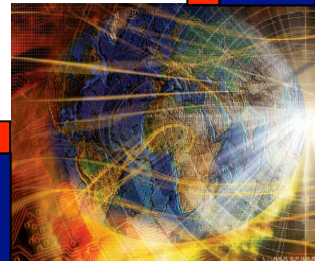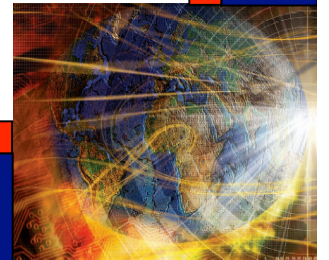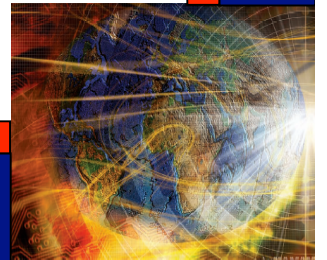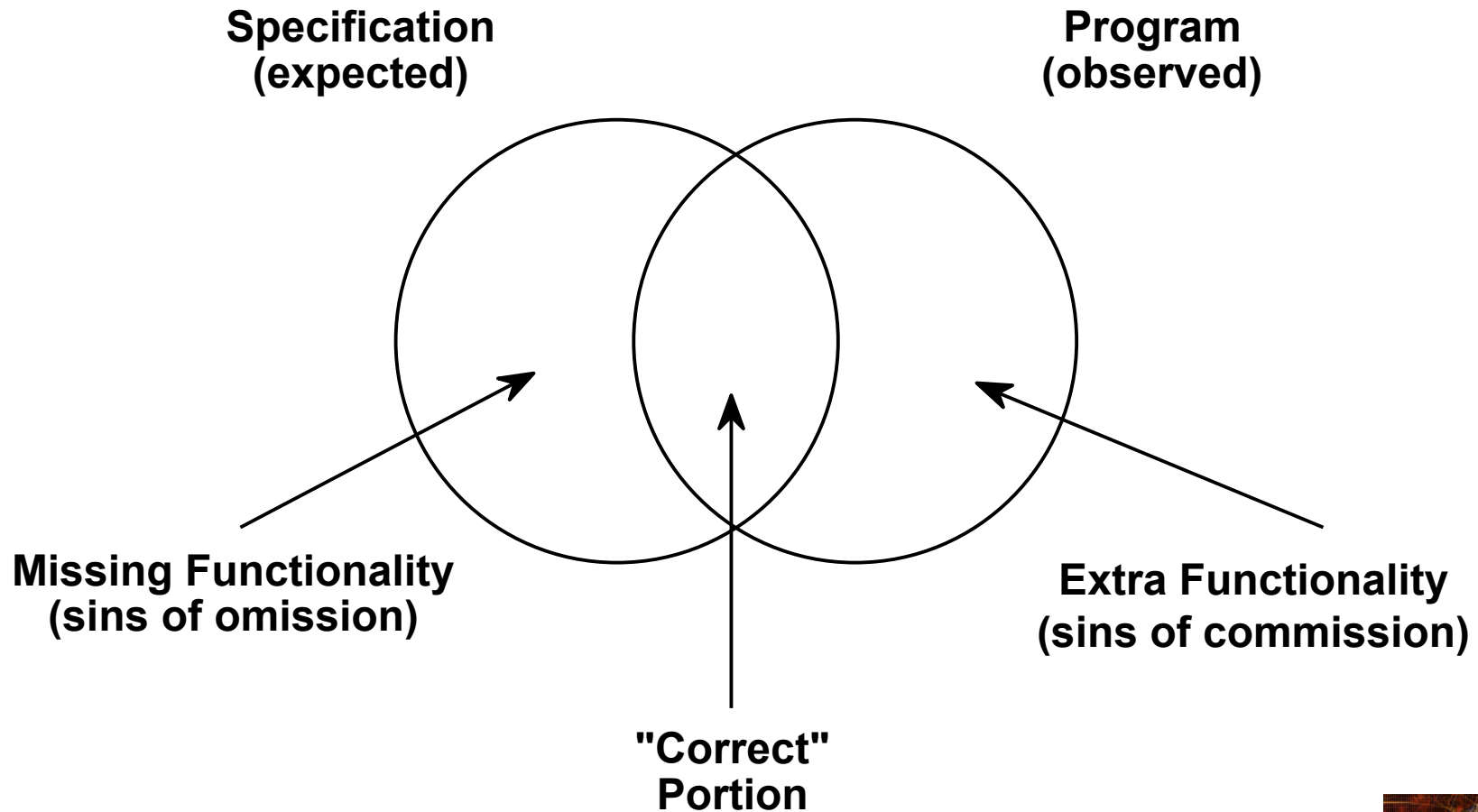# Chapter 1

Introduction

Testing

"   in the beginning of a malady it is easy to cure but difficult to detect, but in the course of time, not having been either detected or treated in the beginning, it becomes easy to detect but difficult to cure."
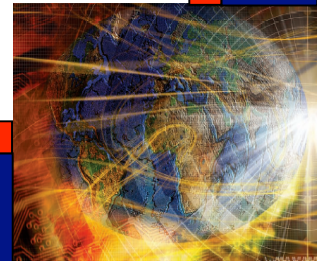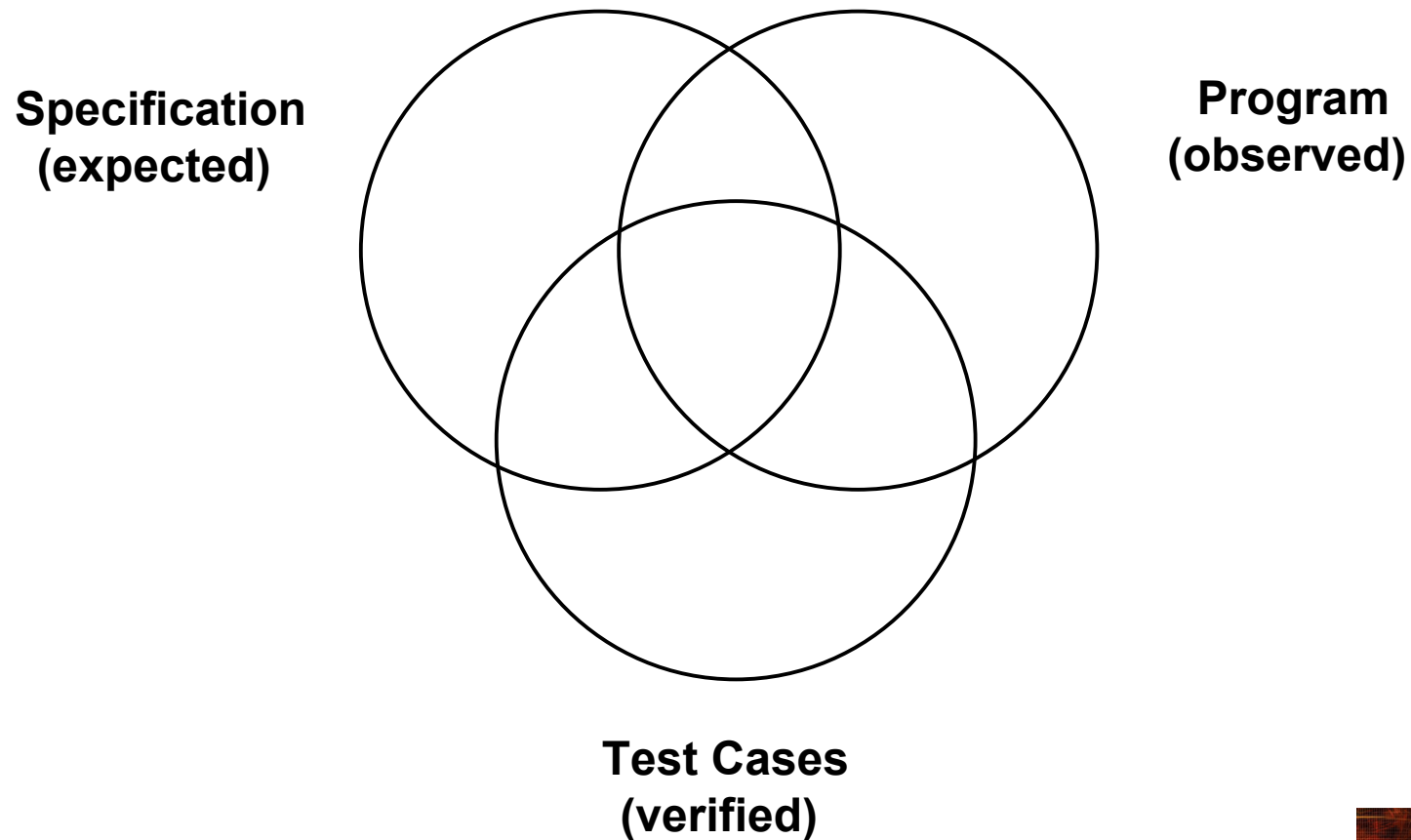
Nicolo Machiavelli
The Prince, 1513

Program Behaviors

Specification
(expected)

Program
(observed)

Missing Functionality
(sins of omission)

Extra Functionality
(sins of commission)

"Correct"
Portion

Correctness

- **Impossible to demonstrate**
- **A term from "classical" computer science**
  - **"proofs" derived from code**
  - **Not derived from specification**
  - **Can only prove that the code does what it does!**
- **Better viewpoint: a relative term—program P is correct with respect to specification S.**
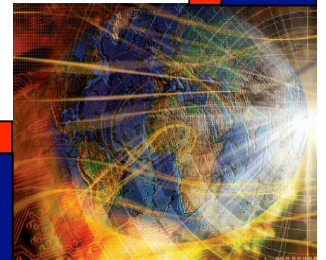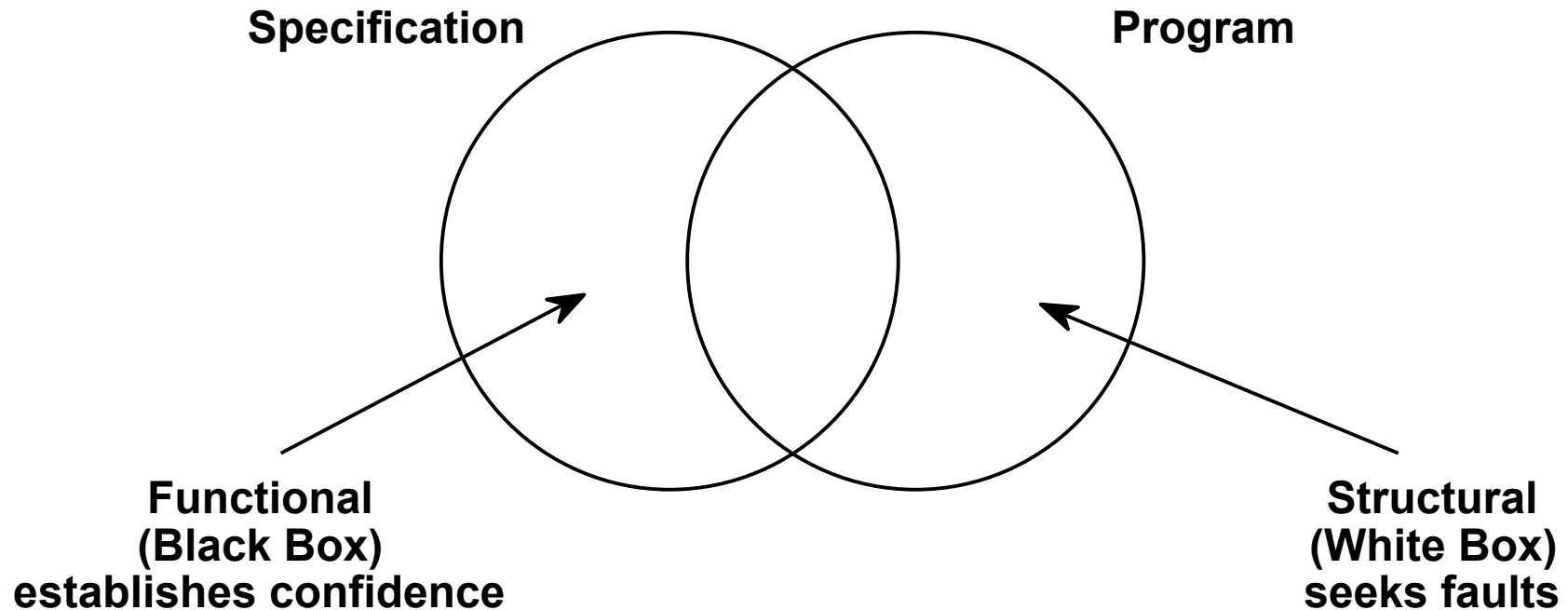- **Bottom Line: do the specification and the program meet the customer/user's expectations?**
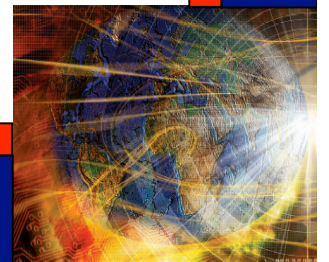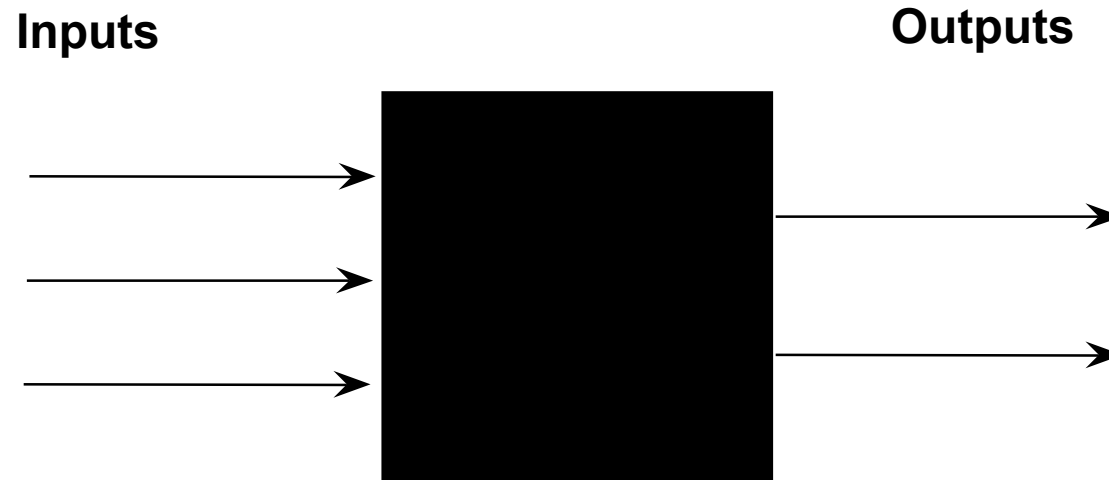
Testing Program Behavior

**Specification
(expected)**

**Program
(observed)**

**Test Cases
(verified)**

Basic Approaches
Preferred terms: Specification-based and code-based testing

Specification   **Program**

**Functional
(Black Box)
establishes confidence**

**Structural
(White Box)
seeks faults**

Black Box
A term borrowed from engineering

Inputs

Outputs

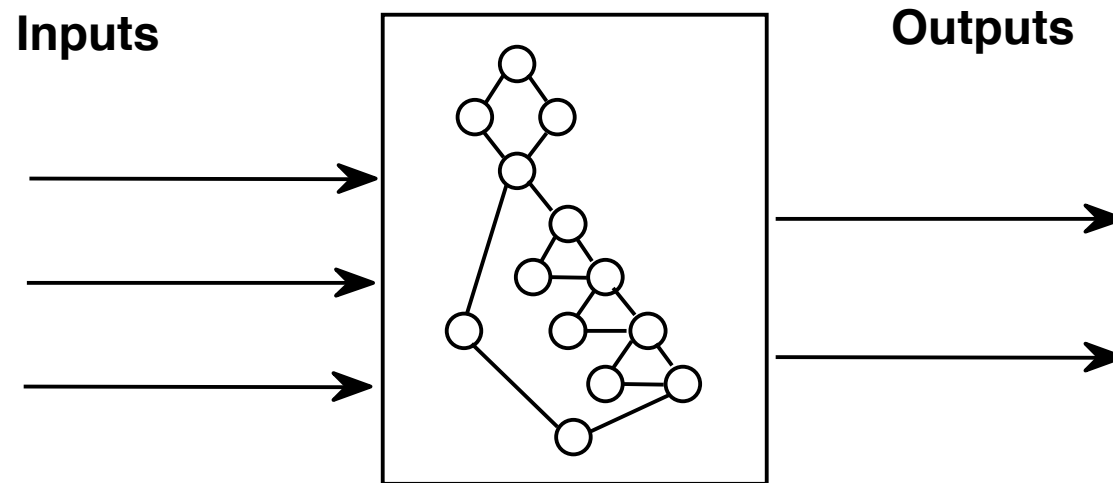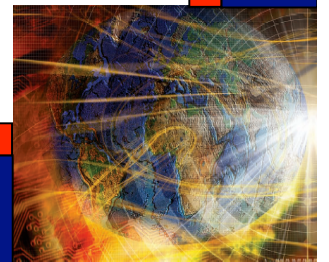**Function is understood only in terms of it's inputs and outputs, with no knowledge of its implementation.**

White (Clear) Box

Inputs

Outputs
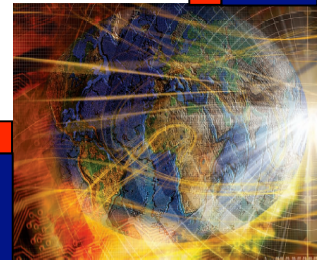
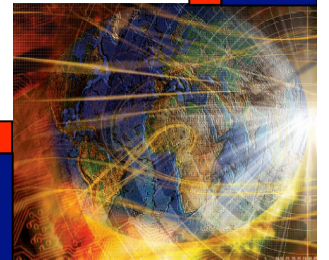**Function is understood only in terms of its implementation.**

IEEE Testing Terminology

```
Error
   ↓
  Fault
     ↓
   Failure
      ↓
    Incident
```
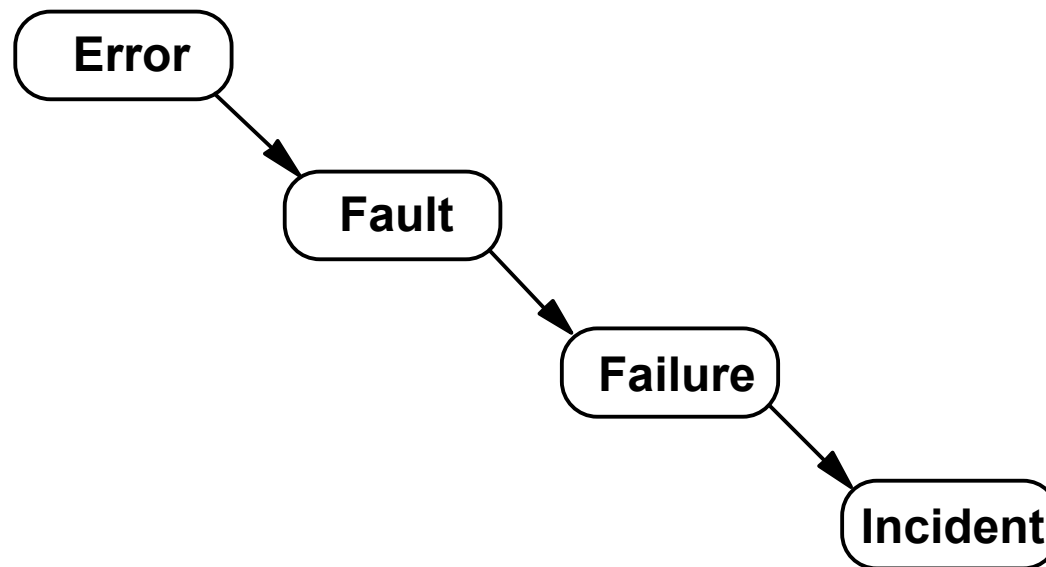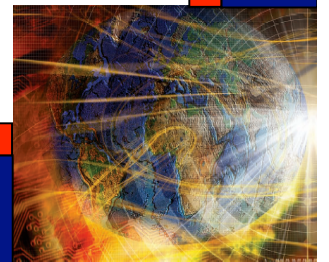
What is a software test?

- An experiment,
- Designed to reveal the presence of a fault,
- By causing a failure when executed by the code being tested.
- *n.b.* a test can never reveal the absence of a fault.
- A test method is a repeatable way to generate test cases.

Content of a Test Case

- **"Boilerplate": author, date, purpose, test case ID**
- **Pre-conditions (including environment)**
- **Inputs**
- **Expected Outputs**
- **Observed Outputs**
- **Pass/Fail**
  - Common usage: Pass when expected and observed outputs agree
  - BUT Myers [*The Art of Software Testing]* defines a successful test as one that reveals a fault, i.e., expected and observed outputs disagree.
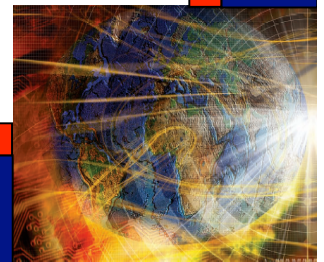
How Might a Test Case Fail?

- **False positive?**

- **False negative?**

- **What do these really mean?**

- **Exercise**
  - **Analyze false positive, false negative**
  - **Suggestion: use a decision table with conditions:**
    - **Expected output correct?**
    - **Observed output correct? (potential circularity here)**

- **Compare with common medical usage**