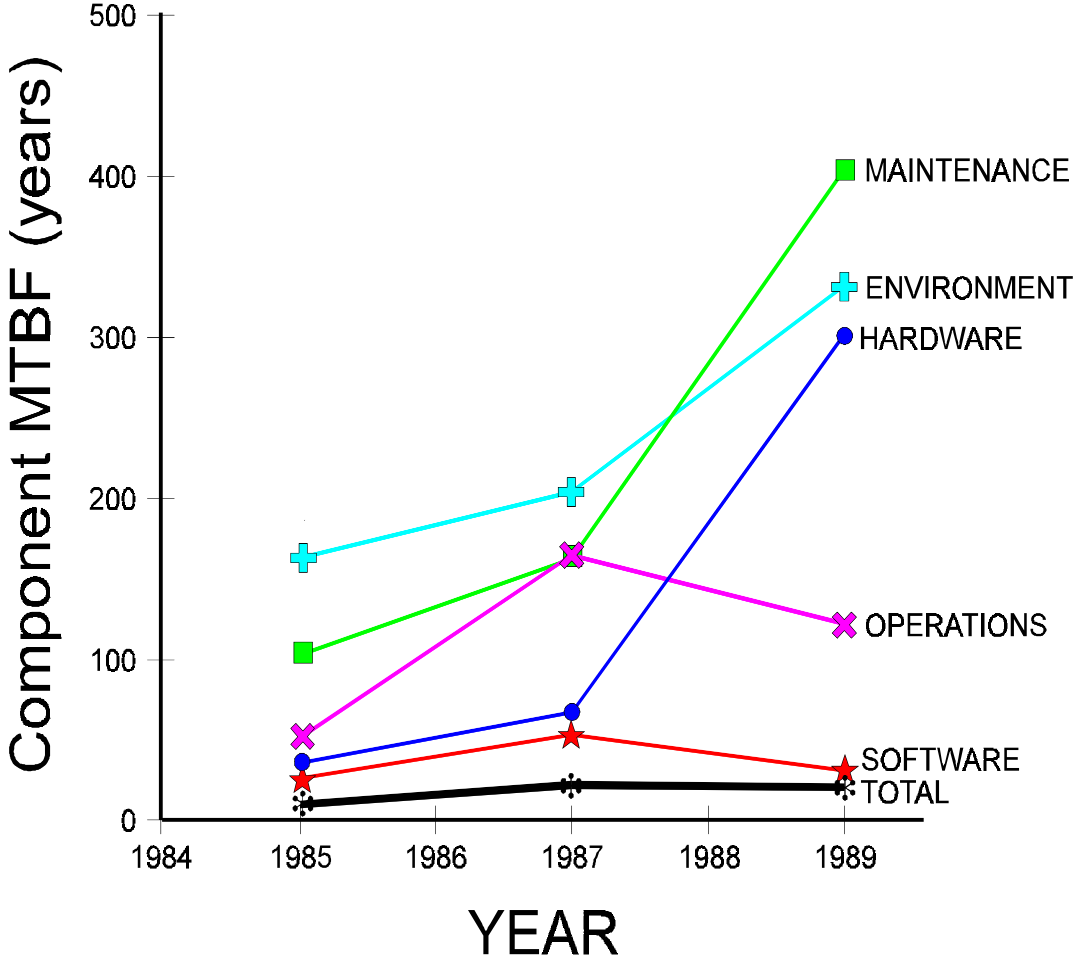
# Tandem Environmental Outages

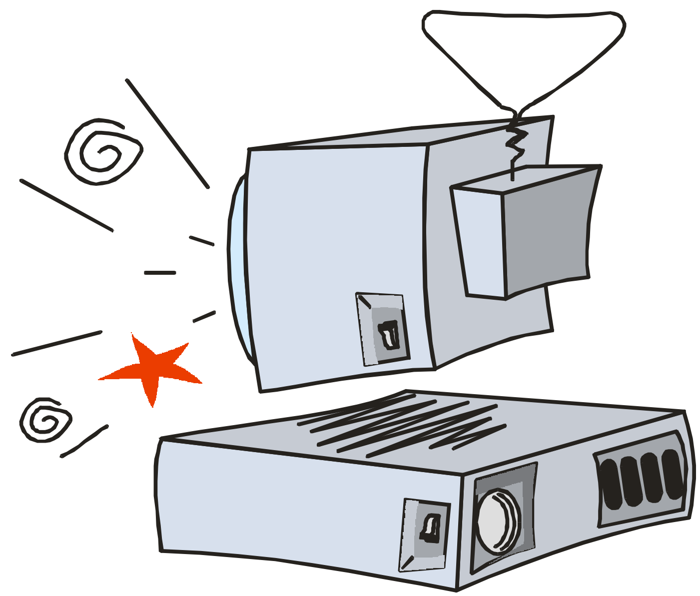
| | |
|---|---|
| ◆ Extended Power Loss | 80% |
| ◆ Earthquake | 5% |
| ◆ Flood | 4% |
| ◆ Fire | 3% |
| ◆ Lightning | 3% |
| ◆ Halon Activation | 2% |
| ◆ Air Conditioning | 2% |

- Total MTBF about 20 years
- MTBAoG* about 100 years
  - Roadside equipment will be more exposed than this

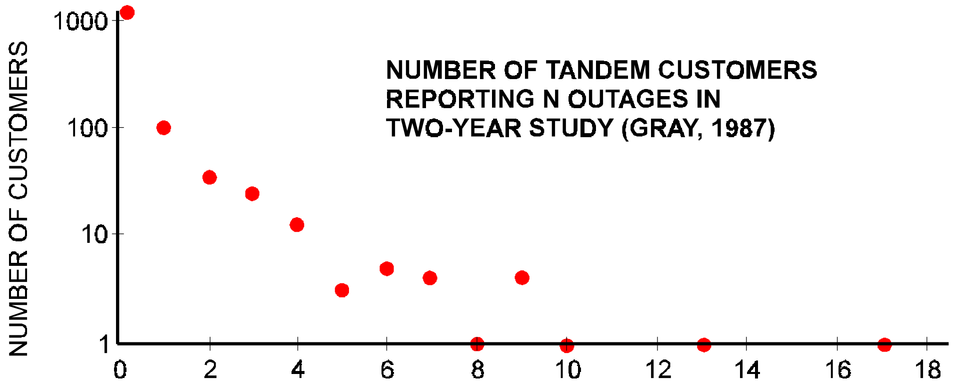
* (AoG= "Act Of God")

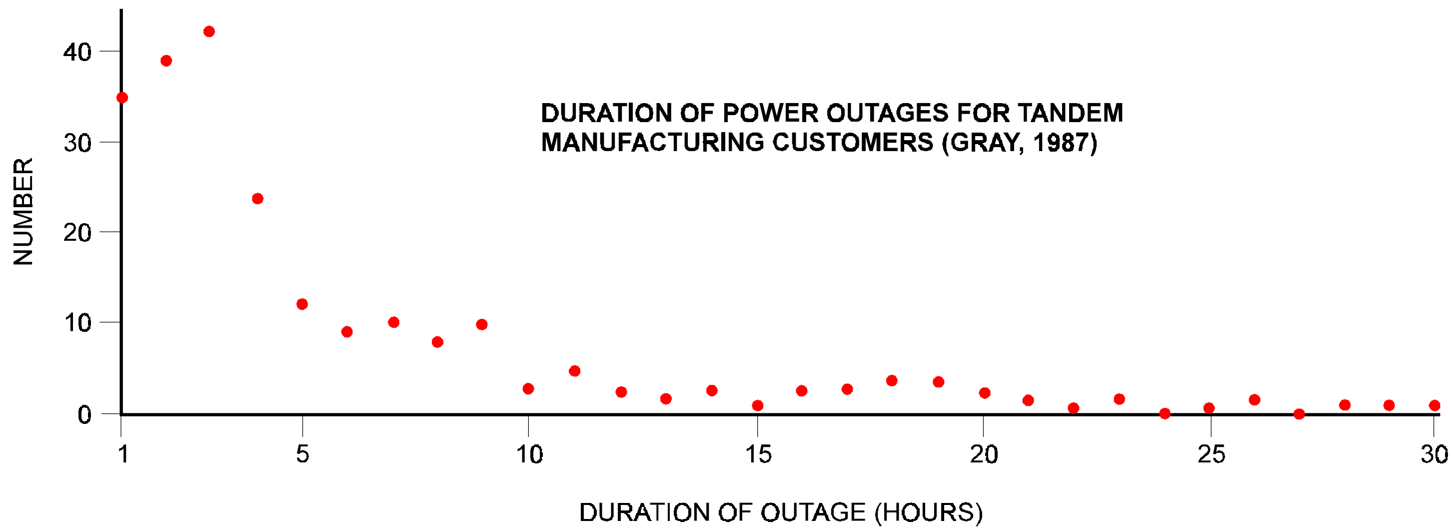
# Tandem Causes of System Failures



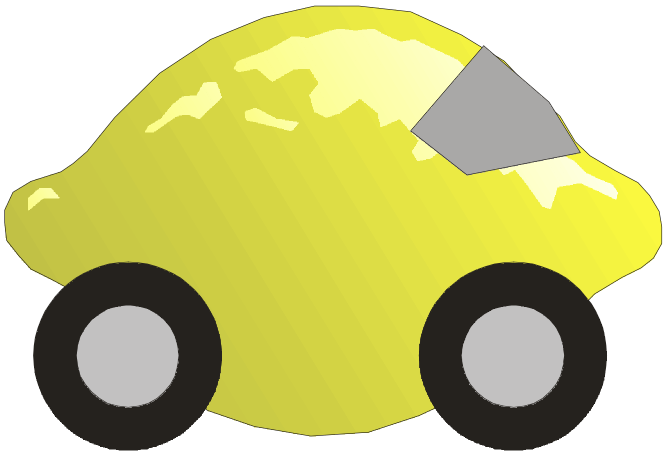



# Tandem Outages



# Lemons Or Just Statistics?



Poisson distributed failures :

$$p(x) = \frac{(\lambda t)^x}{x!} e^{-\lambda t} \quad \mathrm{x} = 0, 1, 2, \ldots$$

| Annual failures for 100,000,000 vehicles | Vehicles failing given 10 year MTBF | Vehicles failing given 100 year MTBF |
|---|---|---|
| 0 | 90,483,741 | 99,004,983 |
| 1 | 9,048,374 | 990,050 |
| 2 | 452,419 | 4,950 |
| 3 | 15,081 | 17 |
| 4 | 377 | 0 |
| 5 | 8 | 0 |
| 6 | 0 | 0 |

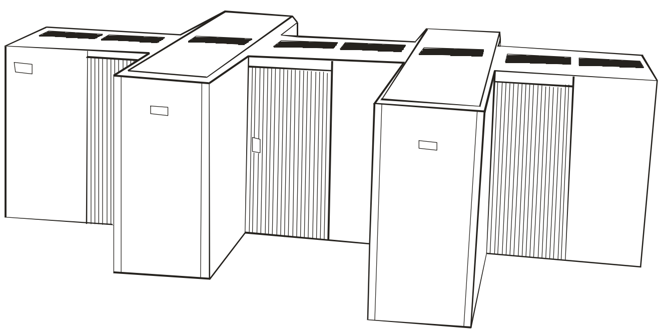
# IBM 3090 Fault Tolerance Features

- Reliability
  - Low intrinsic failure rate technology
  - Extensive component burn-in during manufacture
  - Dual processor controller that incorporates switchover
  - Dual 3370 Direct Access Storage units support switchover
  - Multiple consoles for monitoring processor activity and for backup
  - LSI Packaging vastly reduces number of circuit connections
  - Internal machine power and temperature monitoring
  - Chip sparing in memory replaces defective chips automatically
- Availability
  - Two or tour central processors
  - Automatic error detection and correction in central and expanded storage
  - Single bit error correction and double bit error detection in central storage
  - Double bit error correction and triple bit error detection in expanded storage
  - Storage deallocation in 4K-byte increments under system program control
  - Ability to vary channels off line in one channel increments
  - Instruction retry
  - Channel command retry
  - Error detection and fault isolation circuits provide improved recovery and serviceability
  - Multipath I/O controllers and units

# More IBM 3090 Fault Tolerance

- **Data Integrity**
  - Key controlled storage protection (store and fetch)
  - Critical address storage protection
  - Storage error checking and correction
  - Processor cache error handling
  - Parity and other internal error checking
  - Segment protection (S/370 mode)
  - Page protection (S/370 mode)
  - Clear reset of registers and main storage
  - Automatic Remote Support authorization
  - Block multiplexer channel command retry
  - Extensive I/O recovery by hardware and control programs

- **Serviceability**
  - Automatic fault isolation (analysis routines) concurrent with operation
  - Automatic remote support capability - auto call to IBM if authorized by customer
  - Automatic customer engineer and parts dispatching
  - Trade facilities
  - Error logout recording
  - Microcode update distribution via remote support facilities
  - Remote service console capability
  - Automatic validation tests after repair
  - Customer problem analysis facilities



# IBM 308X/3090 Detection & Isolation

- Hundreds of Thousands of isolation domains
- 25% of IBM 3090 circuits for testability -- only covers 90% of all errors
- Assumed that only 25% of faults are permanent
  - If less than two weeks between events, assume same intermittent source
  - Call service if 24 errors in 2 hours

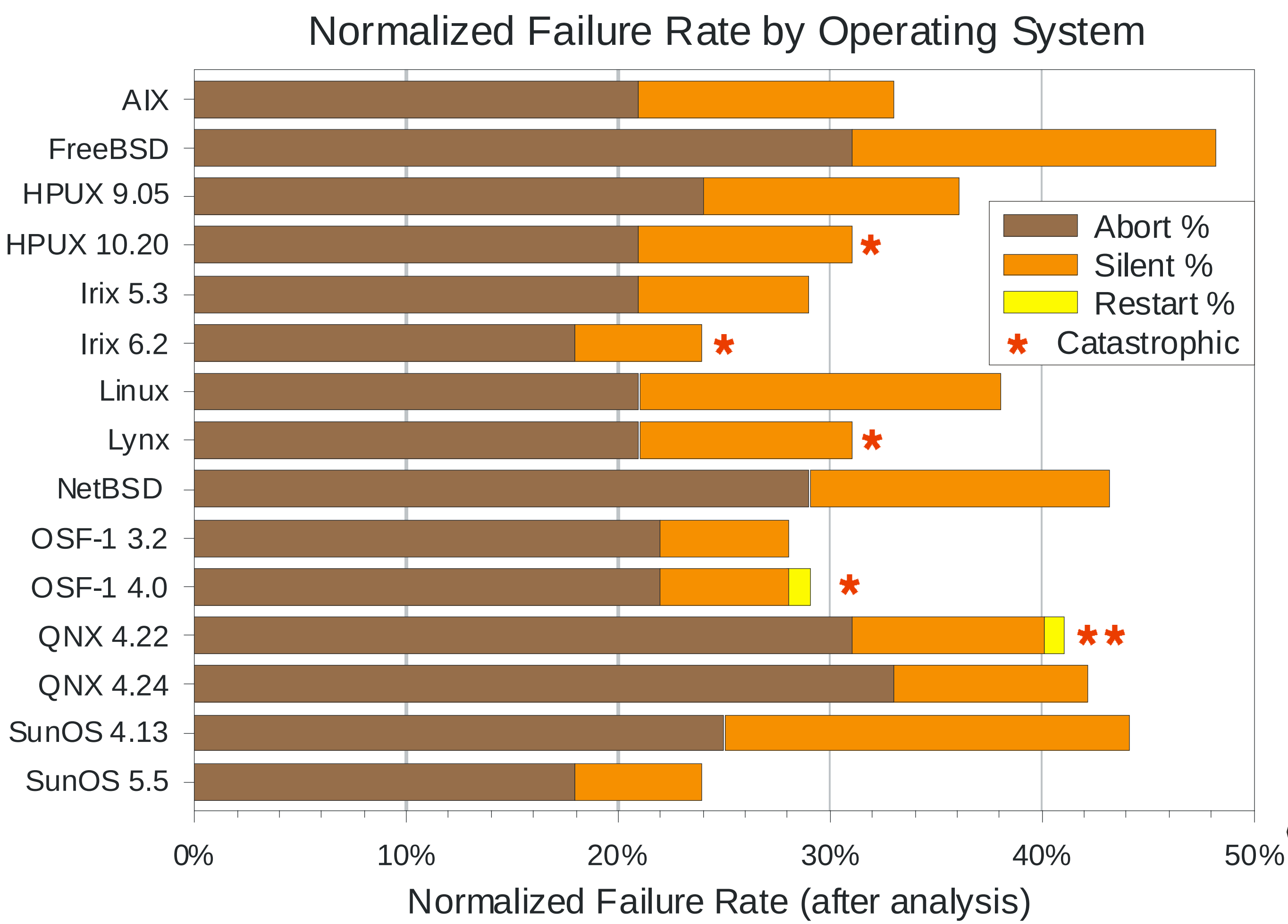
- (Tandem also has 90% FRU diagnosis accuracy)

# Typical PC Hardware ED/FI

this space intentionally blank

# Typical Workstation Software ED/FI

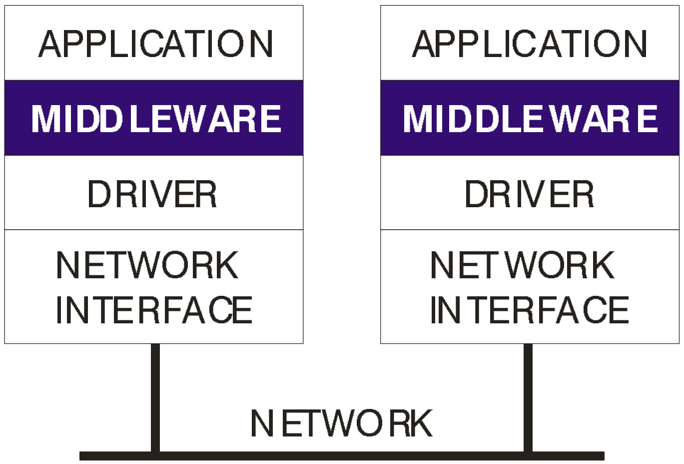
- SW Defects are inevitable -- what happens then?




BALLISTA

# Network Communication Faults

- Communication networks lose bits all the time
  - $10^{-5}$-$10^{-6}$ bit error rate for copper in *workstation* networks
  - $10^{-12}$-$10^{-14}$ bit error rate for fiber (not including xmit/receive)
- What happens when messages are lost or altered?
  - Messages can be lost frequently
    - » ~300 per hour at $10^{-5}$ ber; worse if there are noise bursts
  - Double bit errors can cause messages to get past CAN CRC
    - » 100M vehicles at $10^{-5}$ ber results in ~130 events/year in US fleet
    - » 100M vehicles at $10^{-4}$ ber results in ~13,000 events/year in US fleet
- Research area:
  fault injection middleware



# Conclusions

- Design dependability into the system, not on top of the system
  - Take domain constraints into account when choosing approach
- Historically, goals of 100% unattainable for:
  - Fault detection/isolation
  - Availability
  - Design correctness
- The biggest risk items are people & software

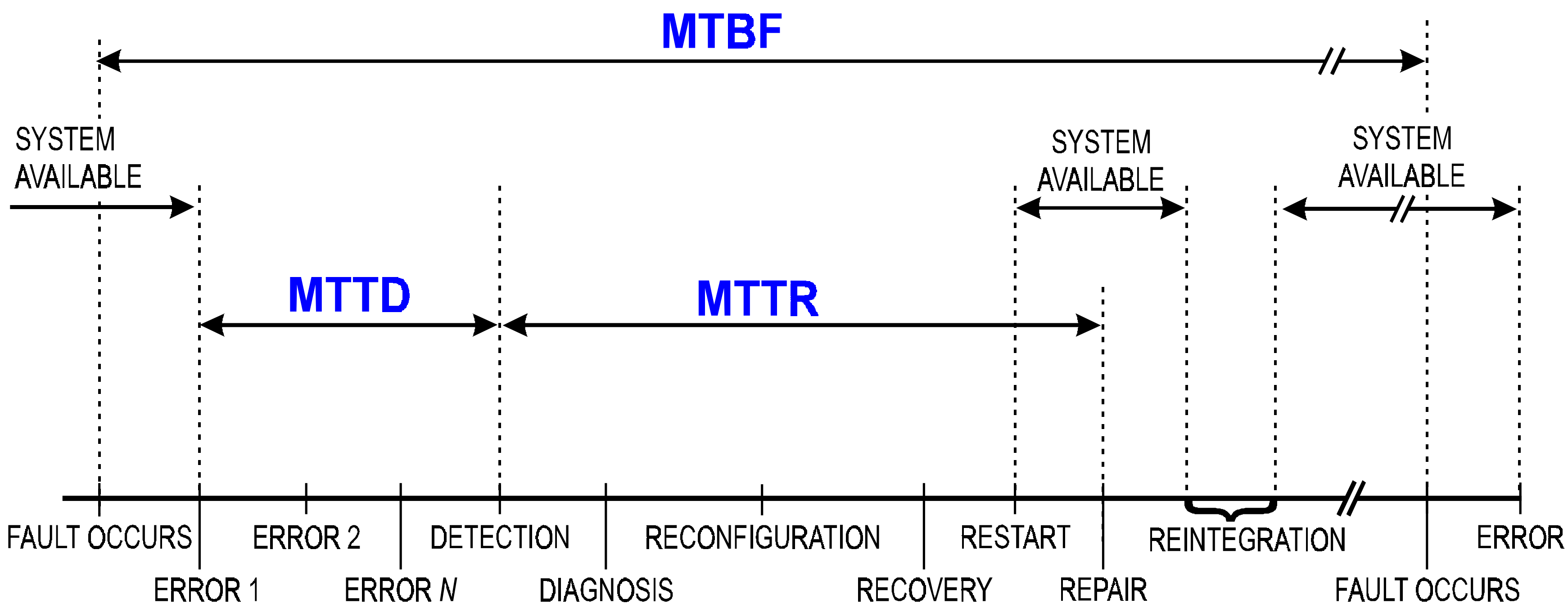
# Cause-Effect Sequence

- FAULT: deviation of function from design value
  - Hardware
  - Software
  - Electromechanical

- ERROR: manifestation of fault by incorrect value

- FAILURE: deviation of system from specification

# MTBF -- MTTD -- MTTR

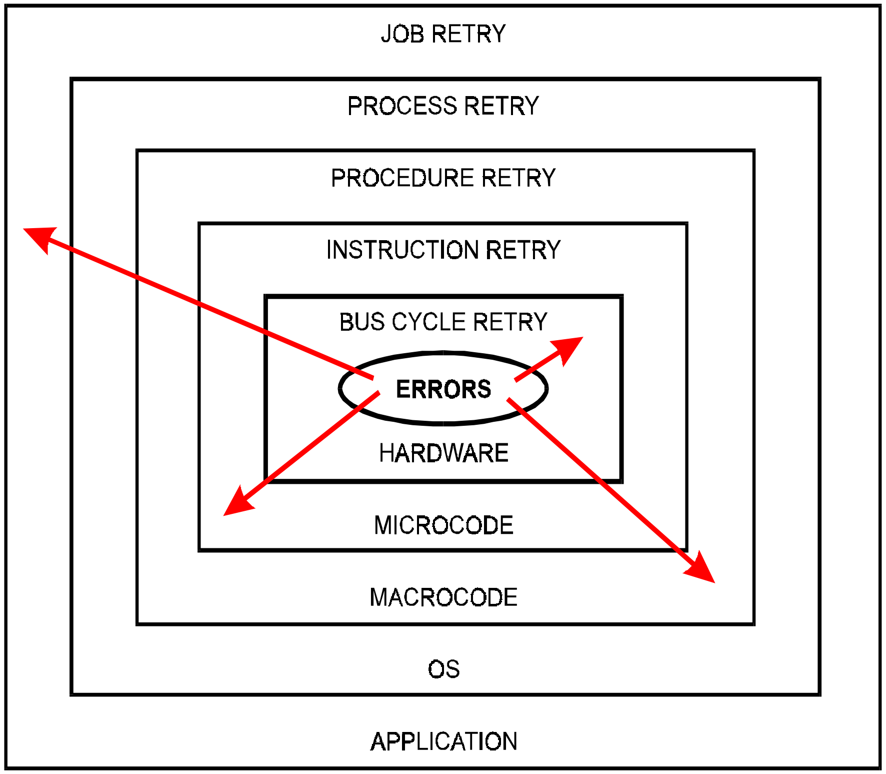



A Scenario for on-line detection and off-line repair. The measures -- MTBF, MTTD, and MTTR are the average times to failure, to detection, and to repair.

# Basic Steps in Fault Handling

- Fault Confinement
- Fault Detection
- Fault Masking
- Retry
- Diagnosis
- Reconfiguration
- Recovery
- Restart
- Repair
- Reintegration

# Error Containment Levels




The further out the error propagates, the more state is involved and the more diverse error manifestations becomes, resulting in more complex error recovery.