Basic Concepts and Taxonomy of Dependable and Secure Computing

Presented by: Hadi Salimi
Instructor: Dr. Abdollahi Azgomi

Reliable Software Design Course
Iran University of Science and Technology
Spring 2008

Introduction

- All the authors of this paper are well-known people in the area of dependable computing.
- A minimal consensus on dependability and security concepts.
- Develop an standard for standardization organizations to accept this taxonomy.

Basic Concepts

- Definition of system function, behavior, structure, and service
- The threats to dependability and security
- Dependability, security, and their Attributes
- The means to attain dependability and security

System and Function

- System is an entity that interacts with its environment (other systems, hardware, software, humans)
- The system boundary is the common frontier between the system and the environment.
- The function of a system is what the system is intended to do and is described by the functional specification

Behavior

- The behavior of a system is what the system does to implement its function and is described by a sequence of states.

- The total state of a given system is the set of the following states: computation, communication, stored information, interconnection, and physical condition.

Structure

- The structure of a system is what enables it to generate the behavior.

- From this viewpoint, a system is composed of a set of components bound together in order to interact.

Service

- The service delivered by a system is its behavior as it is seen by its users.
- A user is another system that receives service from the provider.
- States can be:
  - Internal: cannot be accessed through interfaces.
  - External: can be accessed through interfaces.

The Threats to Dependability and Security

- Correct service is delivered when the service implements the system function.
- Service failure, is an event that occurs when the delivered service deviates from correct service
- A service failure is a transition from correct service to incorrect service to not implementing the system function
- Service outage: the period of delivery of incorrect service
- Service restoration: transition from incorrect service to correct service

Threats

- A service failure means that at least one more external state of the system deviate from the correct service state.
- Error is the part of the total state of the system that may lead to service failure.
- The cause of a error is called a fault.
- A fault is active when it cause an error, otherwise is dormant.

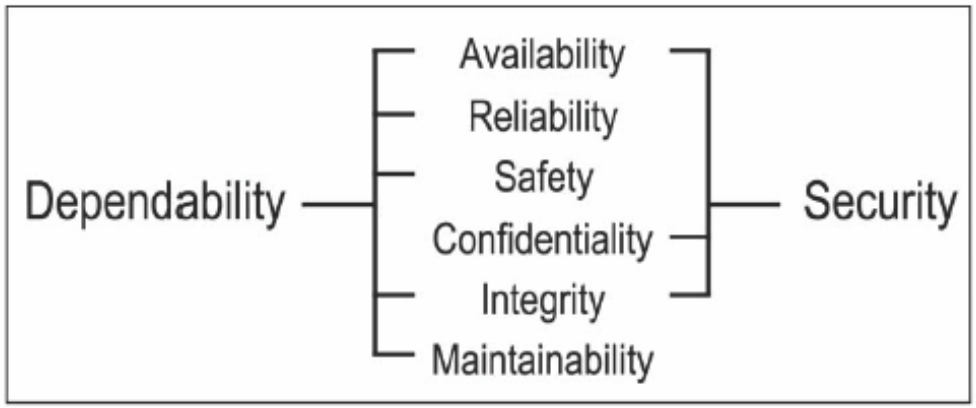
Dependability and Security Attributes

- **Dependability of a system is the ability to avoid service failures that are more frequent and more severe than is acceptable.**

- **Dependability attributes:**
  - availability: readiness for correct service.
  - reliability: continuity of correct service.
  - safety: absence of catastrophic consequences on the users and the environment.
  - integrity: absence of improper system alterations.
  - maintainability: ability to undergo modifications and repairs.

Dependability and Security Attributes (cont'd)

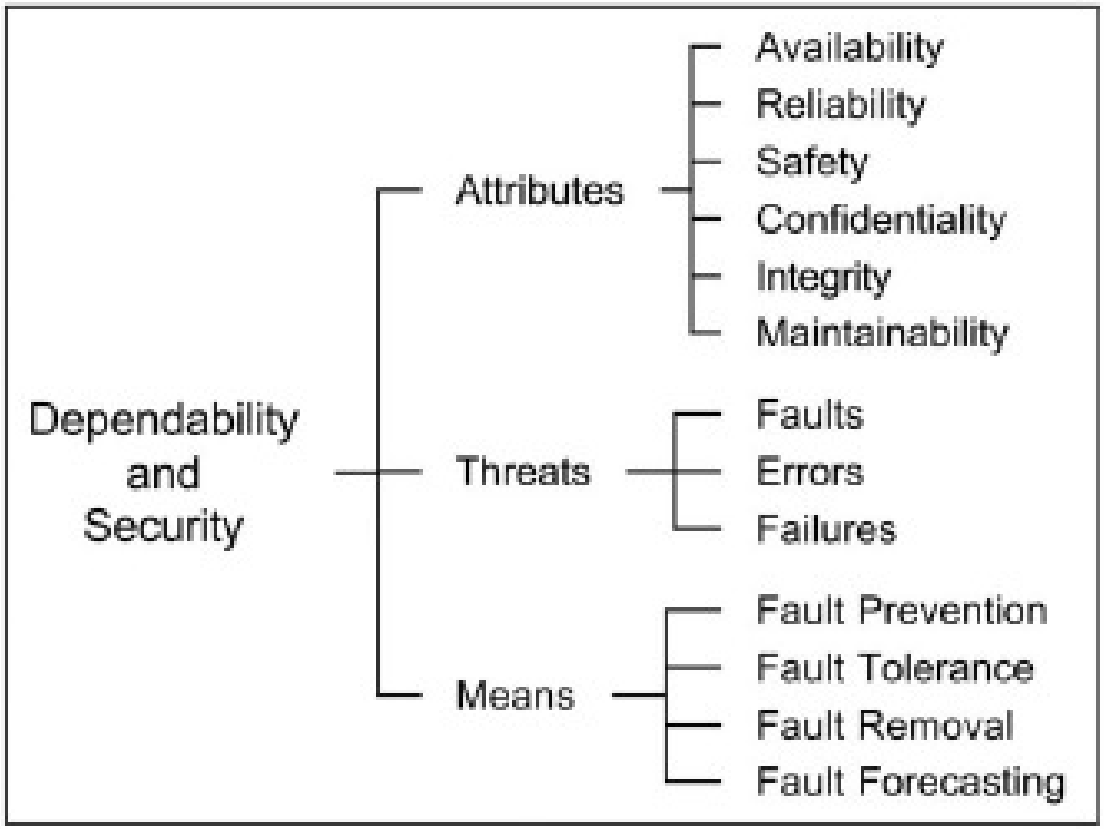
- Security attributes:
  - availability: for authorized action
  - confidentiality: absence of unauthorized disclosure of information
  - integrity: absence of unauthorized system alterations.



Means to Attain Dependability and Security

- Fault prevention
  - prevent the occurrence or introduction of faults.
- Fault tolerance
  - avoid service failures in the presence of faults.
- Fault removal
  - reduce the number and severity of faults.
- Fault forecasting
  - estimate the present number, the future incidence and the likely consequences of faults.

The Dependability and Security Tree



The dependability and security tree.

System Lifecycle

The lifecycle of a system includes two phases:

- Development

  All activities form requirement analysis to test phases

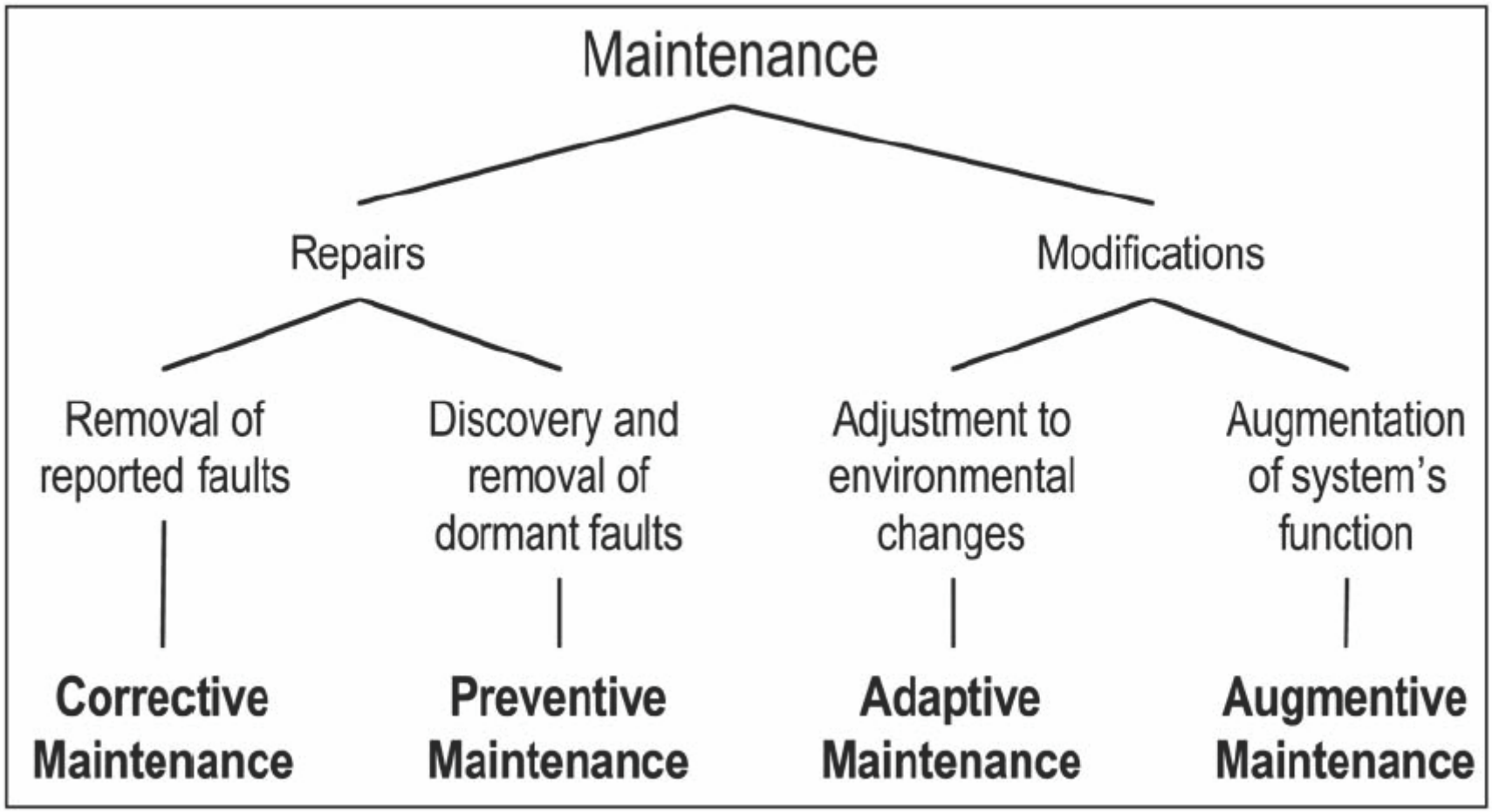
- Use

  begins when the system is accepted for use and starts delivering its services.

Maintenance

- Maintenance includes:
  - Repairs: Removing the fault by removing the cause of fault.
  - Modifications: Removing the fault by changing the system.

Maintenance



Maintenance vs. fault tolerance

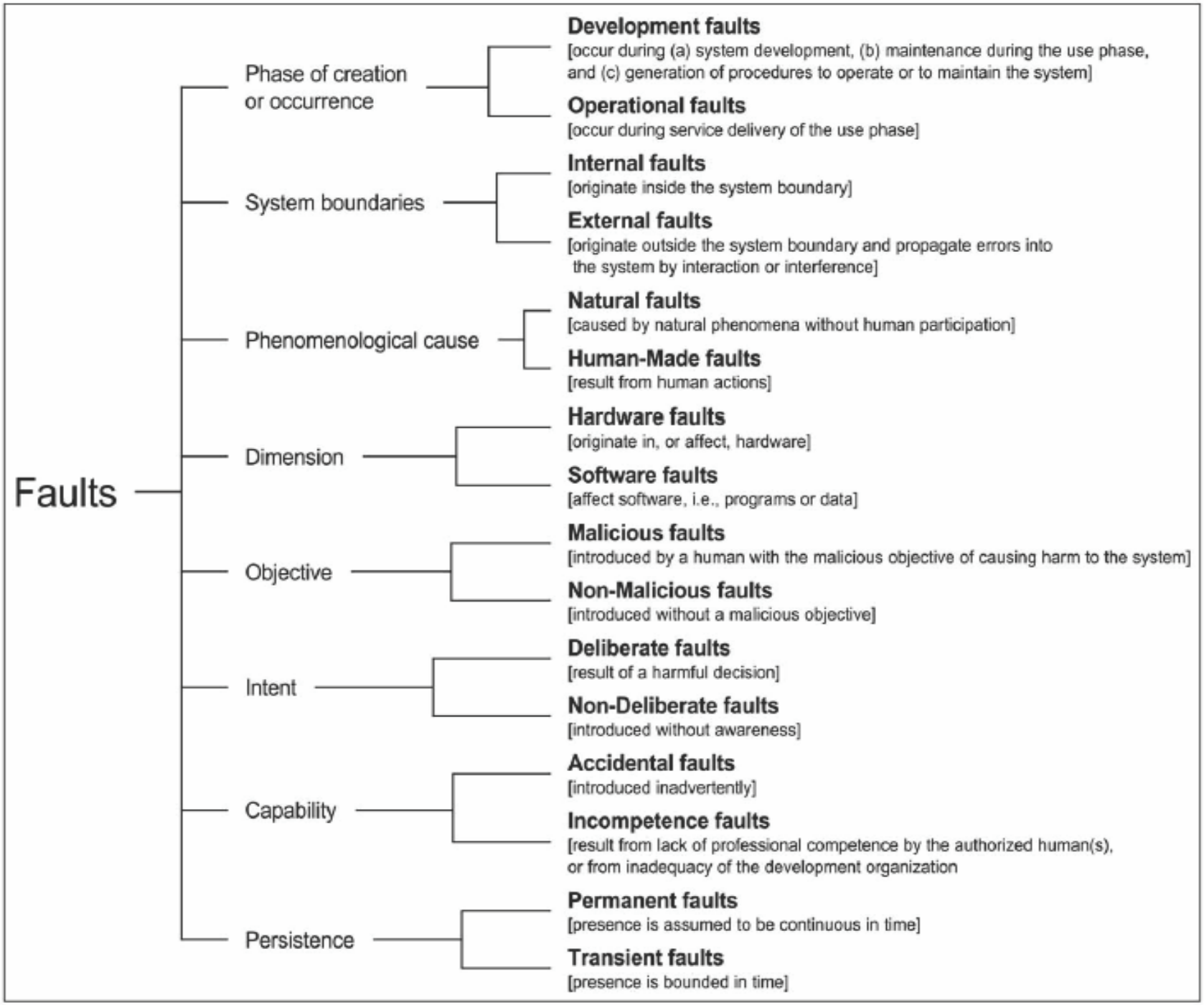
- Distinction between fault tolerance and maintenance:

maintenance involves the participation of an external agent, e.g., a repairman, test equipment, remote reloading of software

- Repair is part of fault removal (during the use phase)

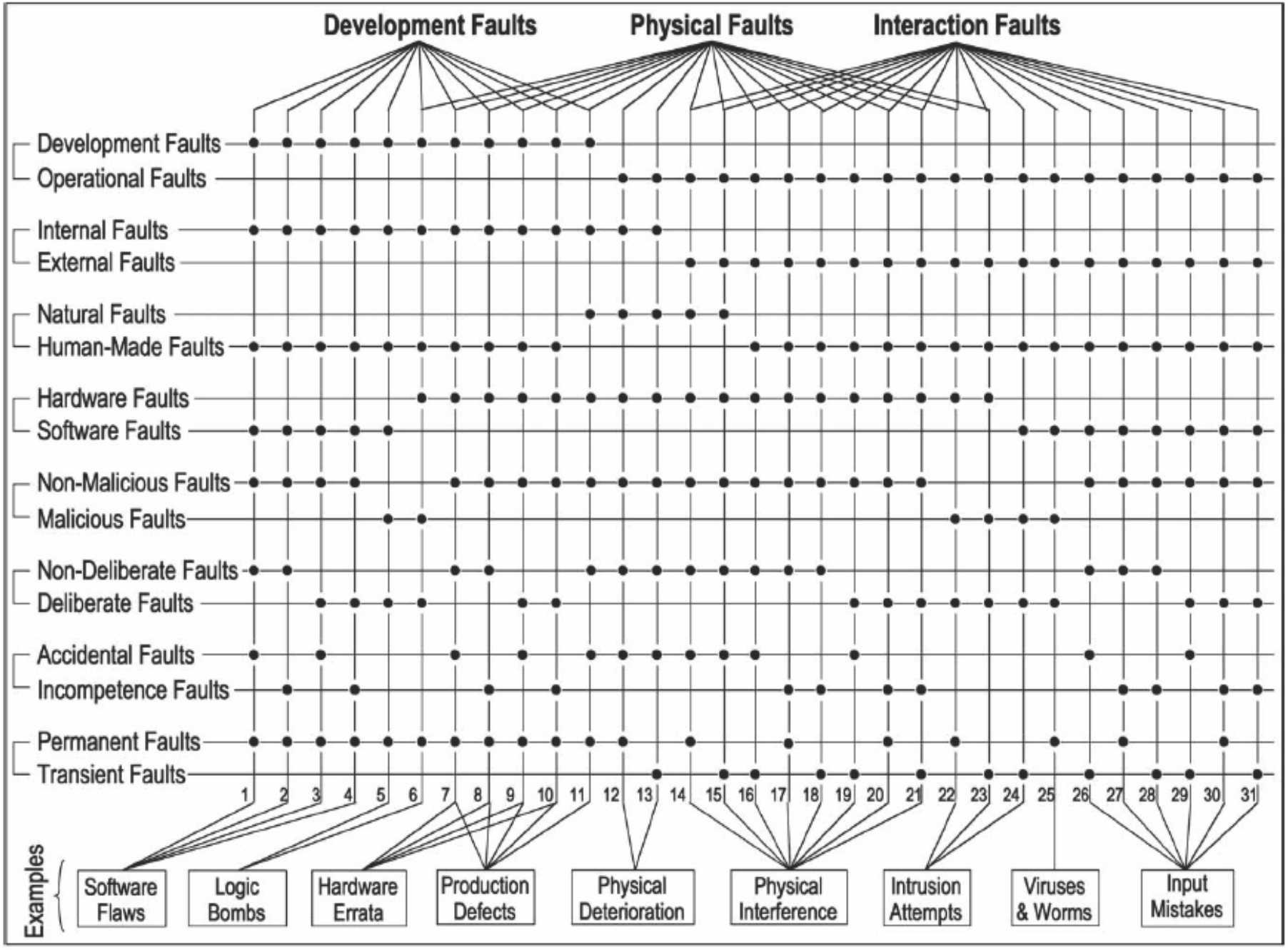
Taxonomy of Faults

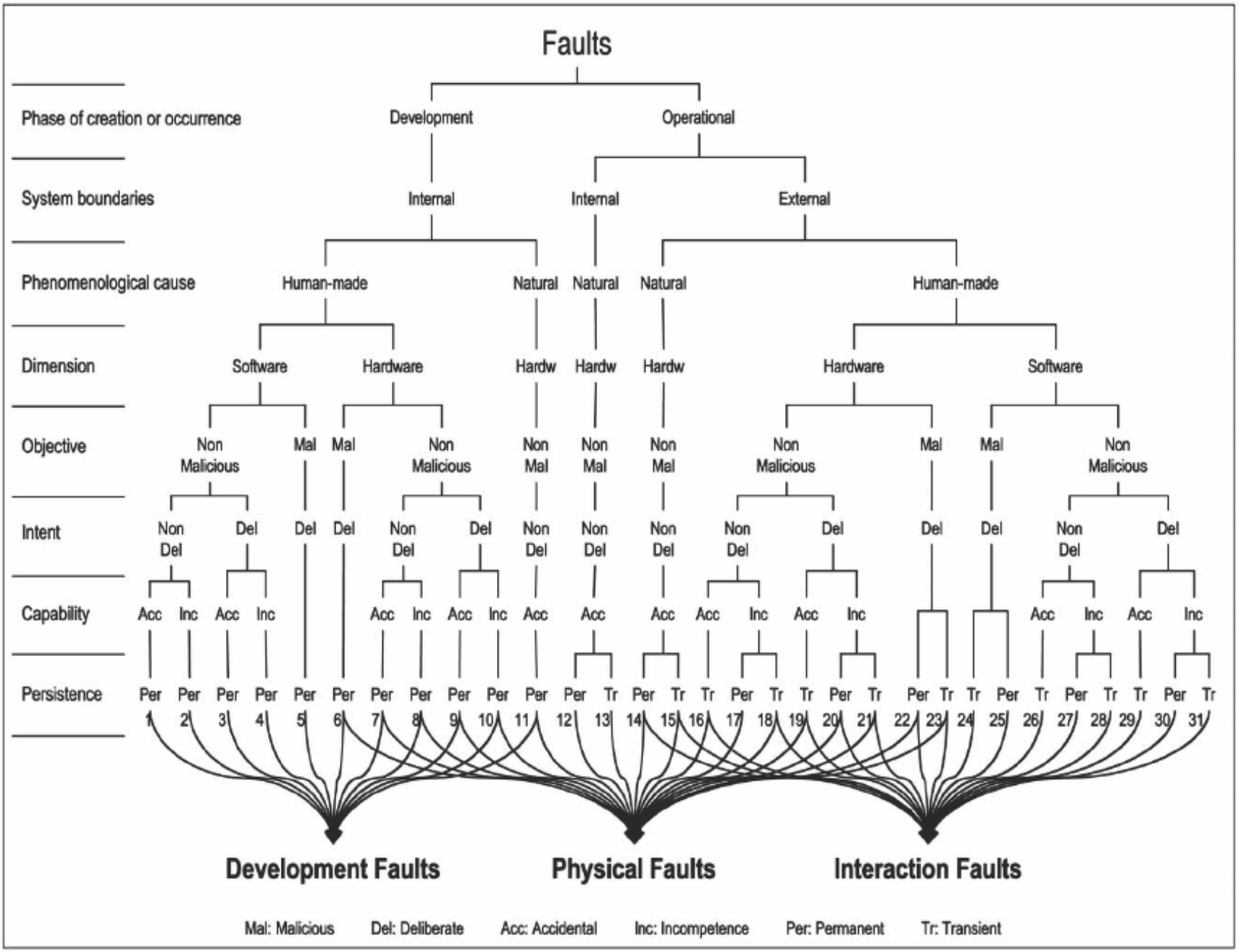
- All faults that may affect a system during its life are classified according to eight basic viewpoints
- If all combinations of the eight elementary fault classes were possible, there would be 256 different combined fault classes
- 31 faults have been identified


Faults
Phase of creation or occurrence
Development faults
[occur during (a) system development, (b) maintenance during the use phase, and (c) generation of procedures to operate or to maintain the system]
Operational faults
[occur during service delivery of the use phase]
System boundaries
Internal faults
[originate inside the system boundary]
External faults
[originate outside the system boundary and propagate errors into the system by interaction or interference]
Phenomenological cause
Natural faults
[caused by natural phenomena without human participation]
Human-Made faults
[result from human actions]
Dimension
Hardware faults
[originate in, or affect, hardware]
Software faults
[affect software, i.e., programs or data]
Objective
Malicious faults
[introduced by a human with the malicious objective of causing harm to the system]
Non-Malicious faults
[introduced without a malicious objective]
Intent
Deliberate faults
[result of a harmful decision]
Non-Deliberate faults
[introduced without awareness]
Capability
Accidental faults
[introduced inadvertently]
Incompetence faults
[result from lack of professional competence by the authorized human(s), or from inadequacy of the development organization
Persistence
Permanent faults
[presence is assumed to be continuous in time]
Transient faults
[presence is bounded in time]

Taxonomy of Faults (cont'd)

- All 31 combined faults are categorized to three major overlapping groups:
  - Development faults : occurring during development
  - Physical faults: affect hardware
  - Interaction faults: external faults


Development Faults
Physical Faults
Interaction Faults
Development Faults
Operational Faults
Internal Faults
External Faults
Natural Faults
Human-Made Faults
Hardware Faults
Software Faults
Non-Malicious Faults
Malicious Faults
Non-Deliberate Faults
Deliberate Faults
Accidental Faults
Incompetence Faults
Permanent Faults
Transient Faults
1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31
Examples
Software Flaws
Logic Bombs
Hardware Errata
Production Defects
Physical Deterioration
Physical Interference
Intrusion Attempts
Viruses & Worms
Input Mistakes


Faults
Phase of creation or occurrence
System boundaries
Phenomenological cause
Dimension
Objective
Intent
Capability
Persistence
Development
Operational
Internal
Internal
External
Human-made
Natural
Natural
Natural
Human-made
Software
Hardware
Hardw
Hardw
Hardw
Hardware
Software
Non Malicious
Mal
Mal
Non Malicious
Non Mal
Non Mal
Non Mal
Non Malicious
Mal
Mal
Non Malicious
Non Del
Del
Del
Del
Non Del
Del
Non Del
Non Del
Non Del
Non Del
Del
Del
Del
Non Del
Del
Acc
Inc
Acc
Inc
Acc
Inc
Acc
Inc
Acc
Acc
Acc
Acc
Inc
Acc
Inc
Acc
Inc
Acc
Inc
Per
Tr
Per
Tr
Tr
Per
Tr
Tr
Per
Tr
Per
Tr
Tr
Per
Tr
Tr
Per
Tr
1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20
21
22
23
24
25
26
27
28
29
30
31
Development Faults
Physical Faults
Interaction Faults
Mal: Malicious
Del: Deliberate
Acc: Accidental
Inc: Incompetence
Per: Permanent
Tr: Transient

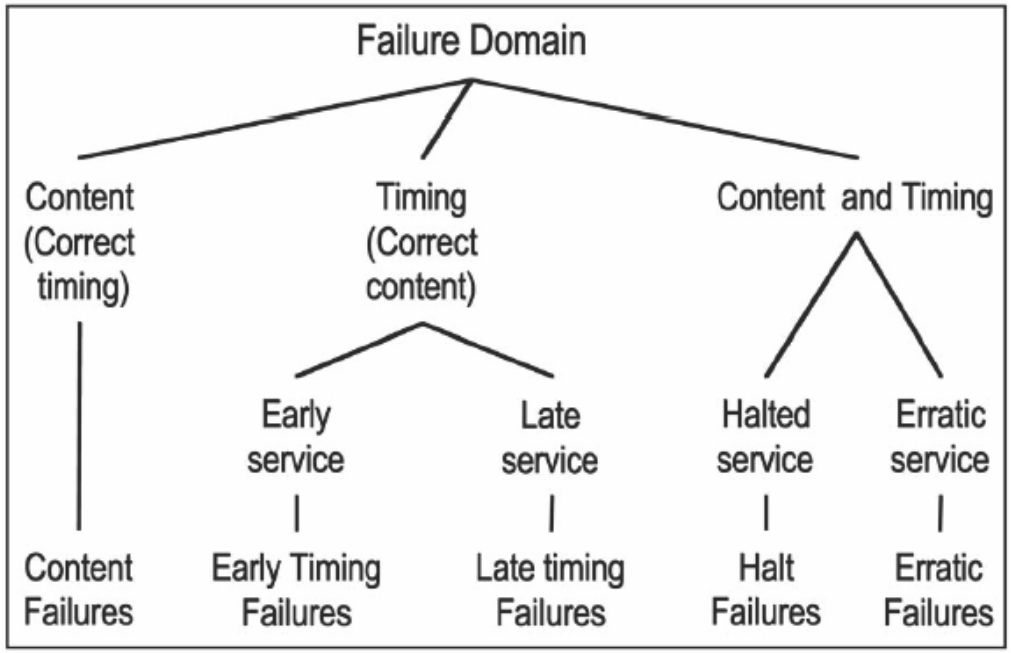
Failures

- Service failure: An event that occurs when the delivered service deviates from correct service.
- Development failure: Be introduced into the system being developed by its environment, especially by human developers, development tools and production facilities.
- Dependability and security failures: occurs when the given system suffers service failures more frequently or more severely than acceptable

Service Failures

- The service failures modes characterize according to four viewpoints:
  - Failure domain
  - Detectability of failures
  - Consistency of failures
  - Consequence of failures on the environment

Failure domain viewpoint



- content failures: service content deviates from implementing the system function
- timing failures: timing of service delivery deviates from implementing the system function
- halt failures: when the service is halted (silent failure)
- erratic failures: a service delivered but is erratic

Detectability viewpoint

- The detectability viewpoint addresses the signaling of service failures to the users

- Signaling at the service interface originates from detecting mechanisms in the system that check the correctness of the delivered service.
  - signaled failures: when the losses are detected and signaled by a warning signal
  - unsignaled failures: otherwise

- The detecting mechanisms themselves have two failure modes:
  - signaled failures :signaling a loss of function when no failure has actually occurred (false alarm)
  - unsignaled failures: not signaling a function loss

Consistency viewpoint

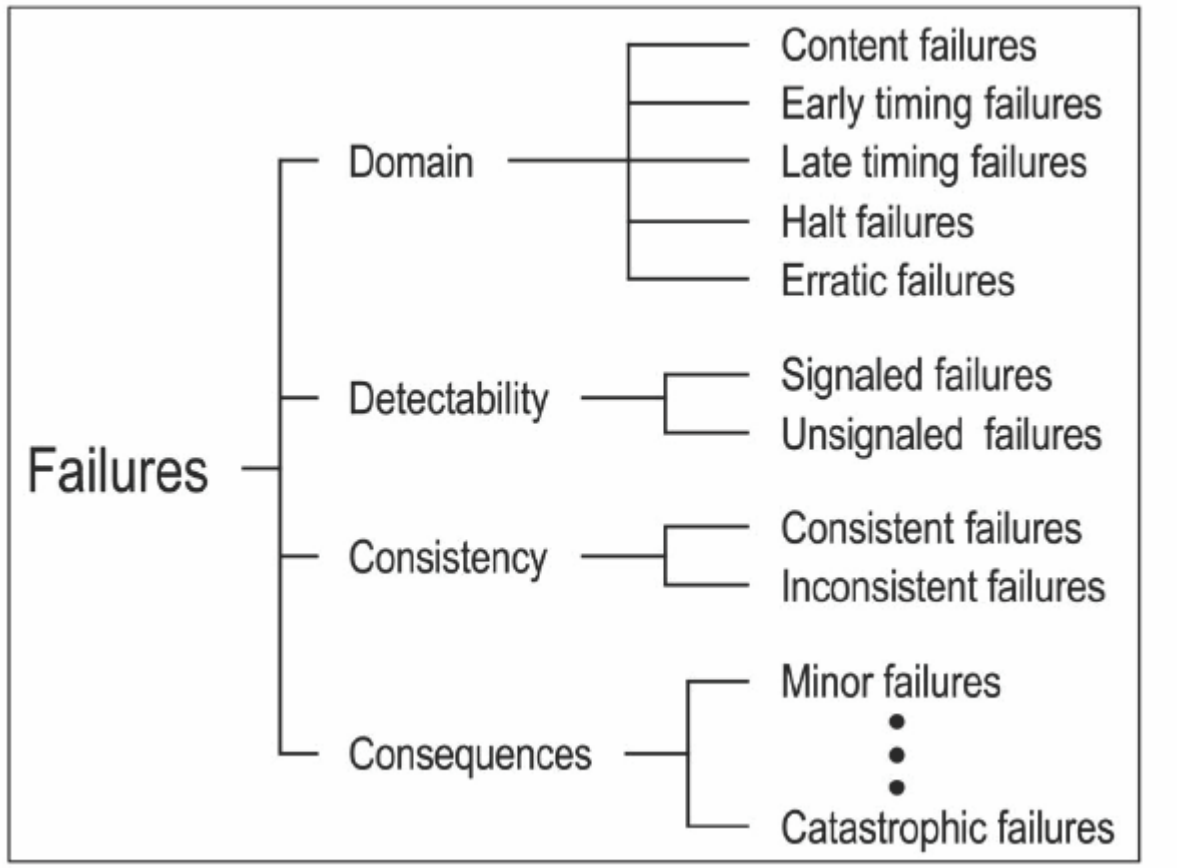
- The consistency of failures leads us to distinguish, when a system has *two or more users*

  - consistent failures: The incorrect service is perceived identically by all system users
  - inconsistent failures: Some or all system users perceive differently incorrect service and some users may actually perceive correct service (Byzantine failures)

Consequence viewpoint

- Grading the consequence of the failures upon the system environment enables failure *severities* to be defined
- Two levels can be defined:
  1. minor failures: the harmful consequences are of similar cost to the benefits provided by correct service delivery
  2. catastrophic failures: the cost of harmful consequences is higher than the benefit provided by correct service delivery

Service failure modes



Development Failures

principle causes: incomplete or faulty specifications, user initiated specification changes, faulty estimates of development costs…

- Complete development failures: the development process will be terminated before the use phase
  - budget failure
  - schedule failure
- Partial development failures: lesser severity than project termination
  - Downgrading
  - Overrun

Errors

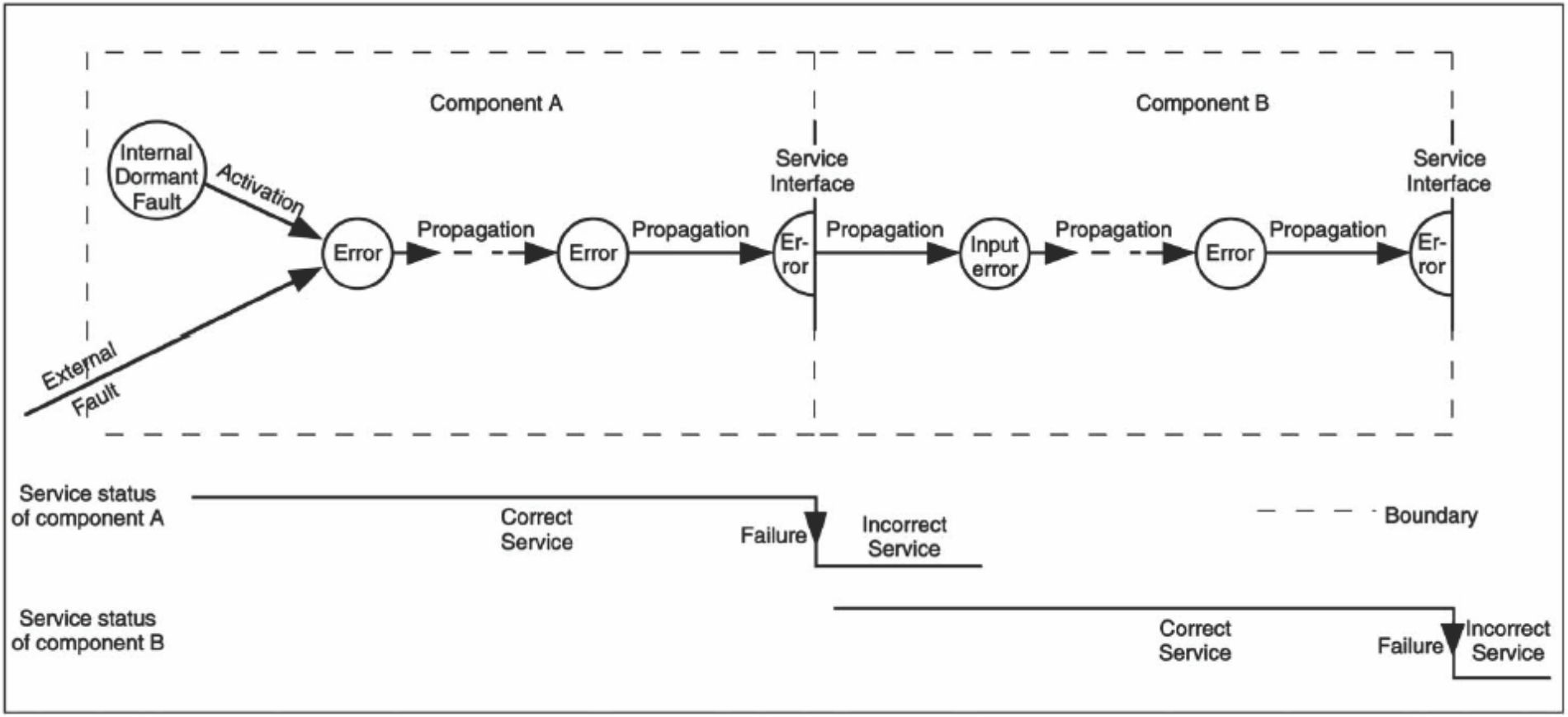
- A failure occurs when the error causes the delivered service to deviate from correct service
- An error is detected if its presence is indicated by an *error message* or *error signal*.
- Errors that are present but not detected are latent errors.

Errors (cont'd)

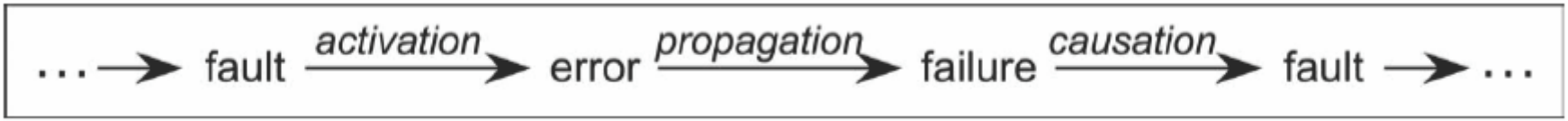
- An error will actually lead to a service failure or not depends on two factors:
  - The structure of the system
    - nature of any redundancy that exists in it
  - The behavior of the system
    - the part of the state that contains an error may never be needed for service

  - Single errors are errors that affect only one component.
  - multiple related errors are errors that affect more than one component simultaneously.

The Pathology of Failure



The Pathology of Failure



The arrows in this chain express a causality relationship between ***faults***, ***errors*** and ***failures***

Fault Activation Reproducibility

- The ability to identify the activation pattern of a fault that had caused one or more errors is the fault activation reproducibility

- Faults can be categorized according to their activation reproducibility:
  - Solid or Hard faults: faults whose activation is reproducible
  - Elusive or Soft faults: faults whose activation is not reproducible