Resilience: an Essential Property for the Sustainability of Computing Systems and Infrastructures

— From Dependability to Resilience —

Jean-Claude Laprie

ReSIST Summer School

Resilience in Computing Systems and Information Infrastructures
— from Concepts to Practice —

24th-28th September 2007, Porquerolles

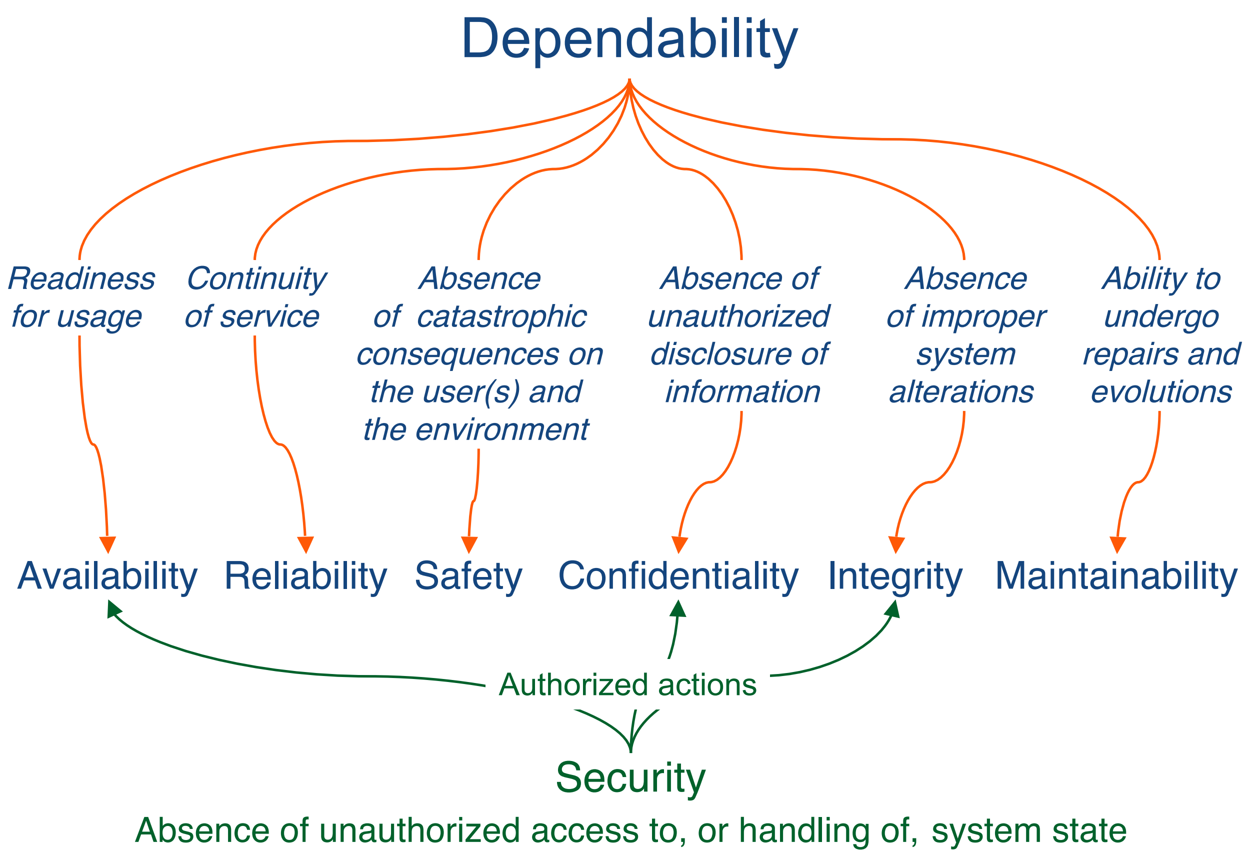
- Dependability
  - Basic concepts
    [From A. Avizienis, JC. Laprie, B. Randell, C. Landwehr, 'Basic Concepts and Taxonomy of Dependable and Secure Computing', IEEE Tr. Dependable and Secure Computing, 2004]
  - State-of-the-art from statistics
- Resilience
  - Definition and technologies
    - ☞continuously evolving (complex) systems

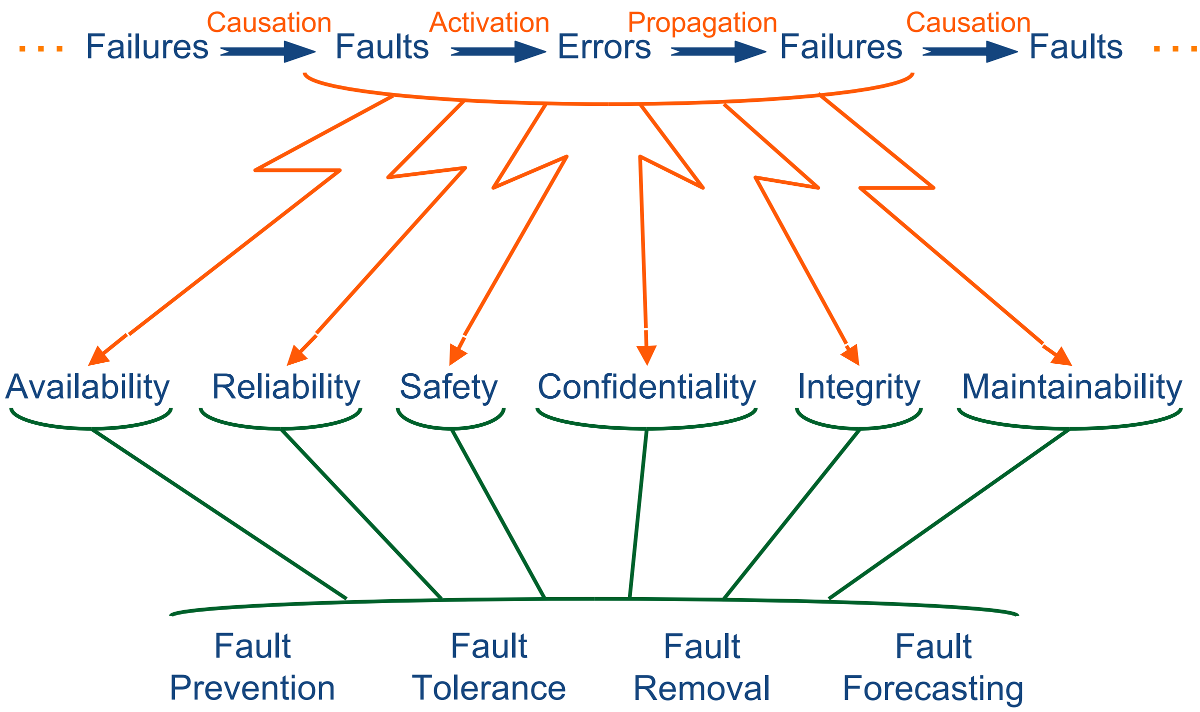
Dependability: ability to deliver service that can justifiably be trusted

- Service delivered by a system: its behavior as it is perceived by its user(s)
- User: another system that interacts with the former
- Function of a system: what the system is intended to do
- (Functional) Specification: description of the system function
- Correct service: when the delivered service implements the system function
- (Service) Failure: event that occurs when the delivered service deviates from correct service, either because the system does not comply with the specification, or because the specification did not adequately describe its function
- Failure modes: the ways in which a system can fail, ranked according to failure severities
- Part of system state that may cause a subsequent service failure: error
- Adjudged or hypothesized cause of an error: fault

Dependability: ability to avoid failures that are unacceptably frequent or severe

- Failures are more frequent or more severe than acceptable: dependability failure

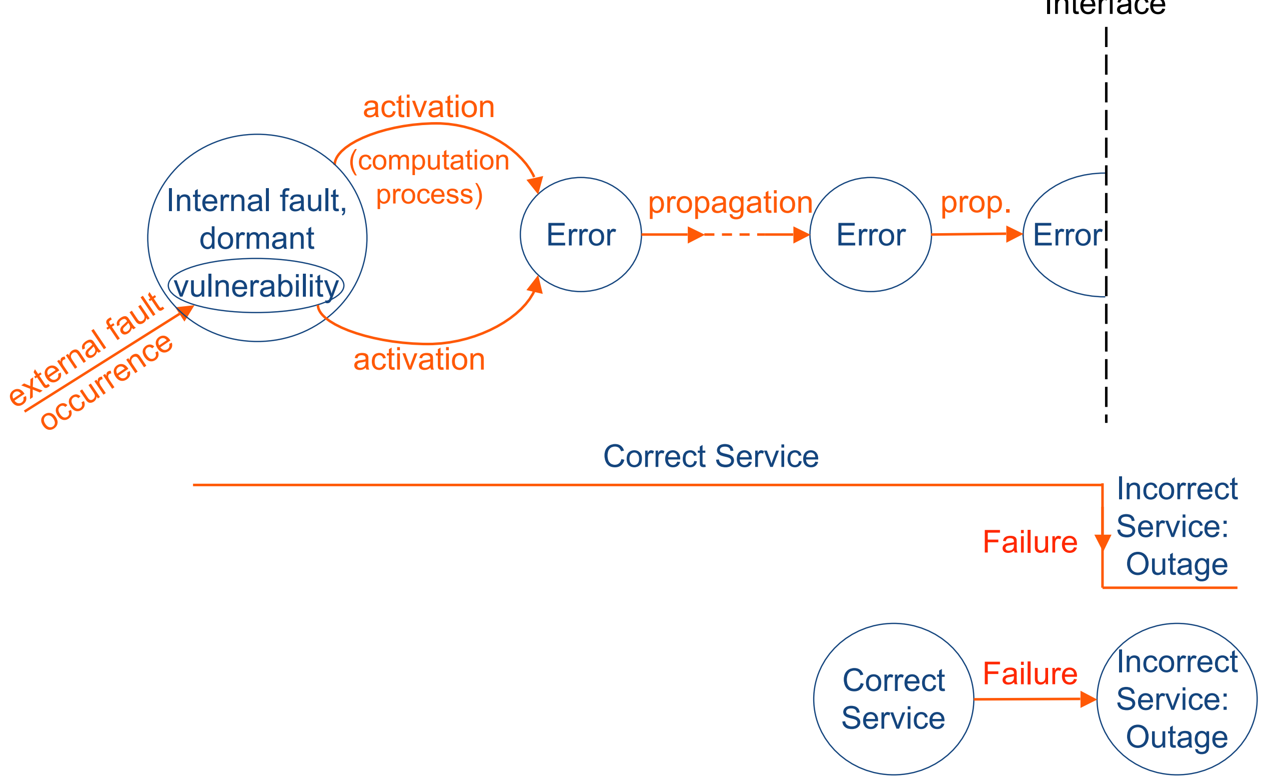


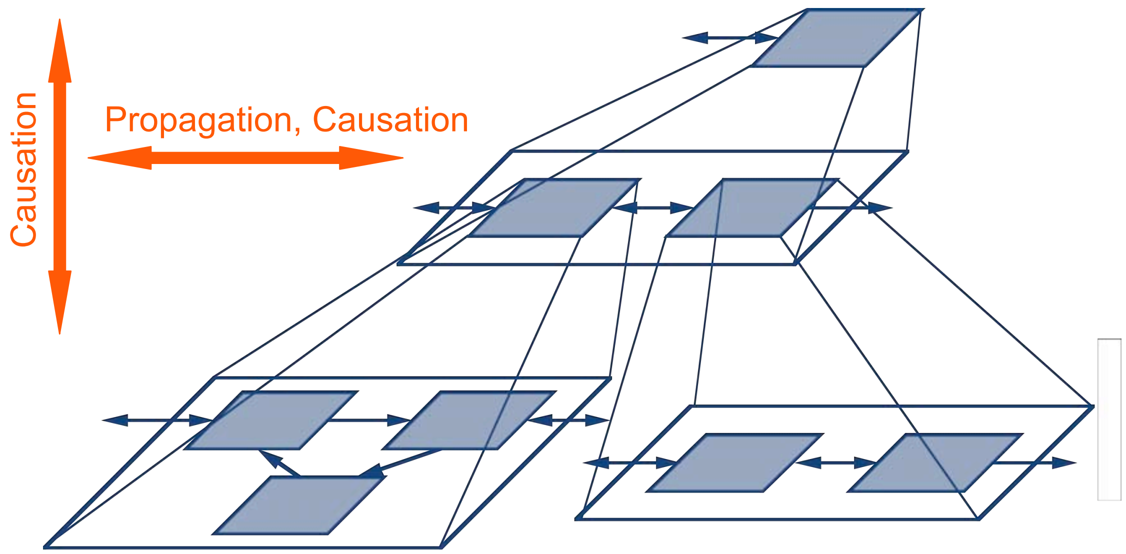


Dependability attributes

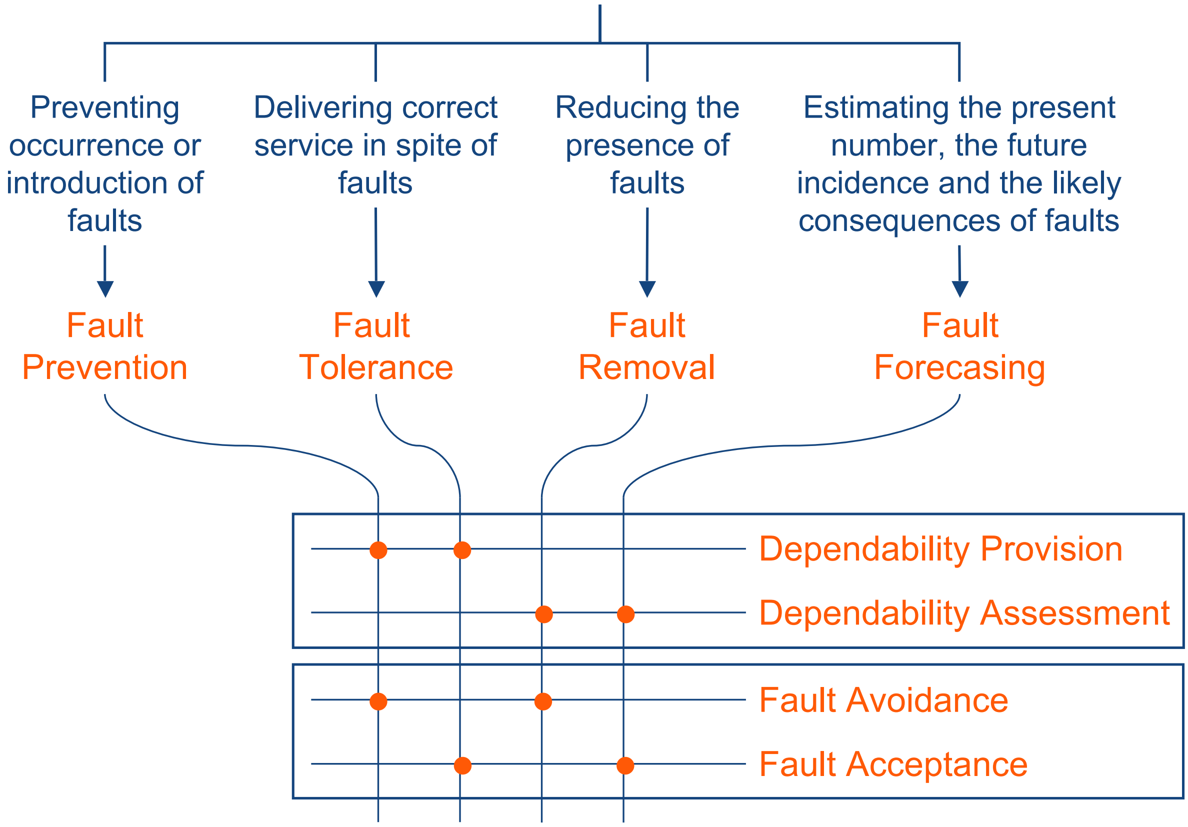
- Availability, Reliability, Safety, Confidentiality, Integrity, Maintainability: Primary attributes
- Secondary attributes
  - Specialization
    - Robustness: dependability with respect to external faults
    - Survivability: dependabilty in the presence of active fault(s)
  - Distinguishing among various types of (meta-)information
    - Accountability: availability and integrity of the person who performed an operation
    - Authenticity: integrity of a message content and origin, and possibly some other information, such as the time of emission
    - Non-repudiability: availability and integrity of the identity of the sender of a message (non-repudiation of the origin), or of the receiver (non-repudiation of reception)

Dependability Threats





Means for Dependability



Dependability definitions

- Original definition: ability to deliver service that can justifiably be trusted
  - ☞ Enables to generalize availability, reliability, safety, confidentiality, integrity, maintainability, that are then attributes of dependability
- Alternate definition: ability to avoid service failures that are unacceptably frequent or severe
  - ☞ A system can, and usually does, fail. Is it however still dependable? When does it become undependable?

    ↓

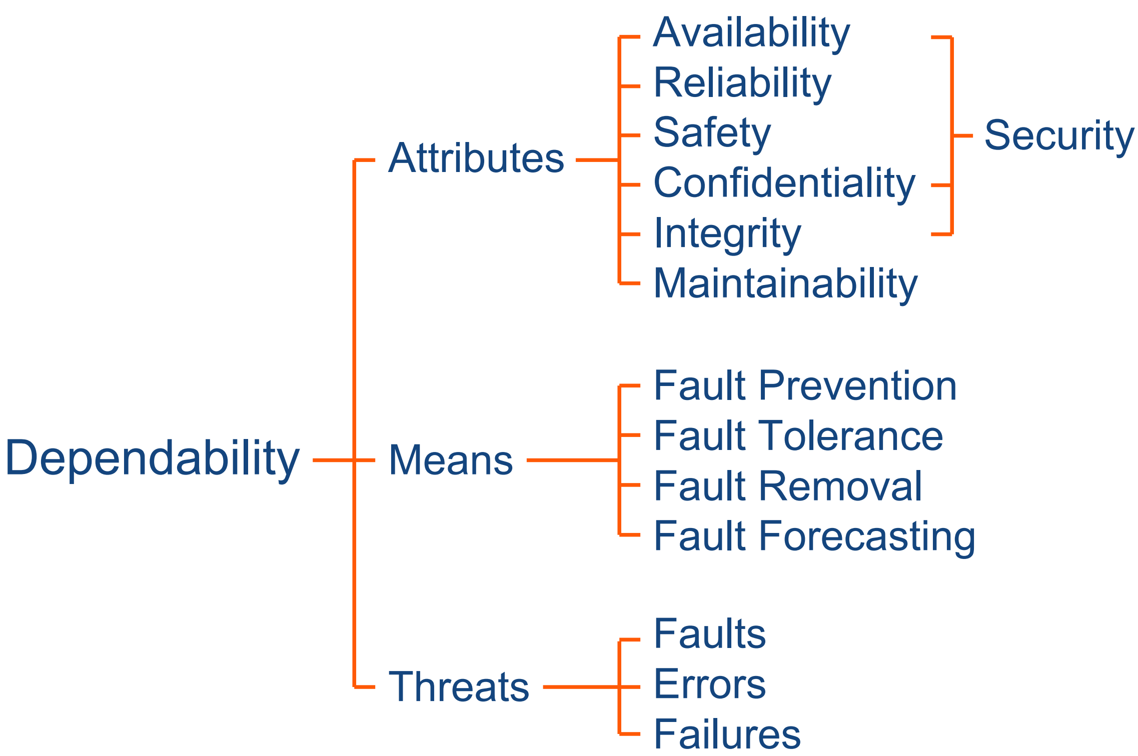
    criterion for deciding whether or not, in spite of service failures, a system is still to be regarded as dependable

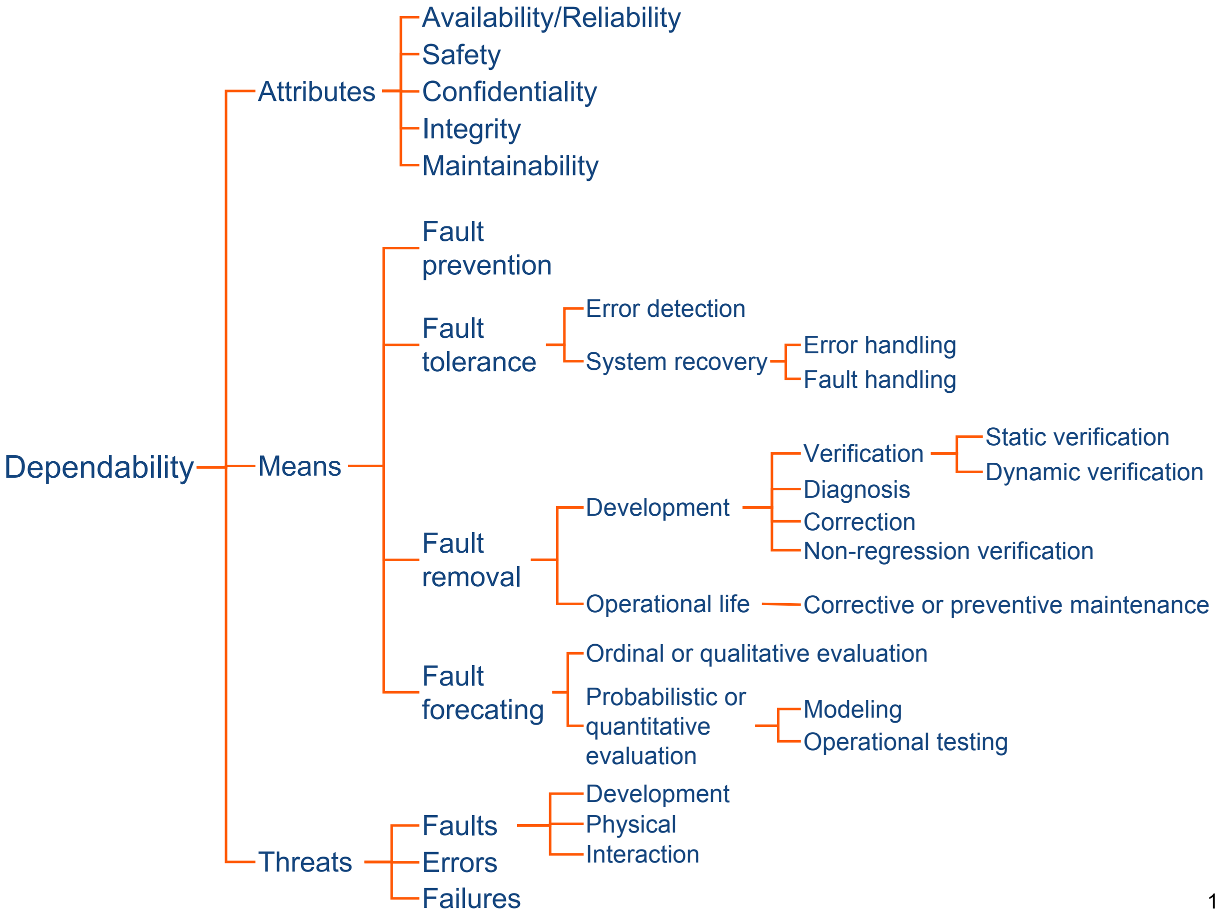
- **Dependence** of system A on system B is the extent to which system A's dependability is (or would be) affected by that of system B
- **Trust**: accepted dependence

  Explicitly   Implicitly

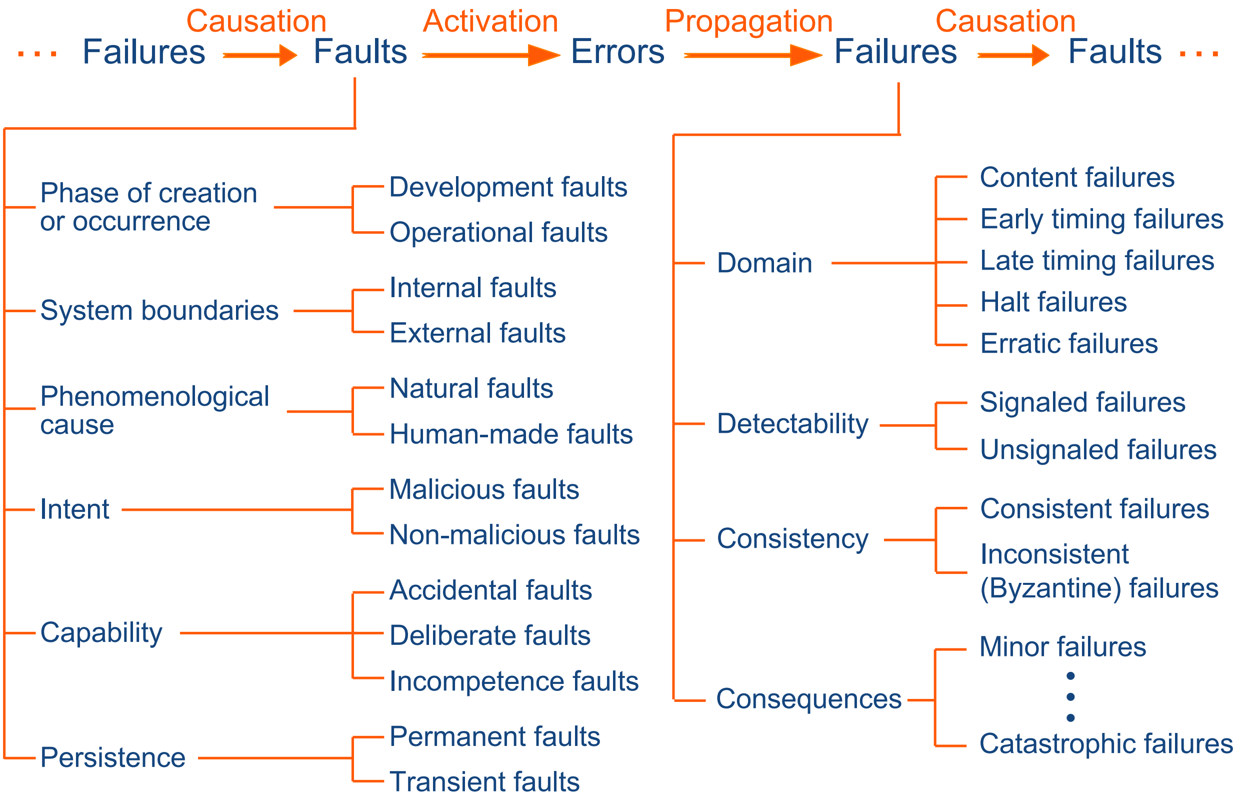
Dependability and similar notions

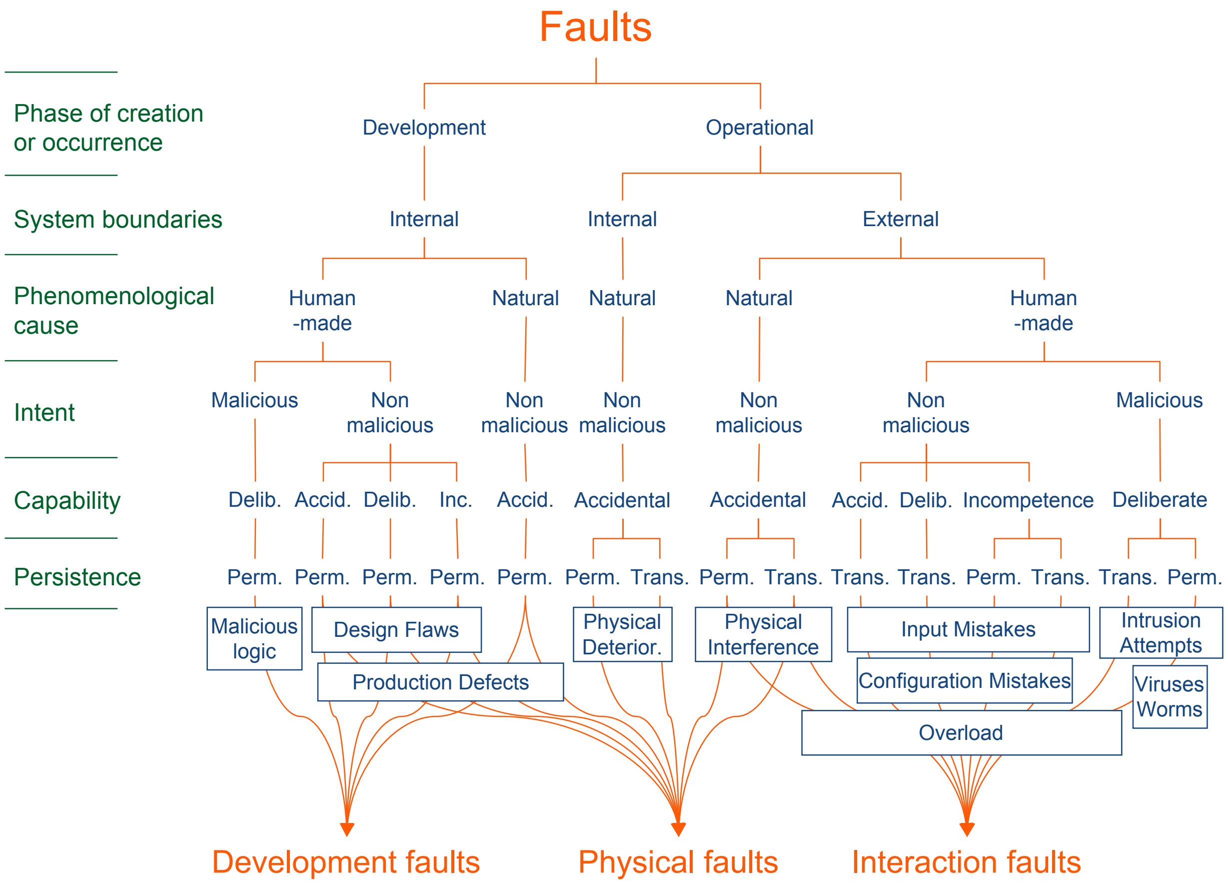
Concept	Dependability	High Confidence	Survivability	Trustworthiness
Goal	1) ability to deliver service that can justifiably be trusted 2) ability of a system to avoid service failures that are unacceptably frequent or severe	consequences of the system behavior are well understood and predictable	capability of a system to fulfill its mission in a timely manner	assurance that a system will perform as expected
Threats present	1) development faults (e.g., software flaws, hardware errata, malicious logic) 2) physical faults (e.g., production defects, physical deterioration) 3) interaction faults (e.g., physical interference, input mistakes, attacks, including viruses, worms, intrusions)	• internal and external threats • naturally occurring hazards and malicious attacks from a sophisticated and well-funded adversary	1) attacks (e.g., intrusions, probes, denials of service) 2) failures (internally generated events due to, e.g., software design errors, hardware degradation, human errors, corrupted data) 3) accidents (externally generated events such as natural disasters)	1) hostile attacks (from hackers or insiders) 2) environmental disruptions (accidental disruptions, either man-made or natural) 3) human and operator errors (e.g., software flaws, mistakes by human operators)
Reference		'Information technology frontiers for a new millenium', Blue Book 2000, NTSC	A. Ellison et al., 'Survivable network systems', SEI Report, 1999	F. Schneider, ed., 'Trust in cyberspace', National Academy Press, 1999

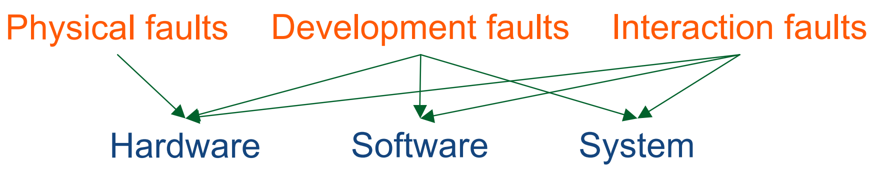


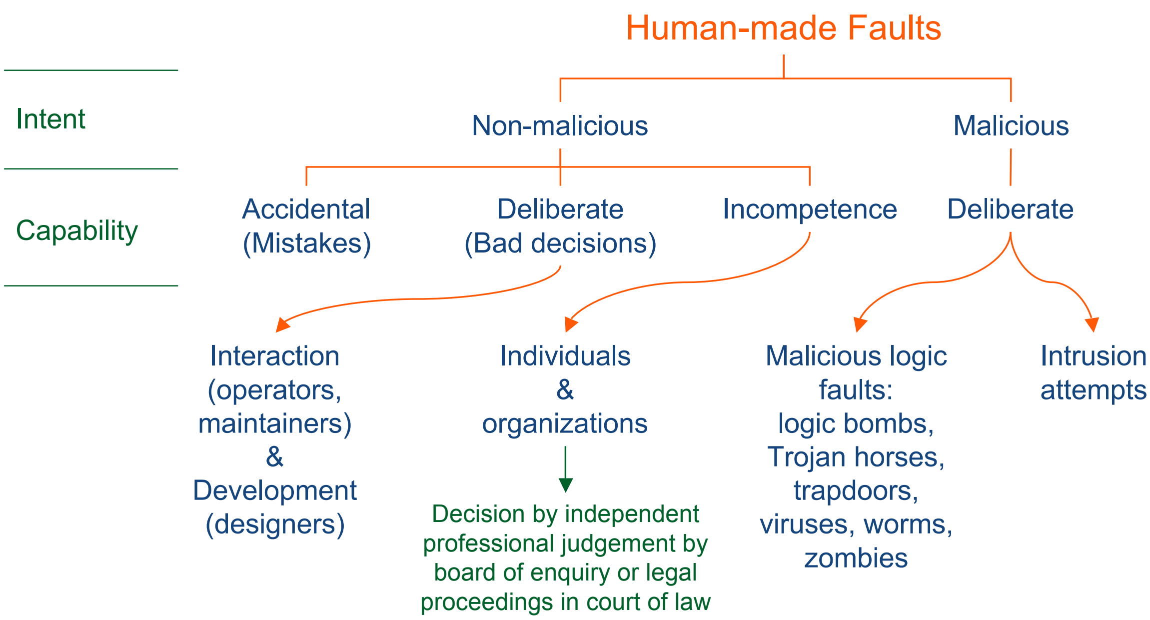

Dependability
Attributes
Availability/Reliability
Safety
Confidentiality
Integrity
Maintainability
Means
Fault prevention
Fault tolerance
Error detection
System recovery
Error handling
Fault handling
Fault removal
Development
Verification
Static verification
Dynamic verification
Diagnosis
Correction
Non-regression verification
Operational life
Corrective or preventive maintenance
Fault forecating
Ordinal or qualitative evaluation
Probabilistic or quantitative evaluation
Modeling
Operational testing
Threats
Faults
Development
Physical
Interaction
Errors
Failures

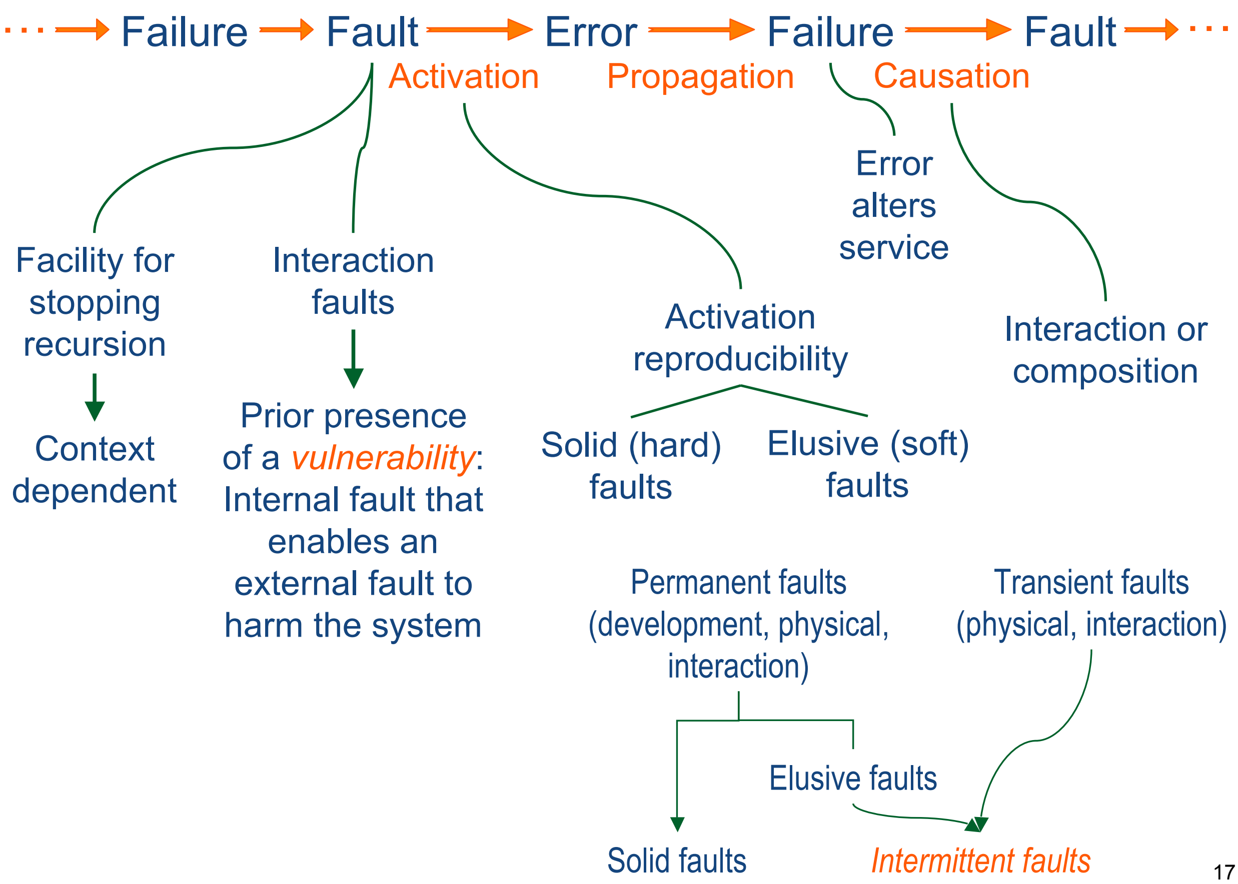
Threats


... Failures
Causation
Faults
Activation
Errors
Propagation
Failures
Causation
Faults ...
Phase of creation or occurrence
Development faults
Operational faults
System boundaries
Internal faults
External faults
Phenomenological cause
Natural faults
Human-made faults
Intent
Malicious faults
Non-malicious faults
Capability
Accidental faults
Deliberate faults
Incompetence faults
Persistence
Permanent faults
Transient faults
Domain
Content failures
Early timing failures
Late timing failures
Halt failures
Erratic failures
Detectability
Signaled failures
Unsignaled failures
Consistency
Consistent failures
Inconsistent (Byzantine) failures
Consequences
Minor failures
Catastrophic failures


Faults
Phase of creation or occurrence
System boundaries
Phenomenological cause
Intent
Capability
Persistence
Development
Operational
Internal
Internal
External
Human -made
Natural
Natural
Natural
Human -made
Malicious
Non malicious
Non malicious
Non malicious
Non malicious
Non malicious
Malicious
Delib.
Accid.
Delib.
Inc.
Accid.
Accidental
Accidental
Accid.
Delib.
Incompetence
Deliberate
Perm.
Perm.
Perm.
Perm.
Perm.
Perm.
Trans.
Perm.
Trans.
Trans.
Trans.
Perm.
Trans.
Trans.
Perm.
Malicious logic
Design Flaws
Production Defects
Physical Deterior.
Physical Interference
Input Mistakes
Configuration Mistakes
Overload
Intrusion Attempts
Viruses Worms
Development faults
Physical faults
Interaction faults


Physical faults
Development faults
Interaction faults
Hardware
Software
System


Human-made Faults
Intent
Capability
Non-malicious
Malicious
Accidental (Mistakes)
Deliberate (Bad decisions)
Incompetence
Deliberate
Interaction (operators, maintainers) & Development (designers)
Individuals & organizations
Decision by independent professional judgement by board of enquiry or legal proceedings in court of law
Malicious logic faults: logic bombs, Trojan horses, trapdoors, viruses, worms, zombies
Intrusion attempts



	Faults: Physical	Faults: Development	Faults: Interaction	Failures: Localized	Failures: Distributed	Availability/ Reliability	Safety	Confidentiality
June 1980: False alerts at the North American Air Defense (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" penetrates several tens of sensitive computing facilities		✔	✔	✔				✔
November 1988: Internet worm		✔	✔		✔	✔		
15 January 1990: 9 hours outage of the long-distance phone in the USA		✔			✔	✔		
February 1991: Scud missed by a Patriot (Dhahran, Gulf War)		✔	✔	✔		✔	✔	
November 1992: Crash of the communication system of the London ambulance service		✔	✔		✔	✔	✔	
26 and 27 June 1993: Authorization denial of credit card operations in France	✔	✔			✔	✔		
4 June 1996: Failure of Ariane 5 maiden flight		✔		✔		✔		
13 April 1998: Crash of the AT&T data network		✔	✔		✔	✔		
February 2000: Distributed denials of service on large Web sites		✔	✔		✔	✔		
May 2000: Virus *I love you*		✔	✔		✔	✔		
July 2001: Worm *Code Red*		✔	✔		✔	✔		
August 2003: Propagation of the electricity blackout in the USA and Canada		✔	✔		✔	✔		
October 2006: 83,000 e-mail addresses, credit card info, banking transaction files stolen in UK		✔	✔		✔			✔

Average outage costs

Industry sector	Energy	2,8	Millions of $ revenue/hour lost
	Manufacturing	1,6	
	Financial institutions	1,4	
	Insurance	1,2	
	Retail	1,1	
	Banking	1	

Yearly cost of failures

Estimates of insurance companies (2000)	France (private sector)	USA	UK
Accidental faults	0,9 G€	4 G$	1,25 G£
Malicious faults	1 G€		
Global estimate	USA : 80 G$	UE : 60 G€	

Maintenance costs

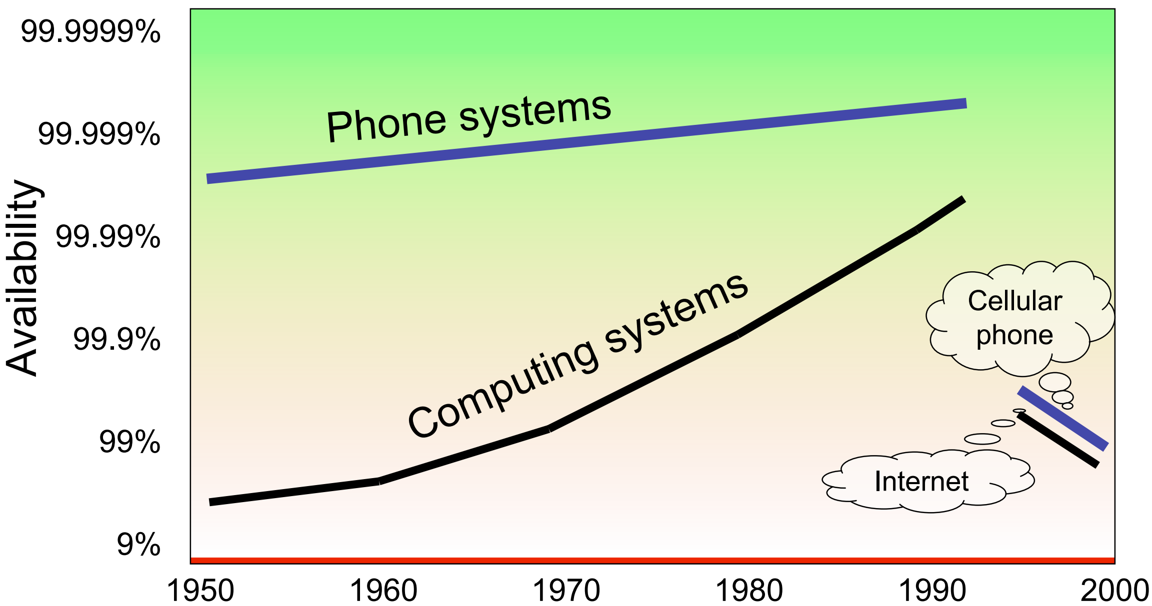
- ☞ Space shuttle on-board software: 100 M $ / an

Cost of software project cancellation (failure of the development process)

	Success	Challenged	Cancelled
☞ USA [Standish Group, 2002, 13522 projects]	34%	51%	15%

loss ~ 38 G$ (out of total 225 G$)

	Estimate 1983	Estimate 1988 (contract passed)	Estimate 1994	Timing shift (estimate 1994)
☞ FAA AAS	1 G $	4 G $	7 G $	6 - 8 yrs



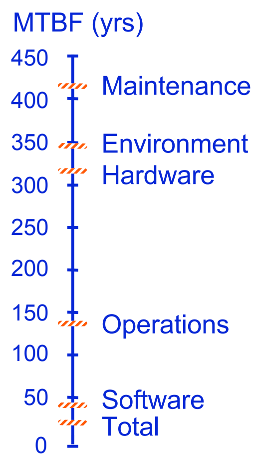
From J. Gray, 'Dependability in the Internet era'

- ☞ Complexity
- ☞ Economic pressure

Availability	Outage duration/yr
0,999999	32s
0,99999	5mn 15s
0,9999	52mn 34s
0,999	8h 46mn
0,99	3d 16h
0,9	36d 12h

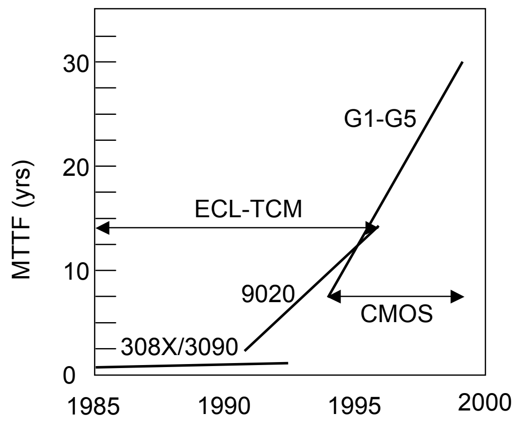
Tandem

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



[From J. Gray, 'A Census of Tandem System Availability Between 1985 and 1990', IEEE Tr. On reliability, Oct.1990]

High end IBM servers

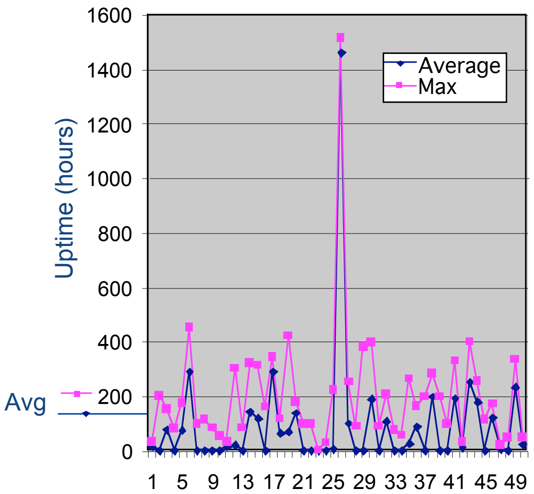


Mean time to system crash, due to hardware failure

[From L. Spainhower and T. A. Gregg, 'IBM S/390 Parallel Enterprise Server G5 fault tolerance: A historical perspective', IBM J. Research and Development, 1999]

Website uptime statistics (Netcraft)

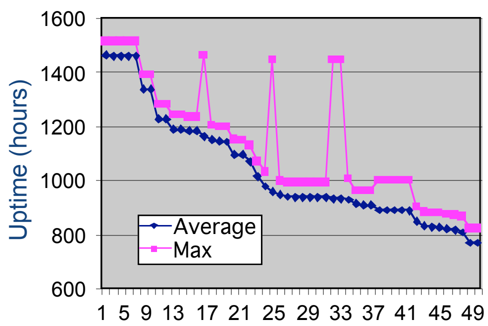
Top 50 most requested sites (July 2006)



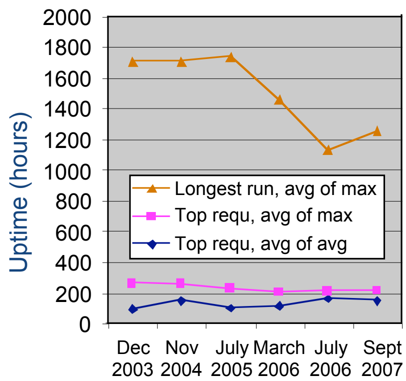
Availability for 100h MTBF

MTTR	Availability
1 min	99.98
10 mins	99.83
1 hour	99.01
8 hours	98.59

Top 50 longest running sites (July 2006)



Evolution over time

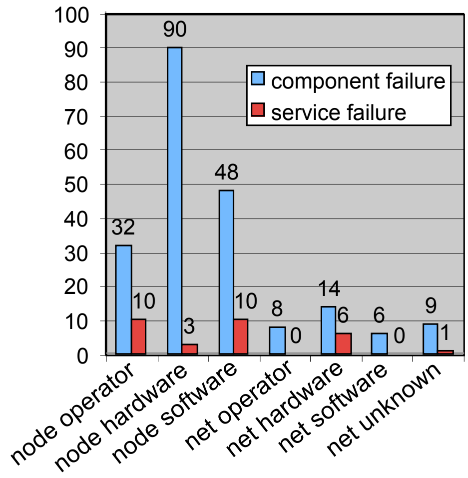


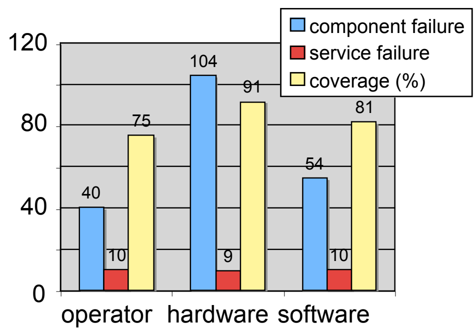
Three large websites [from D. Oppenheimer, A. Ganapathi, D.A. Patterson, ‘Why do Internet services fail, and what can be done about it?’, USISTS ‘03]

Website		*Online* (mature)	*Readmostly* (mature)	*Content* *(bleeding edge)*
Service characteristic	Hits per day	~100 million	~100 million	~7 million
	# of machines	~500, 2 sites	>2000, 4 sites	~500, ~15 sites
	Front-end node architecture	Solaris on SPARC and x86	Open-source OS on x86	Open-source OS on x86
	Back-end node architecture	Network Appliance filters	Open-source OS on x86	Open-source OS on x86
	Period of data stud.	7 months	6 months	3 months
	Component failures	296	N/A	205
	Service failures	40	21	56
	MTTF	126 hours	206 hours	39 hours
Service failure cause by location	Front-end	77%	0%	66%
	Back-end	3%	10%	11%
	Network	18%	81%	18%
	Unknown	2%	9%	4%
Average TTR by part of service (hrs)	Front-end	9.4 (16 serv. fai.)	N/A	2.5 (10 serv. fai.)
	Back-end	7.3 (5 serv. fai.)	0.2 (1 serv. fai.)	14 (3 serv. fai.)
	Network	7.8 (4 serv. fai.)	1.2 (16 serv. fai.)	1.2 (2 serv. fai.)
Average availability		93.5%	97.2%	97.8%

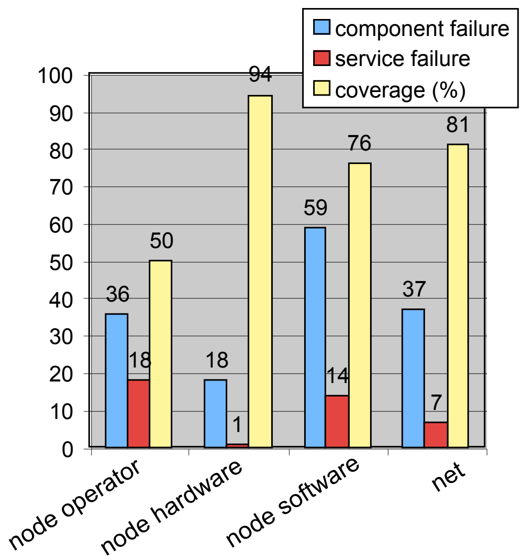
Component failure to service failure

Online

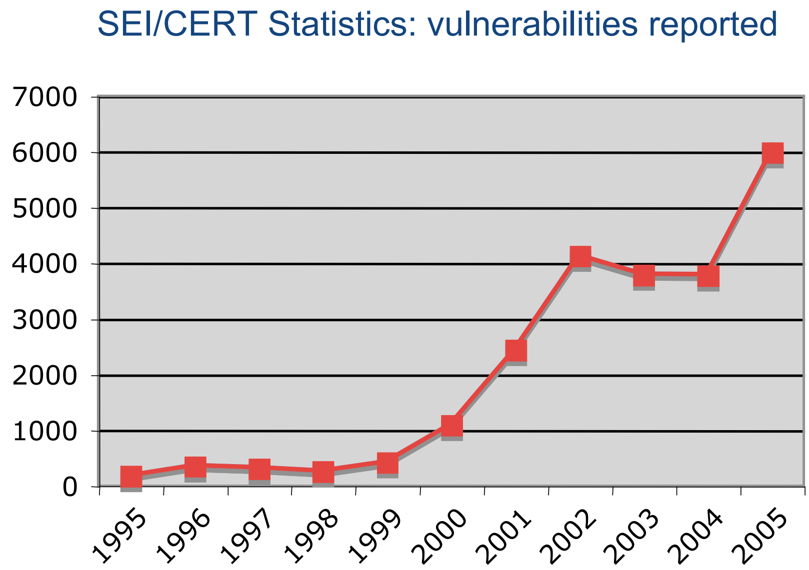




Content



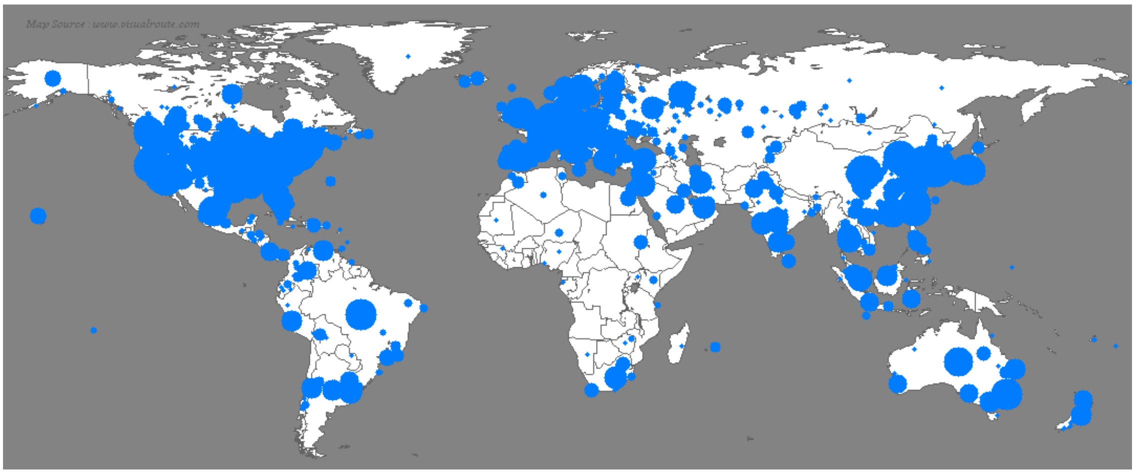
Malicious faults



Slammer/Sapphire worm

[From: http://www.caida.org/publications/papers/2003/sapphire/sapphire.html]

The fastest computer worm in history. As it began spreading throughout the Internet, it doubled in size every 8.5 seconds. It infected more than 90 percent of vulnerable hosts within 10 minutes.The worm began to infect hosts slightly before 05:30 UTC on Saturday, January 25, 2003. Sapphire exploited a buffer overflow vulnerability in computers on the Internet running Microsoft's SQL Server or MSDE 2000 (Microsoft SQL Server Desktop Engine). This weakness in an underlying indexing service was discovered in July 2002; Microsoft released a patch for the vulnerability before it was announced. The worm infected at least 75,000 hosts, perhaps considerably more, and caused network outages and such unforeseen consequences as canceled airline flights, interference with elections, and ATM failures.

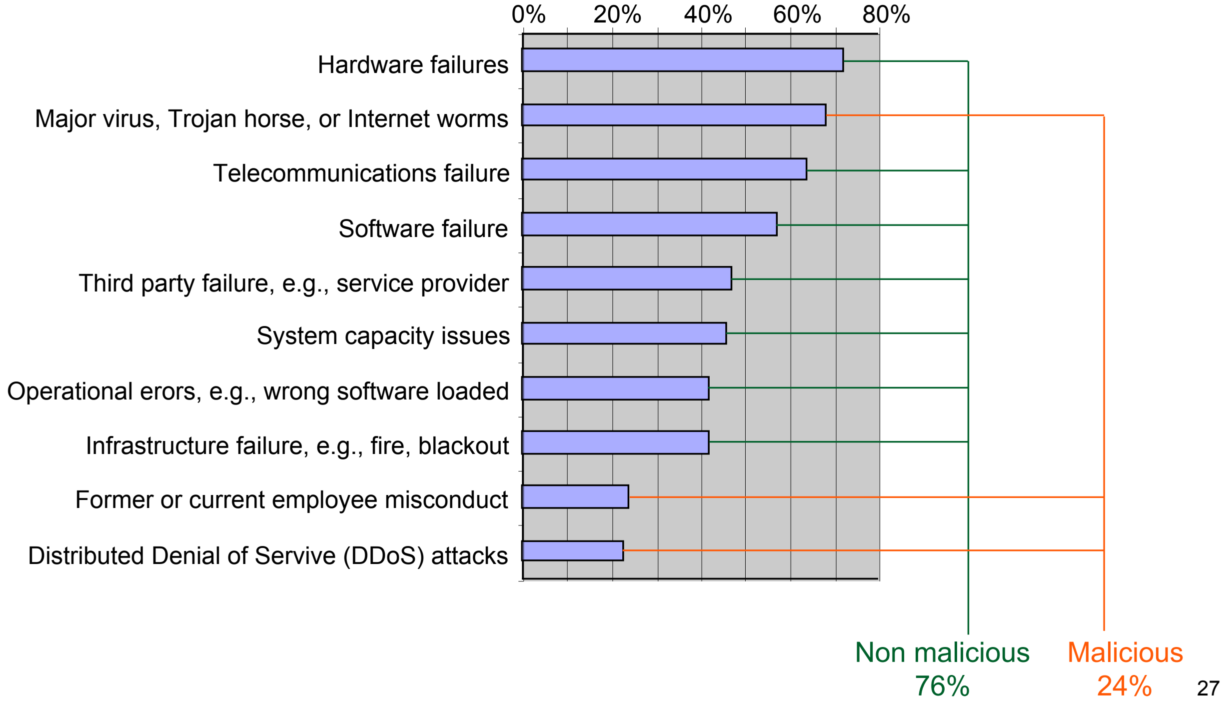


The geographic spread of Sapphire in the 30 minutes after release. The diameter of each circle is a function of the logarithm of the number of infected machines, so large circles visually underrepresent the number of infected cases in order to minimize overlap with adjacent locations.

Global Information Security Survey 2004 — Ernst & Young

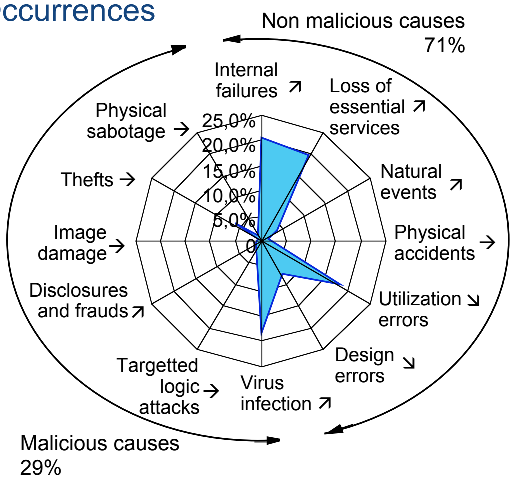
Loss of availability: Top ten incidents

Percentage of respondents that indicated the following incidents resulted in an unexpected or unscheduled outage of their critical business

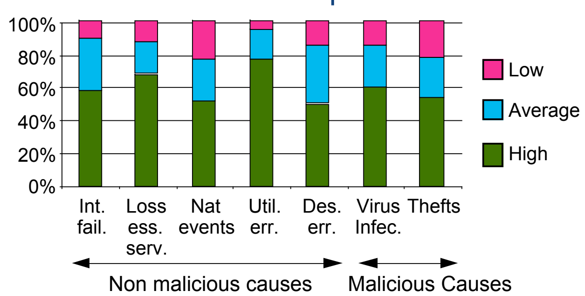


Yearly survey on computer damages in France — CLUSIF (2000, 2001, 2002)

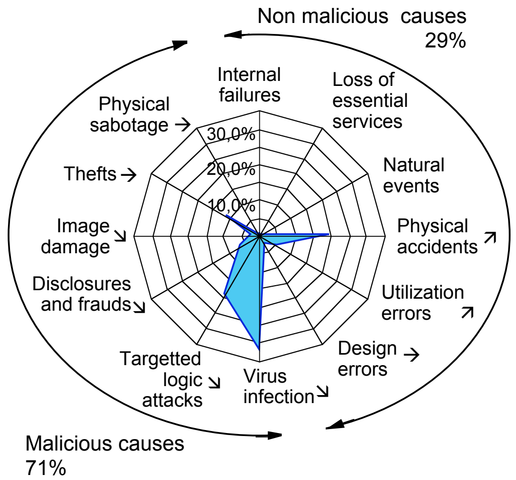
Occurrences



Occurrence impact



Risk perception



3 year trends
→ stable
↗ increase
↘ decrease

About Resilience

Social psychology	Material science	Child psychiatry and psychology	Ecology	Business	Industrial safety
↓	↓	↓	↓	↓	↓
Elasticity, spirit, resource, good mood	Robustness and elasticity	Living, developing successfully when facing adversity	Moving from a stability domain to another one	Capacity to reinvent a business model before circumstances force to	Anticipating risk changes before damage occurrence

Adaptation to evolutionary changes, and getting back after a setback

↓

Fault and evolution tolerance

Ecology, C.S. Holling, "Resilence and stability of ecological systems", *Annual Review of Ecology and Systematics*, 1973

- «resilience determines the persistence of relationships within a system and is a measure of the ability of these systems to absorb changes of state variables, driving variables, and parameters, and still persist»
- Relationship between resilience and stability in open systems
  - «a system can be very resilient and still fluctuate greatly, i.e., have low stability »
  - «low stability seems to introduce high resilience»
- diversity pointed out as of significant influence on both stability (decreasing it, as it may create several stability domains) and resilience (increasing it)

In computing systems

- Resilient
  - In use for 30+ years
  - Recently, escalating use ➔ buzzword
  - Used essentially as synonym to fault tolerant
  - Noteworthy exception: preface of *Resilient Computing Systems*, T. Anderson (Ed.), Collins, 1985

    "A *resilient* computing system is capable of providing dependable service to its users over a wide range of potentially adverse circumstances. The two key attributes here are dependability and robustness. […] A computing system can be said to be *robust* if it retains its ability to deliver service in conditions which are beyond its normal domain of operation, whether due to harsh treatment, or unreasonable service requests, or misoperation, or the impact of faults, or lack of maintenance »

☞ Fault-tolerant computing systems are known for exhibiting some robustness with respect to fault and error handling, in the above sense, i.e., for situations exceeding their specification, e.g.:

- Tolerance of elusive software faults thanks to loosely-coupled architectures in Tandem systems
- Tolerance errors that escaped detection and thus did not trigger recovery in Delta-4

☞ This of course should not lead to forget that, contrariwise, total coverage with respect to specified faults is hardly achievable

Moving to ubiquitous systems

Large, networked, evolving systems constituting complex information infrastructures — perhaps involving everything from super-computers and huge server farms to myriads of small mobile computers and tiny embedded devices

At stake: maintain dependability in spite of continuous evolutionary changes

functional   environmental   technological

Examples of changes:

- ✓ Dynamically changing systems, e.g., spontaneous, or ‘ad-hoc’, networks of mobile nodes and sensors
- ✓ Growth of systems as demand increases
- ✓ Interactions between systems of differing natures, e.g., large-scale information infrastructure on the one hand and networks of sensors on the other
- ✓ Merging of systems, e.g., in company acquisitions, or coupling of systems, e.g., in military coalitions
- ✓ Ever-evolving and growing problem of attacks both by amateur hackers and by professional criminals

Definition of resilience for computing systems and information infrastructures

The persistence of dependability when facing evolutionary changes

↓

The persistence of the ability to deliver service that can justifiably be trusted, when facing evolutionary changes

- Functional
- Environmental
- Technological

- Foreseen, e.g., new versioning
- Foreseeable, e.g., advent of new hardware platforms
- Unforeseen, e.g., drastic changes in service requests or new types of threats

- Short term, e.g., seconds to hours, as in dynamicity or mobility
- Medium term, e.g., hours to months, as in new versioning or reconfigurations
- Long term, e.g., months to years, as in reorganisations

☞ The definition does not exclude the possibility of failure

Alternate definition of dependability

↓

Ability to avoid service failures that are unacceptably frequent or severe

✻ Especially relevant in the context of evolutionary changes, as the changes can be directly a source of failure

↓

Incompatibilities between the formerly existing systems and the augmentations performed

Technologies for resilience

Evolutionary changes → **Evolvability**

☞ Adaptation

Trusted service → **Assessability**

☞ Verification and evaluation

Ubiquitous systems → **Usability**

☞ Human and system users

Complex systems → **Diversity**

☞ Taking advantage of existing diversity for avoiding single points of failure, and augmenting diversity

Means for dependability		Evolvability	Assessability	Usability	Diversity
	Fault Prevention	●		●	
	Fault Tolerance	●		●	●
	Fault Removal	●	●		
	Fault Forecasting	●	●		