

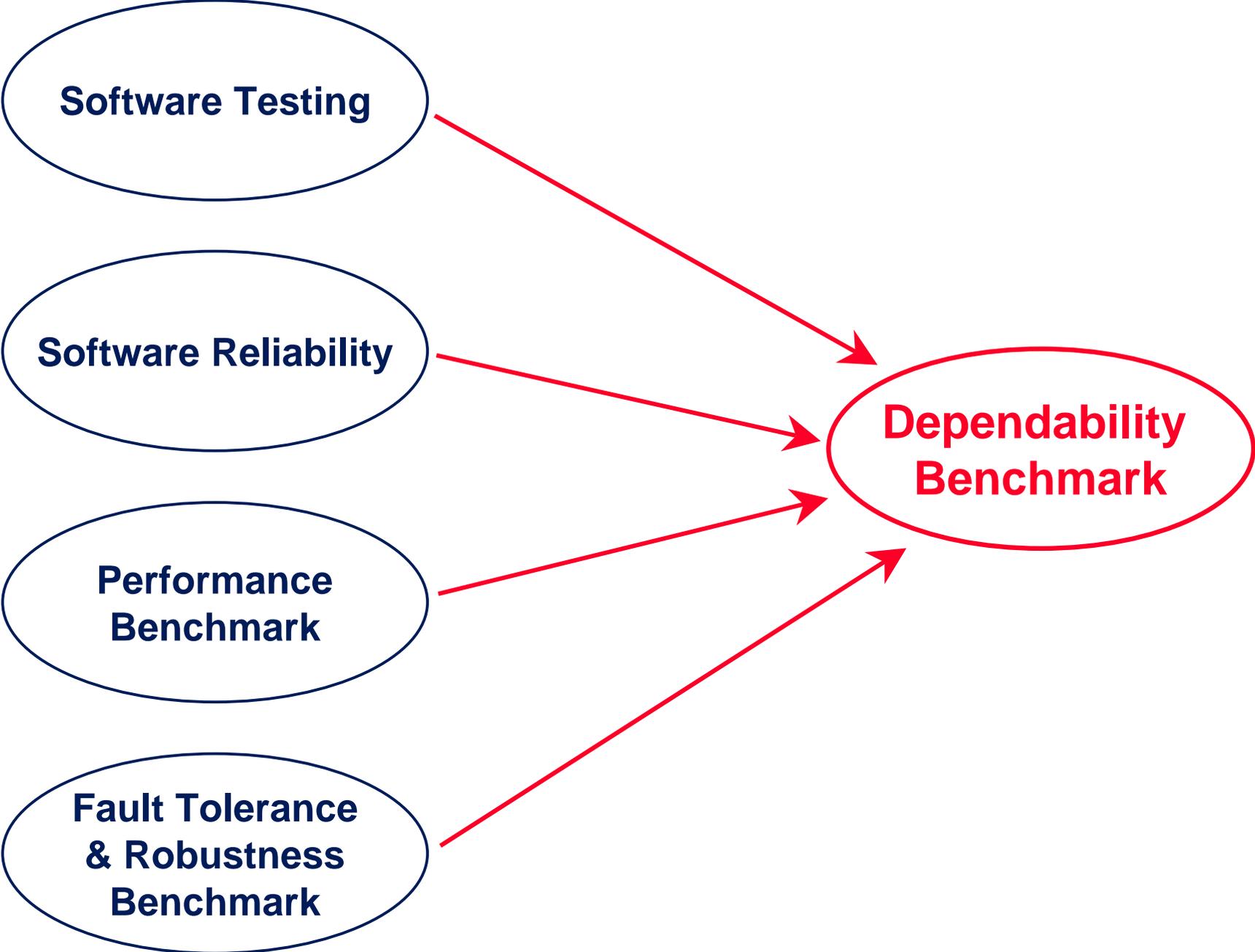
# **Panel: Software Testing & Evaluation and Dependability Benchmarking**

**(Goal: discuss contact points between software testing & evaluation and dependability benchmarking and find possible contribution from these well established areas to dependability benchmarking)**

**Karama Kanoun**



**39th Meeting IFIP Working Group 10.4 — Paraty, Brazil — 28 February-3 March, 2001**



# What is expected from a Software Dependability Benchmark?

## Software characterization with respect to:

- ◆ **Internal faults**
- ◆ **External faults**
  - ➡ other software component(s)
  - ➡ hardware

## Aims

- ◆ **Properties + quantification of some specific measures**
- ◆ **Avoid undesirable behavior** → “environment” modification
- ◆ **Enhancement**
  - ➡ Correction
  - ➡ Wrapping

# Software Testing

## ☞ Aims

- ◆ **Activate / identify faults ⇒ correction**
- ◆ **Characterize software behavior**

## ☞ Various kinds of tests for characterization

### ◆ **Functional (statistical / operational) testing**

- ➡ **Validate functionality under ordinary operating conditions (typical load)**
- ➡ **Software reliability evaluation**

### ◆ **Load testing**

- ➡ **Performance under heavy load (peak, worst case)**

### ◆ **Robustness / stress testing**

- ➡ **Pushes the software beyond its specified limits**  
(invalid inputs, stressful environmental conditions)

# Software Reliability

## Aim

- ◆ Evaluate software reliability measures (MTTF, failure intensity)

## Means

- ◆ Failure data collection and processing

From functional test

## Advantage

Early estimation

## Limitation

Representative operational profile

Duration ↔ accuracy

# Software Reliability

## 👉 Aim

- ◆ Evaluate software reliability measures (MTTF, failure intensity)

## 👉 Means

- ◆ Failure data collection and processing

From functional test

Operational / field data

## 👉 Advantage

Early estimation

Accuracy / representativeness

## 👉 Limitation

Representative operational profile

Too late?

Duration ↔ accuracy

# Software Performance Benchmark

## ☞ Most of the time

Performance benchmark = system benchmark

## ☞ Aims of software performance benchmark

◆ Measure system performance (or price / performance) to compare:

➡ different software products on the same machine

Competition between software vendors

➡ different releases/versions of the same product on the same machine

Performance improvement?

## ☞ Two categories of benchmarks

◆ **Coarse-grain benchmarks** ⇒ execution times of the entire application

◆ **Fine-grain benchmarks** ⇒ execution time (rate) of specific operations

# Fault Tolerance & Robustness Benchmarks

## ☞ Fault tolerance benchmarks

- ◆ Ability to tolerate faults
- ◆ Effectiveness of error detection and recovery mechanisms
- ◆ Performance degradation due to faults

## ☞ Robustness benchmarks

### ◆ Ability to tolerate / resist to unexpected conditions caused by:

- ➡ Hardware failures
- ➡ Other user programs (system call with illegal parameters)

### ◆ Failure modes

= Robustness test + repeatable

## ☞ Fault injection

# Conclusion



## Conclusion



👉 **Dependability benchmarks rely on:**

development / validation process & field information

+ additional specific work

👉 **What about COTS?**

well tested

**BUT information not available all the time**