# Dependability – Concepts*, State-of-the-Art, Challenges

Jean-Claude Laprie

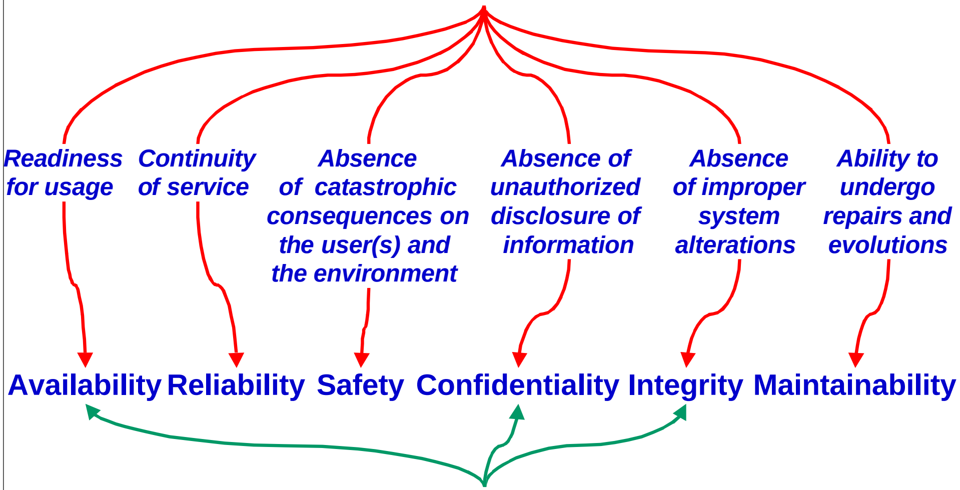
LAAS

* Based on
A. Avizienis (UCLA), J.C. Laprie, B. Randell (Univ. Of Newcastle upon Tyne): *Fundamental Concepts of Dependability*

Critical Systems Conference — Birmingham, October 23-24, 2001

## Dependability

ability to deliver service that can justifiably be trusted



*Readiness for usage*

*Continuity of service*

*Absence of catastrophic consequences on the user(s) and the environment*

*Absence of unauthorized disclosure of information*

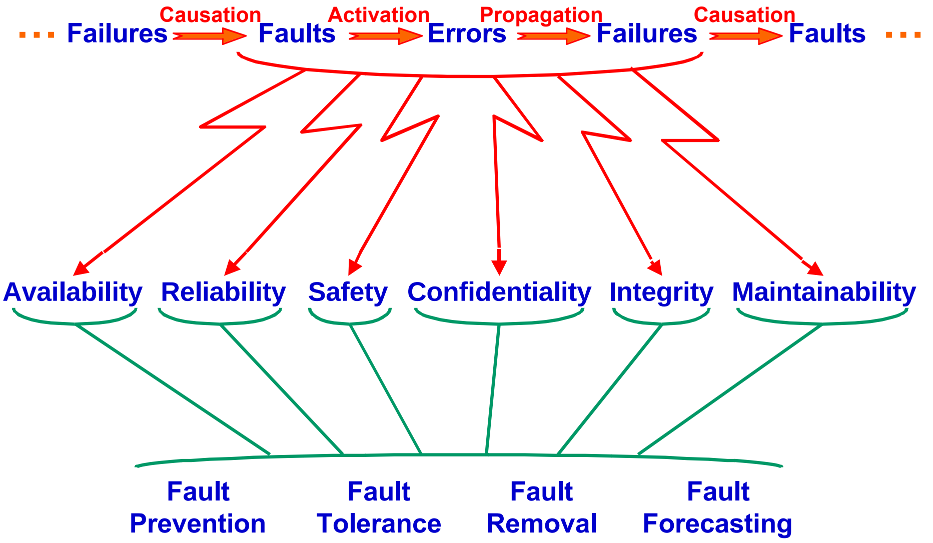
*Absence of improper system alterations*

*Ability to undergo repairs and evolutions*

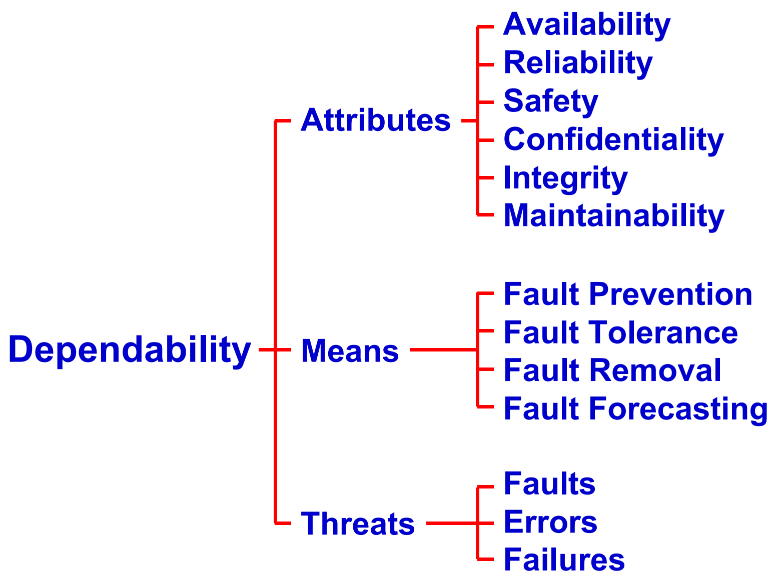
Availability Reliability Safety Confidentiality Integrity Maintainability

Security

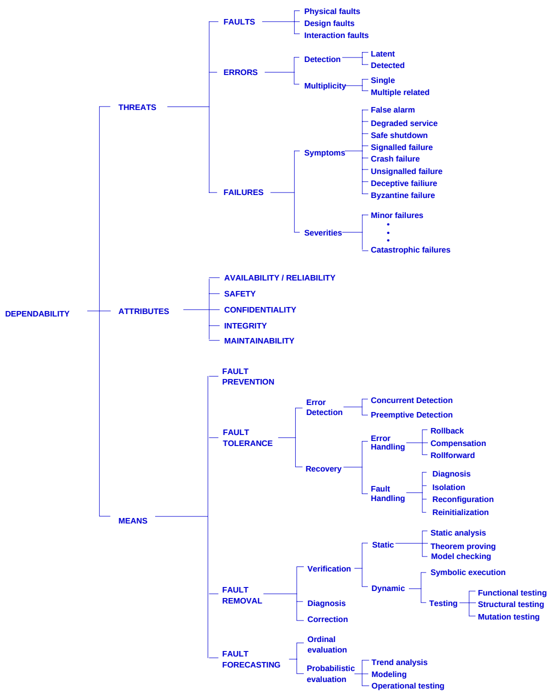
Absence of unauthorized access to, or handling of, system state



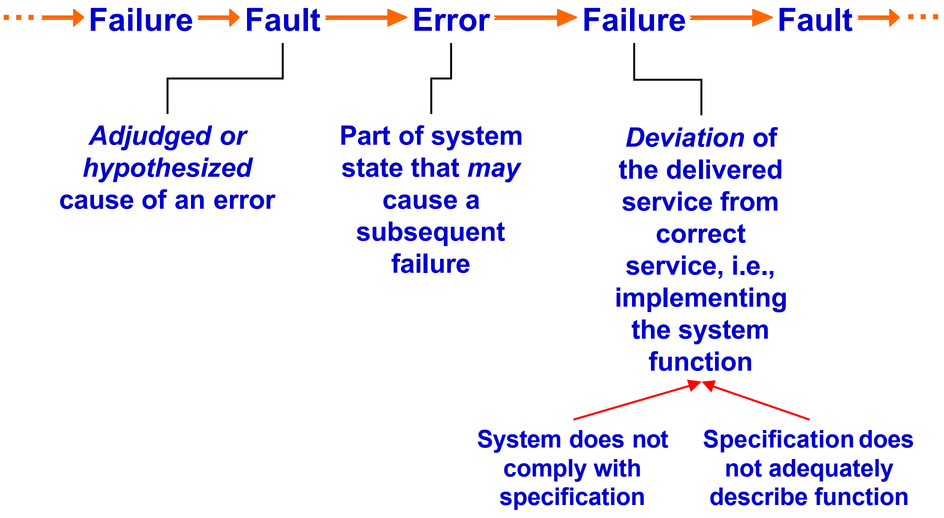
**Dependability**: ability to avoid failures that are more frequent or more severe, and outage durations that are longer, than is acceptable to the user(s)



- Dependability
  - Attributes
    - Availability
    - Reliability
    - Safety
    - Confidentiality
    - Integrity
    - Maintainability
  - Means
    - Fault Prevention
    - Fault Tolerance
    - Fault Removal
    - Fault Forecasting
  - Threats
    - Faults
    - Errors
    - Failures



- DEPENDABILITY
  - THREATS
    - FAULTS
      - Physical faults
      - Design faults
      - Interaction faults
    - ERRORS
      - Detection
        - Latent
        - Detected
      - Multiplicity
        - Single
        - Multiple related
    - FAILURES
      - Symptoms
        - False alarm
        - Degraded service
        - Safe shutdown
        - Signalled failure
        - Crash failure
        - Unsignalled failure
        - Deceptive faiilure
        - Byzantine failure
      - Severities
        - Minor failures
        - •
        - •
        - •
        - Catastrophic failures
  - ATTRIBUTES
    - AVAILABILITY / RELIABILITY
    - SAFETY
    - CONFIDENTIALITY
    - INTEGRITY
    - MAINTAINABILITY
  - MEANS
    - FAULT PREVENTION
    - FAULT TOLERANCE
      - Error Detection
        - Concurrent Detection
        - Preemptive Detection
      - Recovery
        - Error Handling
          - Rollback
          - Compensation
          - Rollforward
        - Fault Handling
          - Diagnosis
          - Isolation
          - Reconfiguration
          - Reinitialization
    - FAULT REMOVAL
      - Verification
        - Static
          - Static analysis
          - Theorem proving
          - Model checking
        - Dynamic
          - Symbolic execution
          - Testing
            - Functional testing
            - Structural testing
            - Mutation testing
      - Diagnosis
      - Correction
    - FAULT FORECASTING
      - Ordinal evaluation
      - Probabilistic evaluation
        - Trend analysis
        - Modeling
        - Operational testing



··· → Failure → Fault → Error → Failure → Fault → ···

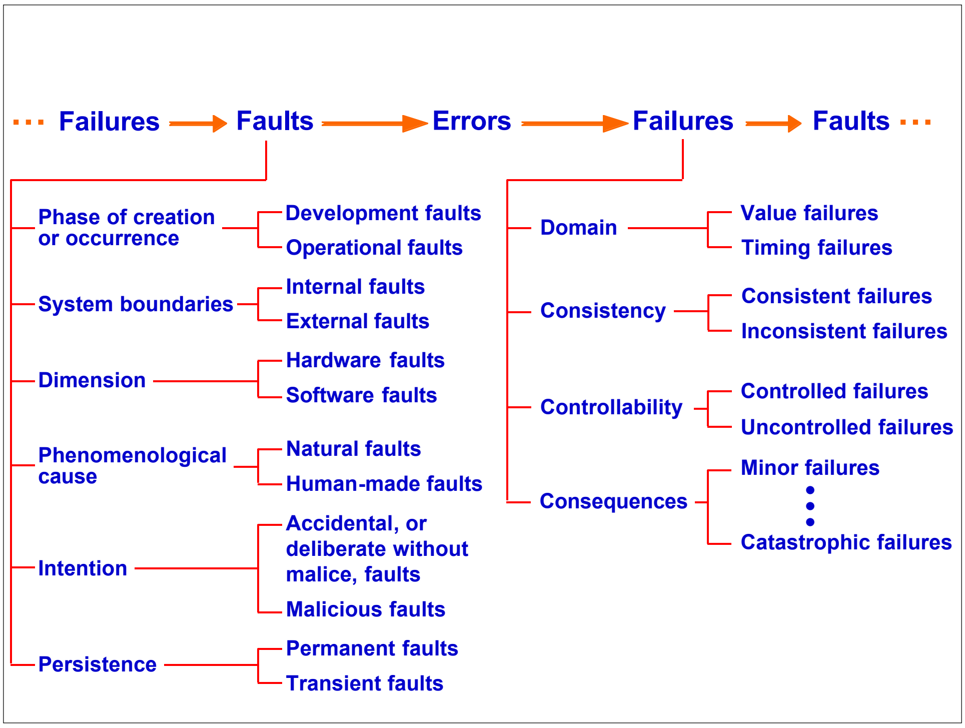
***Adjudged or hypothesized*** **cause of an error**

**Part of system state that *may* cause a subsequent failure**

***Deviation* of the delivered service from correct service, i.e., implementing the system function**

**System does not comply with specification**

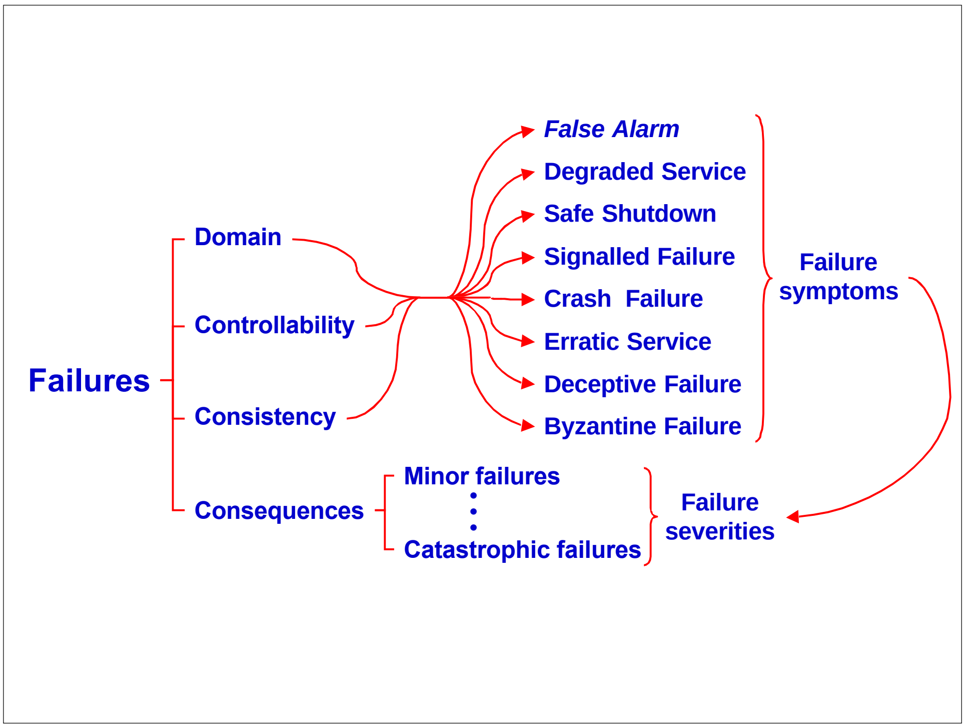
**Specification does not adequately describe function**



··· Failures → Faults → Errors → Failures → Faults ···

- Phase of creation or occurrence
  - Development faults
  - Operational faults
- System boundaries
  - Internal faults
  - External faults
- Dimension
  - Hardware faults
  - Software faults
- Phenomenological cause
  - Natural faults
  - Human-made faults
- Intention
  - Accidental, or deliberate without malice, faults
  - Malicious faults
- Persistence
  - Permanent faults
  - Transient faults

- Domain
  - Value failures
  - Timing failures
- Consistency
  - Consistent failures
  - Inconsistent failures
- Controllability
  - Controlled failures
  - Uncontrolled failures
- Consequences
  - Minor failures
  - ⋮
  - Catastrophic failures



Failures

- Domain
- Controllability
- Consistency
- Consequences

*False Alarm*

Degraded Service

Safe Shutdown

Signalled Failure

Crash Failure

Erratic Service

Deceptive Failure

Byzantine Failure

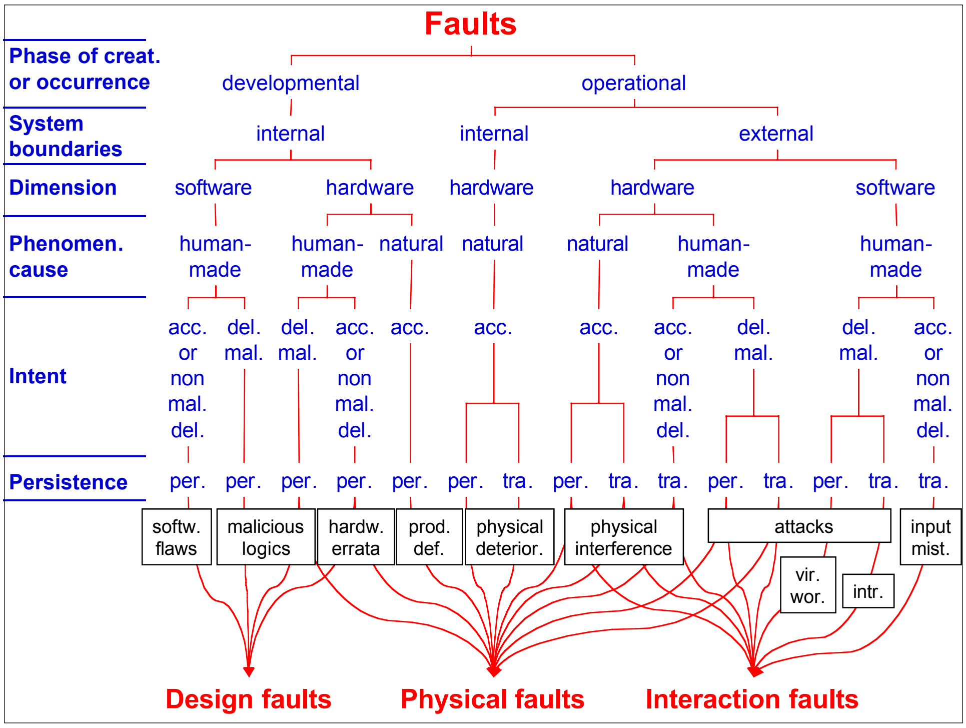
Failure symptoms

Minor failures

⋮

Catastrophic failures

Failure severities



Faults

Phase of creat. or occurrence: developmental; operational

System boundaries: internal; internal; external

Dimension: software; hardware; hardware; hardware; software

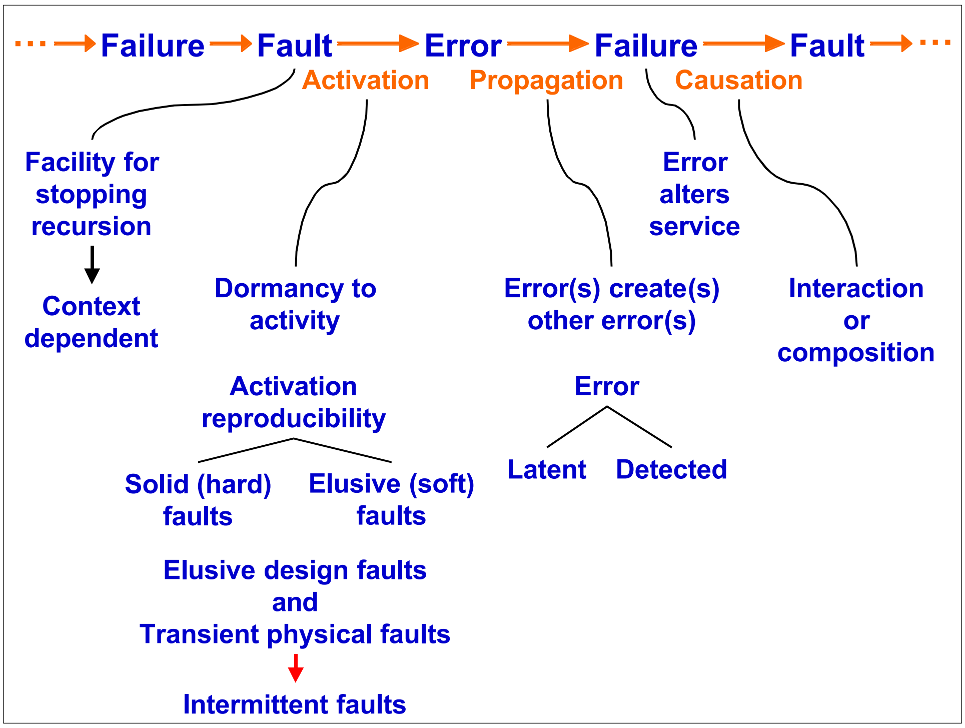
Phenomen. cause: human-made; human-made; natural; natural; natural; human-made; human-made

Intent: acc. or non mal. del.; del. mal.; del. mal.; acc. or non mal. del.; acc.; acc.; acc.; acc. or non mal. del.; del. mal.; del. mal.; acc. or non mal. del.

Persistence: per.; per.; per.; per.; per.; per.; tra.; per.; tra.; tra.; per.; tra.; per.; tra.; tra.

softw. flaws; malicious logics; hardw. errata; prod. def.; physical deterior.; physical interference; attacks; vir. wor.; intr.; input mist.

Design faults; Physical faults; Interaction faults



··· → Failure → Fault → Error → Failure → Fault → ···

Activation; Propagation; Causation

Facility for stopping recursion ↓ Context dependent

Dormancy to activity

Error alters service

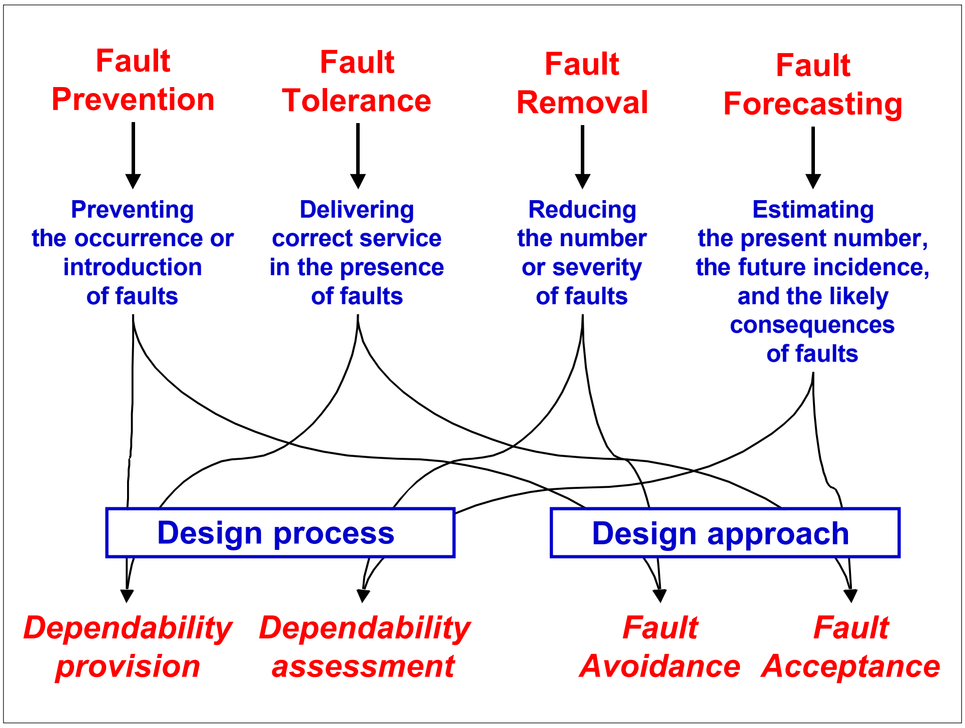
Error(s) create(s) other error(s)

Interaction or composition

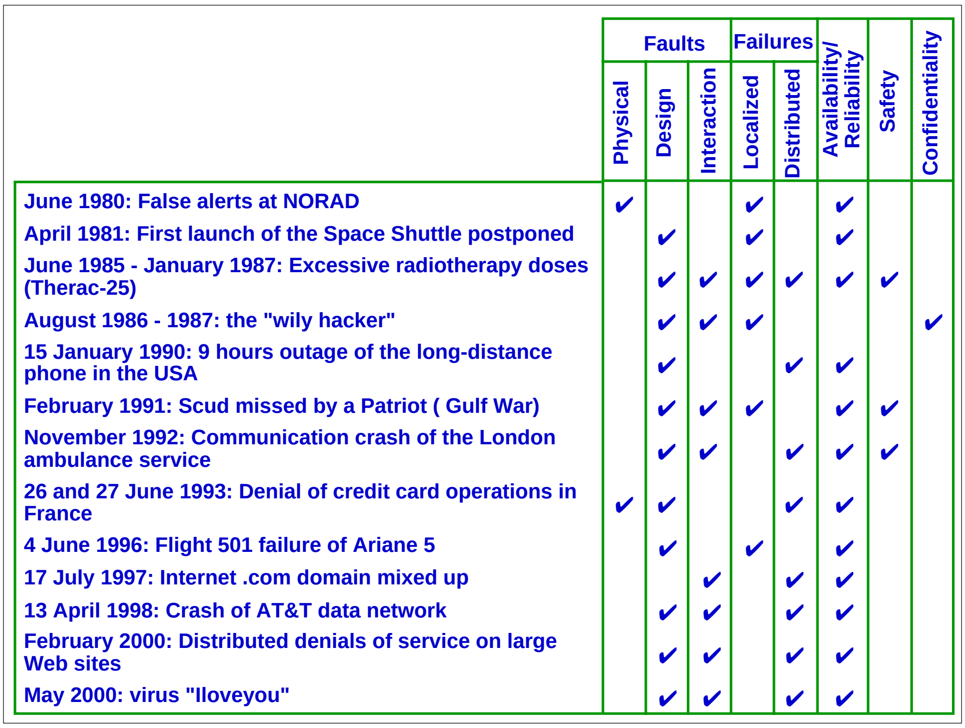
Activation reproducibility: Solid (hard) faults; Elusive (soft) faults

Error: Latent; Detected

Elusive design faults and Transient physical faults ↓ Intermittent faults



Fault Prevention
Fault Tolerance
Fault Removal
Fault Forecasting
Preventing the occurrence or introduction of faults
Delivering correct service in the presence of faults
Reducing the number or severity of faults
Estimating the present number, the future incidence, and the likely consequences of faults
Design process
Design approach
Dependability provision
Dependability assessment
Fault Avoidance
Fault Acceptance



| | Faults | | | Failures | | | | |
|---|---|---|---|---|---|---|---|---|
| | Physical | Design | Interaction | Localized | Distributed | Availability/ Reliability | Safety | Confidentiality |
| June 1980: False alerts at NORAD | ✓ | | | ✓ | | ✓ | | |
| April 1981: First launch of the Space Shuttle postponed | | ✓ | | ✓ | | ✓ | | |
| June 1985 - January 1987: Excessive radiotherapy doses (Therac-25) | | ✓ | ✓ | ✓ | ✓ | ✓ | ✓ | |
| August 1986 - 1987: the "wily hacker" | | ✓ | ✓ | ✓ | | | | ✓ |
| 15 January 1990: 9 hours outage of the long-distance phone in the USA | | ✓ | | | ✓ | ✓ | | |
| February 1991: Scud missed by a Patriot ( Gulf War) | | ✓ | ✓ | ✓ | | ✓ | ✓ | |
| November 1992: Communication crash of the London ambulance service | | ✓ | ✓ | | ✓ | ✓ | ✓ | |
| 26 and 27 June 1993: Denial of credit card operations in France | ✓ | ✓ | | | ✓ | ✓ | | |
| 4 June 1996: Flight 501 failure of Ariane 5 | | ✓ | | ✓ | | ✓ | | |
| 17 July 1997: Internet .com domain mixed up | | | ✓ | | ✓ | ✓ | | |
| 13 April 1998: Crash of AT&T data network | | ✓ | ✓ | | ✓ | ✓ | | |
| February 2000: Distributed denials of service on large Web sites | | ✓ | ✓ | | ✓ | ✓ | | |
| May 2000: virus "Iloveyou" | | ✓ | ✓ | | ✓ | ✓ | | |

## Accidental (and non-malicious deliberate) faults

| Number of failures<br>[consequences and outage durations highly-application dependent] | Computer systems<br>(e.g. Transactions, Electronic switching) | | Larger, controlled, systems<br>(e.g. Commercial airplanes; telephone network) | |
|---|---|---|---|---|
| | Rank | Proportion | Rank | Proportion |
| Physical internal | 3 | ~ 10% | 2 | 15-20% |
| Physical external | 3 | ~ 10% | 2 | 15-20% |
| Human-machine interaction * | 2 | ~ 20% | 1 | 40-50% |
| Design | 1 | ~ 60% | 2 | 15-20% |

* Forensics evidence that interaction faults can often be traced back to design faults

| Persistence | Solid | Intermittent |
|---|---|---|
| Physical and design | ~ 10% | ~ 90% |

## Deliberately malicious faults

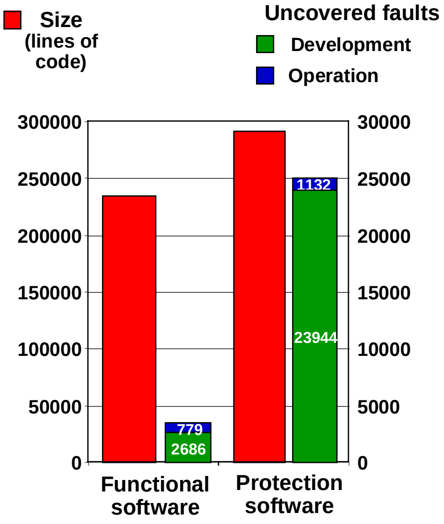
[Ernst & Young, 1998 ; 1200 companies in 32 countries]

Companies having experienced frauds during the last 12 months

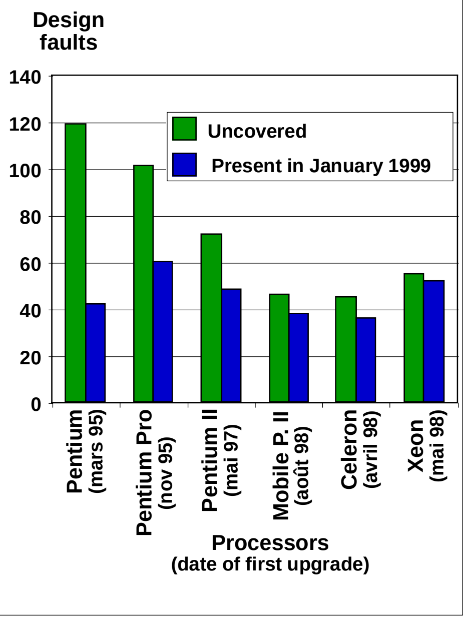
one at least: 66 %  more than 5: 17 %

+ 85 % of frauds by employees

Upgrades of AT&T ESS-5

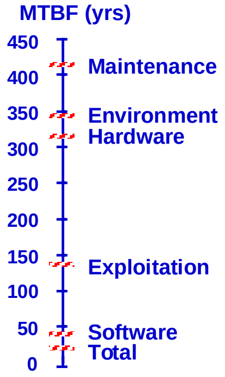


Intel processors (UCLA)



## Tandem Fault Tolerant Systems

| | Number | Duration (yrs) |
|---|---|---|
| Clients | 2000 | 7000 |
| Systems | 9000 | 30000 |
| Processors | 25500 | 80000 |
| Disks | 74000 | 200000 |
| Reprted outages | | 438 |
| MTBF System | | 21 yrs |



## FAA Airworthiness Directives (AD)
## (Brooklyn Polytechnic University)

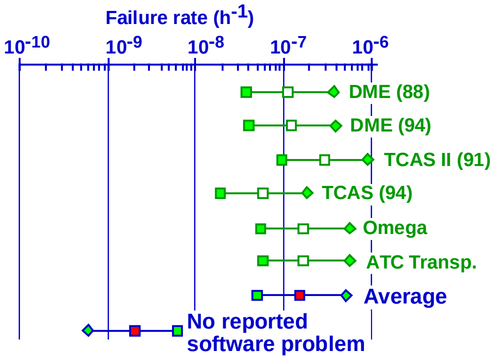
January 1st, 1980 - Sept. 21, 1994

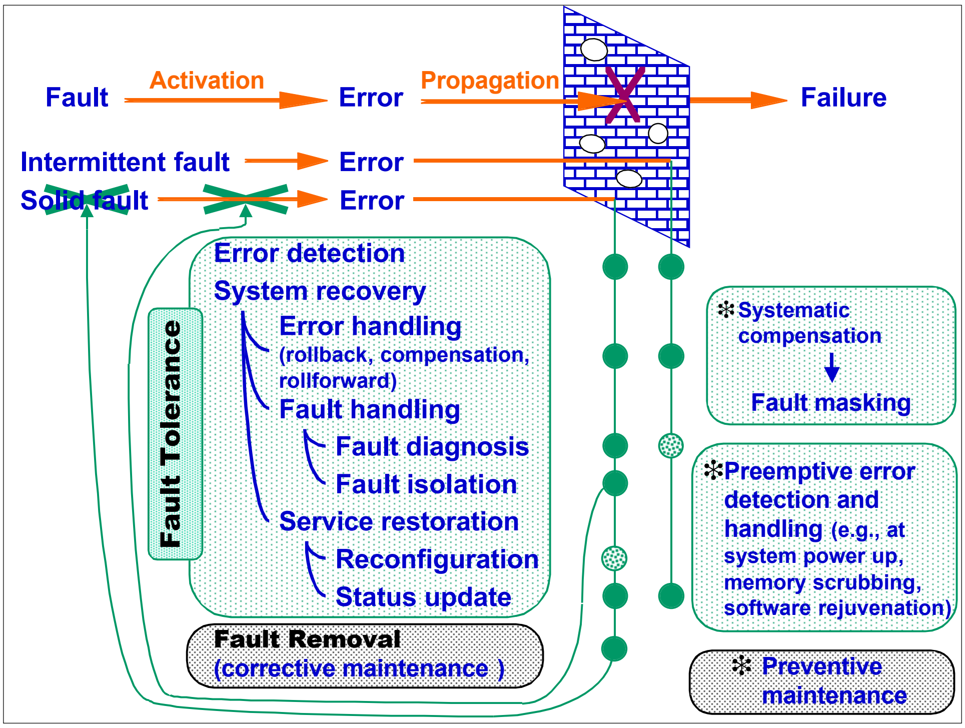
Confirmed avionics AD : 33

Hardware : 20 Software : 13

| Equipments | Rockwell/Collins<br>Bendix/King<br>Honeywell/Sperry<br>Tracor Aerospace |
|---|---|

Estimation of software reliability





Fault

Activation

Error

Propagation

Failure

Intermittent fault

Error

~~Solid fault~~

Error

Error detection

System recovery

Error handling

(rollback, compensation, rollforward)

Fault handling

Fault diagnosis

Fault isolation

Service restoration

Reconfiguration

Status update

Fault Tolerance

**Fault Removal**

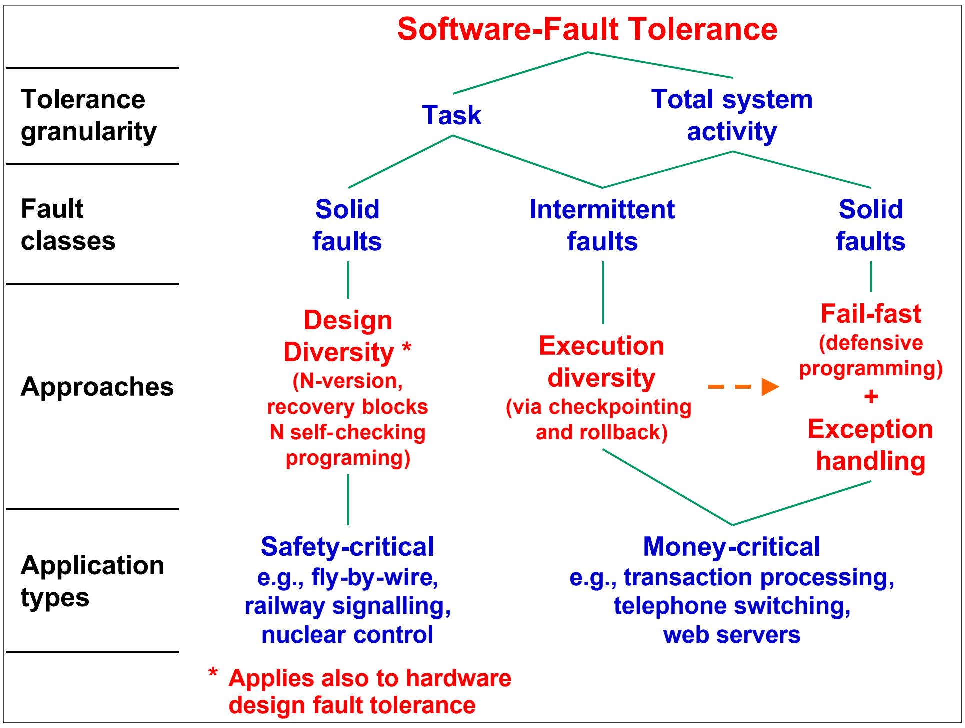
(corrective maintenance )

✻Systematic compensation

Fault masking

✻Preemptive error detection and handling (e.g., at system power up, memory scrubbing, software rejuvenation)

✻ Preventive maintenance



## Software-Fault Tolerance

Tolerance granularity

Task

Total system activity

Fault classes

Solid faults

Intermittent faults

Solid faults

Approaches

Design Diversity * (N-version, recovery blocks N self-checking programing)

Execution diversity (via checkpointing and rollback)

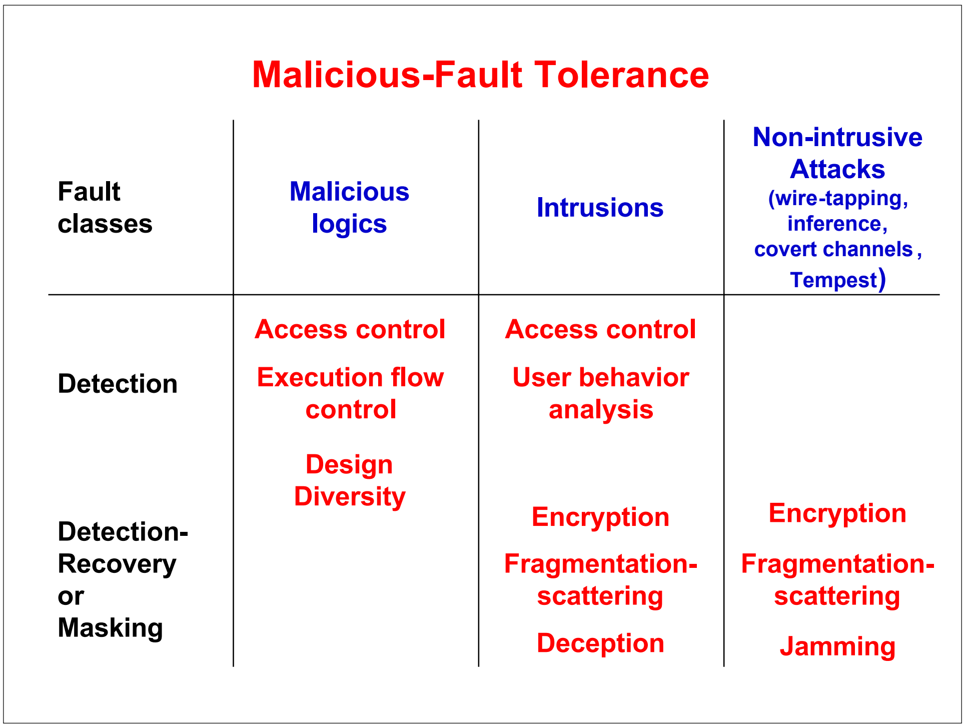
Fail-fast (defensive programming) + Exception handling

Application types

Safety-critical e.g., fly-by-wire, railway signalling, nuclear control

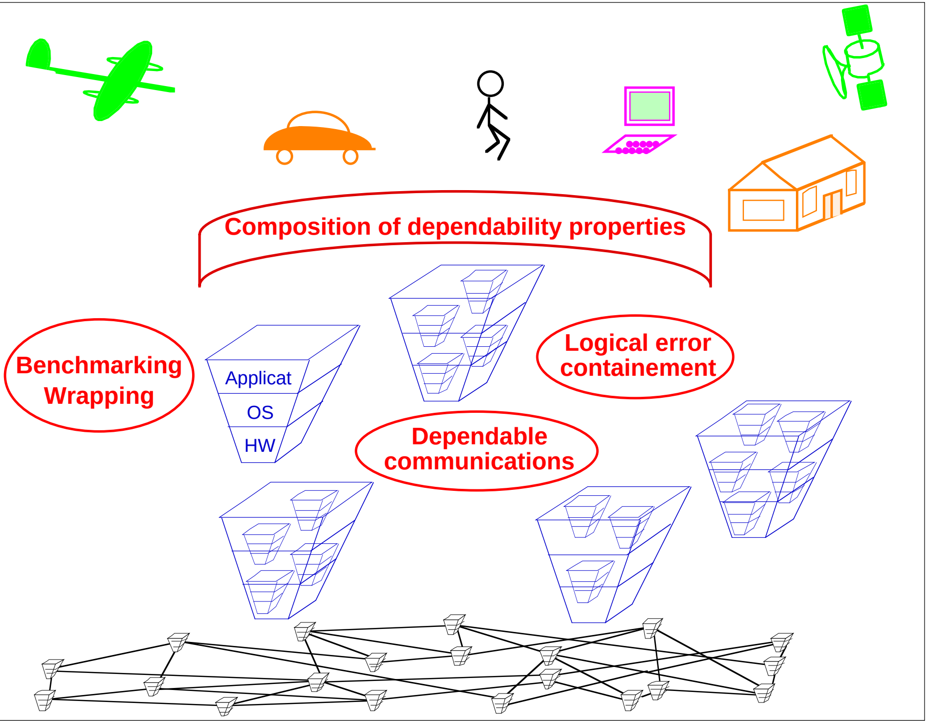
Money-critical e.g., transaction processing, telephone switching, web servers

* Applies also to hardware design fault tolerance



## Malicious-Fault Tolerance

| | | | |
|---|---|---|---|
| Fault classes | Malicious logics | Intrusions | Non-intrusive Attacks (wire-tapping, inference, covert channels , Tempest) |
| Detection | Access control<br>Execution flow control<br>Design Diversity | Access control<br>User behavior analysis | |
| Detection-Recovery or Masking | | Encryption<br>Fragmentation-scattering<br>Deception | Encryption<br>Fragmentation-scattering<br>Jamming |



Composition of dependability properties

Benchmarking Wrapping

Applicat

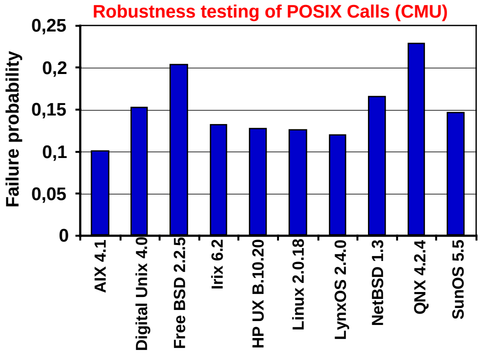
OS

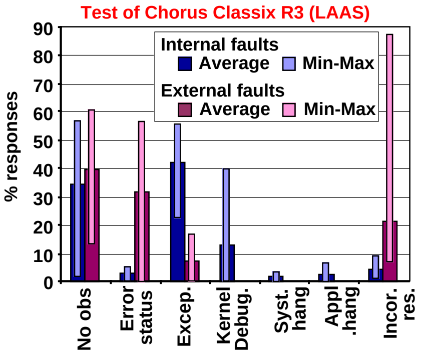
HW

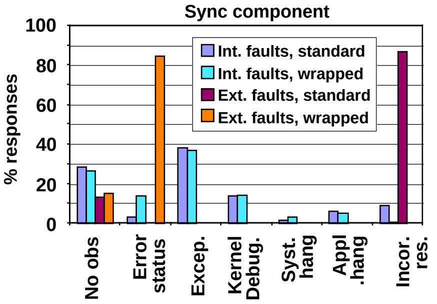
Logical error containement

Dependable communications

## (C)OTS

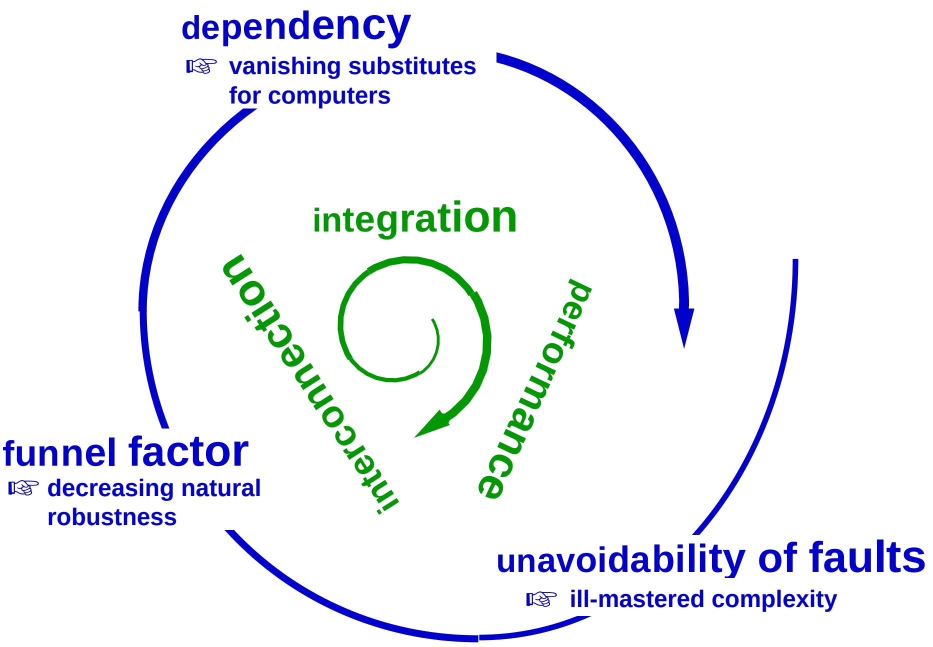






## Dependability

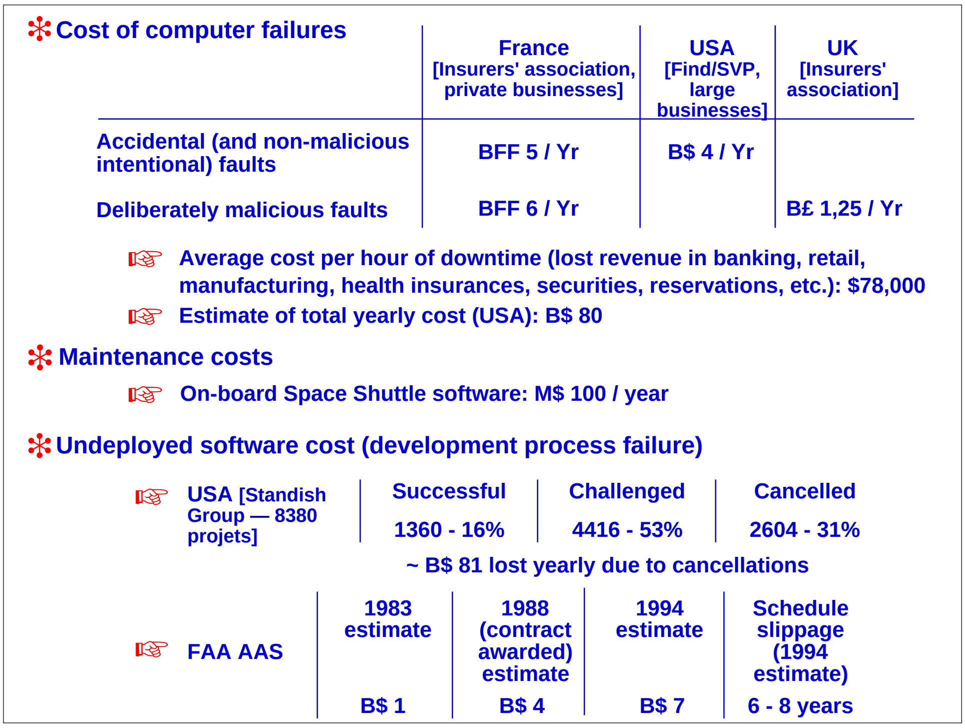
- Subsumes concerns in reliability, availability, safety, confidentiality, integrity, maintenability — the *attributes of dependability* — within a unified conceptual framework; enables the appropriate balance between the attributes to be addressed
- *Means for dependability* — fault prevention, fault tolerance, fault removal, fault forecasting — provide an orthogonal classification of development activities; essential for abstract and discrete systems (nonexistent or vanishing safety factor)
- Causal chain of *threats to dependability* — fault - error - failure
  - Central to understanding and mastering various threats likely to affect a system
  - Provides for a unified presentation of those threats, though preserving their specificities via the various classes
- Rigorous terminology — not just definitions: a model
  - abstraction structuration recursion
  - Avoiding intellectual confusion(s)
  - Focusing on scientific problems and technical choices



dependency
☞ vanishing substitutes for computers

funnel factor
☞ decreasing natural robustness

unavoidability of faults
☞ ill-mastered complexity



## Cost of computer failures

| | France<br>[Insurers' association, private businesses] | USA<br>[Find/SVP, large businesses] | UK<br>[Insurers' association] |
|---|---|---|---|
| Accidental (and non-malicious intentional) faults | BFF 5 / Yr | B$ 4 / Yr | |
| Deliberately malicious faults | BFF 6 / Yr | | B£ 1,25 / Yr |

- Average cost per hour of downtime (lost revenue in banking, retail, manufacturing, health insurances, securities, reservations, etc.): $78,000
- Estimate of total yearly cost (USA): B$ 80

## Maintenance costs

- On-board Space Shuttle software: M$ 100 / year

## Undeployed software cost (development process failure)

| | Successful | Challenged | Cancelled |
|---|---|---|---|
| USA [Standish Group — 8380 projets] | 1360 - 16% | 4416 - 53% | 2604 - 31% |

~ B$ 81 lost yearly due to cancellations

| | 1983 estimate | 1988 (contract awarded) estimate | 1994 estimate | Schedule slippage (1994 estimate) |
|---|---|---|---|---|
| FAA AAS | B$ 1 | B$ 4 | B$ 7 | 6 - 8 years |