Software Reliability and Security

Module 6

Winter 2017

Presentation/Lecture Schedule and Report Due Dates

- Presentation 1
  - Related background paper
  - Jan 27, Feb 1, 3
- Presentation 2
  - Project proposal
  - March 1, 3, 8
- Presentation 3
  - Final project report
  - March 24, 29, 31

- Lectures
  Jan 13, 18, 20, 25, 27
  Feb 1, 3, 8, 10, 15, 17
  March 1, 3, 8, 10, 15, 17, 22, 24, 29, 31

- Project Proposal Due
  Tuesday, February 28

- Final Project Report Due
  Monday, April 10

- Final Exam
  Wednesday, April 12, 10:00am

Outline

- Dependability – A Generic Concept
  - Attributes
  - Impairments
  - Means
- The Impairments to Dependability
  - Faults, errors, and failures
  - Classifications of faults, errors, and failures
- Means for Dependability – Fault Tolerance
  - Phases of fault tolerance
  - Approaches for software fault tolerance

Dependability as a Generic Concept

- Historically many disciplines considered other related disciplines as special cases
- Similar trends for reliability, safety, and …
- An wider concept – due to the complex nature of current system's quality of service – "Dependability"
  - Available: readiness for usage
  - Reliable: continuity of service
  - Safe: avoidance of catastrophic consequences
  - Secure: unauthorized access and/or handling of information

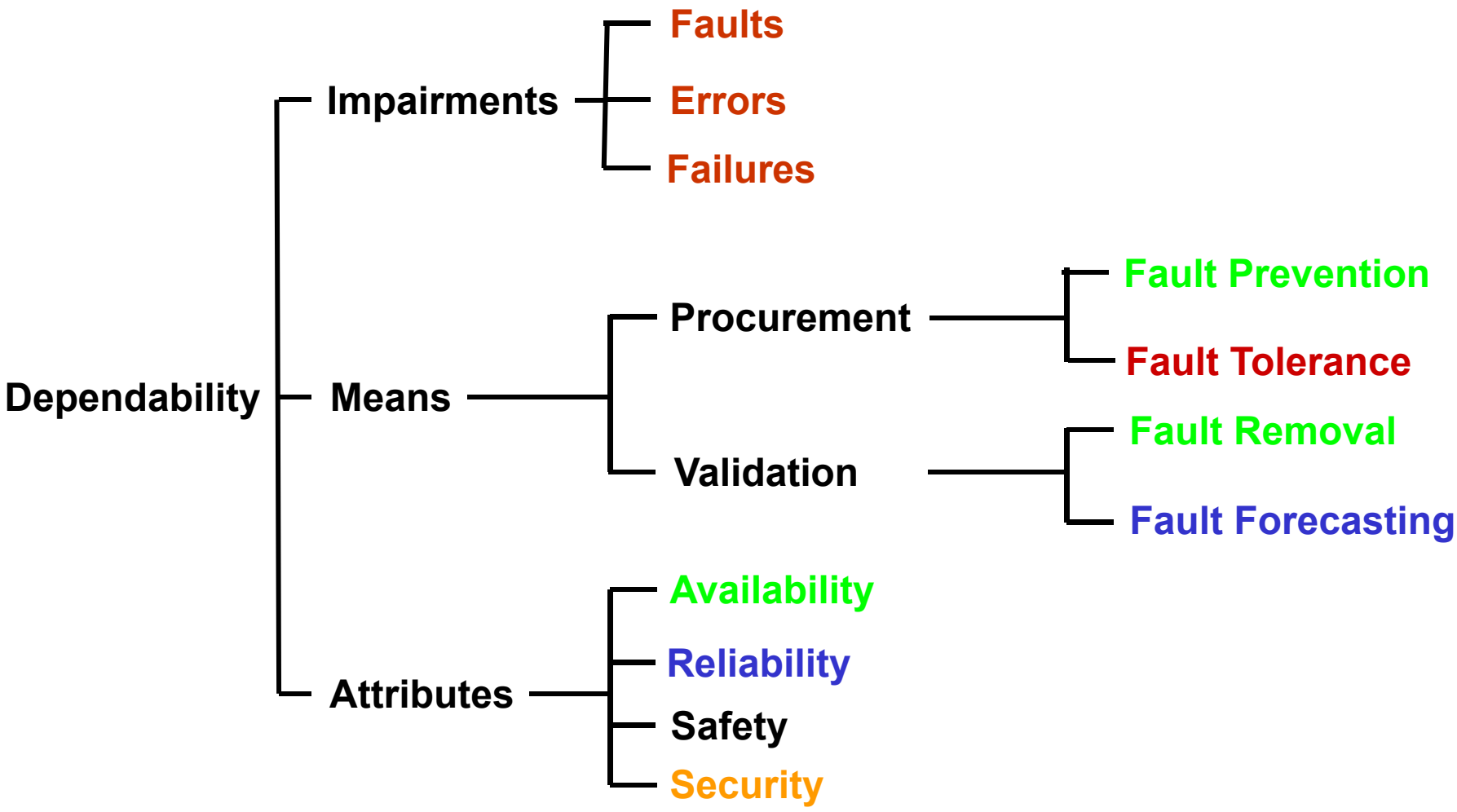
Dependability

- Defined by Working Group 10.4: Dependable Computing and Fault Tolerance of International Federation for Information Processing (IFIP)
- Trustworthiness of a computer system so that we can rely on the delivered service
  - The service is its behavior as it is observed by its user(s)
  - A user may be another system (human or physical)

Why is this course not called as "Software Dependability"?

- The term "dependability" is still not well-known among the software engineering community
- The term "reliability" and "security" are more popular in relevant disciplines such as software engineering
- Reliability and security issues of software are addressed while safety and availability are not addressed directly
- Books on "Reliability" & "Security" mainly focus on
  - How to evaluate, measure, and predict the reliability of systems (reliability)
  - How to protect computer networks (security)
  - Methods and issues for building reliable and secure software systems are ignored (except a few)

The Dependability Tree



Dependability: Basic Definitions

- Impairments to dependability
  - Failure: Delivered service does not meet its specification (predefined expected service)
  - Fault: Hypothesized cause of an error
  - Error: A system state which may lead to subsequent failure

Fault Pathology – A Fundamental Chain

- Error
  - An error is "latent" when it has not been discovered ("detected")
  - An error may disappear before detection
  - An error usually propagate and may create new errors
  - During operation, the existence of faults is determined by detecting errors
- Failure
  - A failure occurs when an error "passes through" the system-user interface and affects the service delivered by the system
  - A failure often caused from the combined action of several faults

Fault Pathology – A Fundamental Chain

- Fault
  - A fault is active when it produces an error
  - Most internal faults cycle between their dormant and active states
  - A given fault in a component may result from different sources
  - A type of fault may create another type of fault through error propagation

Fault Pathology – contd.

- A Fundamental Chain
- … --> failure --> fault --> error --> failure --> fault --> …
- An Example
  - A memory cell always returns the value 0 independent of what is stored in it – it contains a fault
  - This fault may not manifest as an error until that faulty memory cell is used for storing 1 in it
  - An error may be overwritten before creating an error or a failure
  - If you use AND operation with another cell which contains 0, it may not appear as failure (result of an AND operation: X AND 0 = 0)

Error, Fault, and Failure

- Different people may view a failure differently
  - If there is no definition in the requirements specification
- There exist other special situation terminologies
  - Faults – bugs, defect (no clear difference between fault and failure) deficiency
  - Failures – breakdown, malfunction, denial-of-service, outage

Dependability: Basic Definitions – contd.

- Means for Dependability
  - Procurement: enable the system to deliver a specified service
    - Fault prevention – prevent fault occurrence or introduction
    - Fault tolerance – provide a specified service in spite of faults
  - Validation: certify that the system delivers a specified service
    - Fault removal – reduce the number or consequence of faults
    - Fault forecasting – estimate the future number or consequence of faults

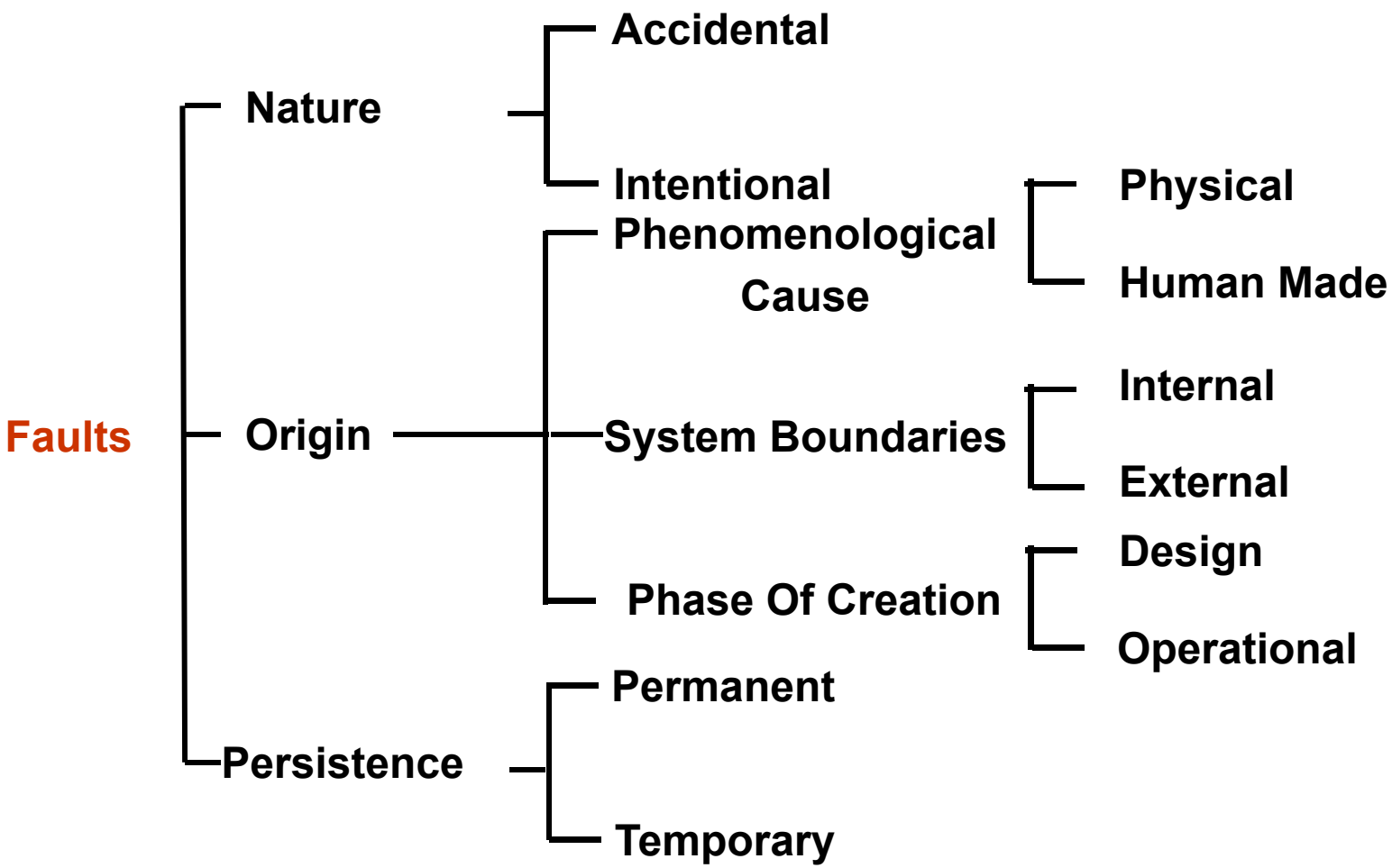
Fault Classification

- Based on origin
  - Phenomenological causes
    - Physical – due to adverse physical phenomena
    - Human made – due to human imperfection
  - System boundaries
    - Internal – parts of the system state that generates an error
    - External – due to the interference with its physical (electromagnetic, radiation, etc.) or human environment
  - Phase of creation with respect to system's life
    - Design – from requirement specification to implementation phases
    - Operational – appear at system's runtime

Fault Classification – contd.

- Based on nature
  - Accidental
  - Intentional
- Based on temporal persistence
  - Permanent
  - Temporary
- There may be different combinations of the above faults – as many as 32 - only 11 are usually used in practice

Classification of Faults – contd.



Combinations of Faults – contd.

Nature		Phenomenological Cause		System Boundaries		Phase of Creation		Persistence		Conventional Labeling
Accidental	Intentional	Physical	Human made	Internal	External	Design	Operational	Permanent	Temporary	
X		X		X			X	X		Physical
X		X			X		X	X		
X		X			X		X		X	Transient
X		X		X			X		X	Intermittent
X			X	X		X			X	
X			X	X		X		X		Design
X			X		X		X		X	Interaction
	X		X	X		X		X		Malicious Logic
	X		X	X		X			X	
	X		X		X		X	X		Intrusions
	X		X		X		X		X	

Types of Failures

- Types of failures from three perspectives
  - Failure Domain
  - Failure Perception
  - Consequences of the Failures – Failure Severities

Failure Types - Failure Domain

- Value failures – the output value is not consistent with the specification
- Timing failures – the timing of output does not meet the specification: late or early
- Stopping failures – a constant value (e.g., last correct value, some predetermined value) is delivered
  - Related to both value and timing failures
  - A system whose failures are stopping failures is called a fail-stop system
- Omission failures – a special case of stopping failures where no service is delivered
  - A common limiting case for both value (null value) and timing failures (infinitely late failures)
  - Crash failures – a persistent omission failure
  - A system whose failures are crash failures is called a fail-silent system

Failure Types - Failure Perception and Severities

- Failure Perception
  - Consistent failures – all users have the same perception of the failures
  - Inconsistent failures – Also called Byzantine failures. The system users may have different perceptions of a given failure (arbitrary failure)
- Consequences - Failure Severities
  - Benign failures – the consequence are of the same order of magnitude (generally in terms of cost) as the benefit provided by the service delivery in the absence of failure
    - A system whose failures are benign failures is called a fail-safe system.
  - Catastrophic failures – the consequences are incommensurably greater than the benefit provided by service delivery in the absence of failure.

Summary

- Concept of Dependability
  - Attributes, impairments, and means
- The Impairments to Dependability
  - Faults, errors, and failures
  - Classifications of faults, errors, and failures
- Phases of Fault Tolerance
- Software Fault Tolerance Approaches

Lecture Sources

- J.C. Laprie, Dependabilty: Basic Concepts and Terminology in English, French, German, Italian and Japanese, Springer-Verlag, NY, 1991.
- Dependable Computing and Fault Tolerance: Concepts and Terminology, J.C. Laprie, FTCS-15, IEEE 1985.
- Pankaj Jalote, Fault Tolerance in Distributed Systems, Prentice-Hall, New Jersy, 1998, Chap 1 and 6.