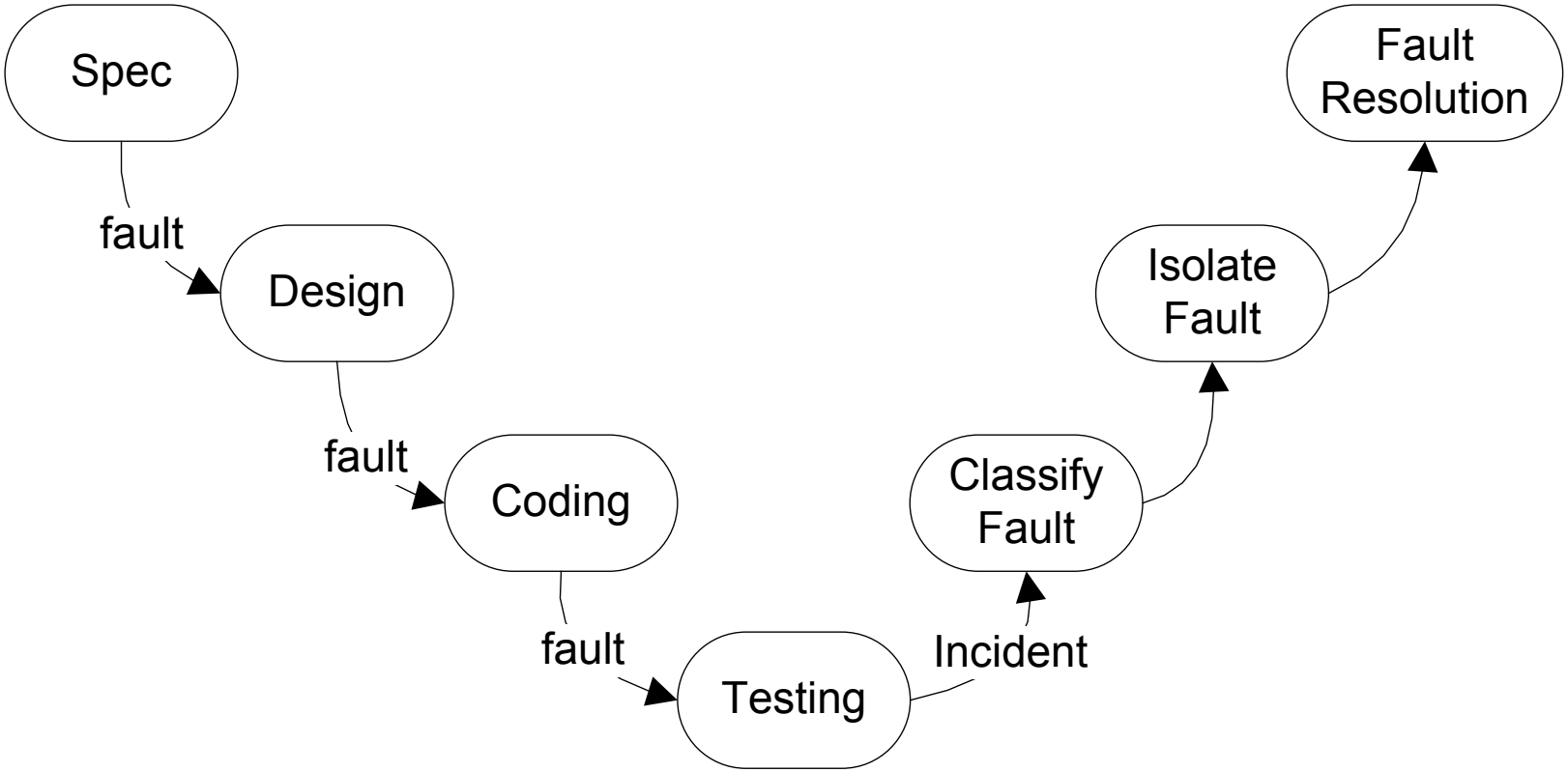
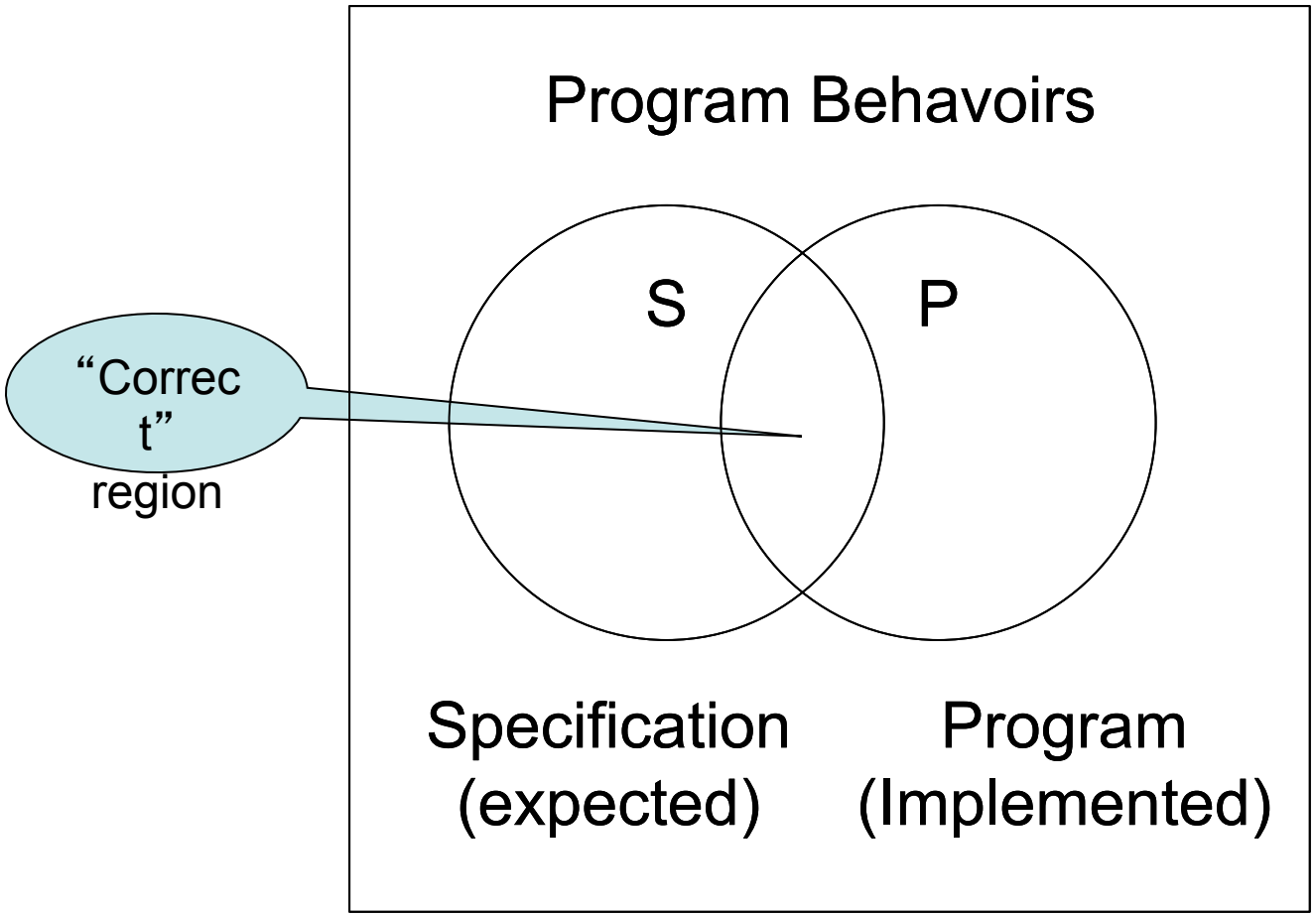
Chapter 1

Introduction

Errors, Faults, Failures and Incidents



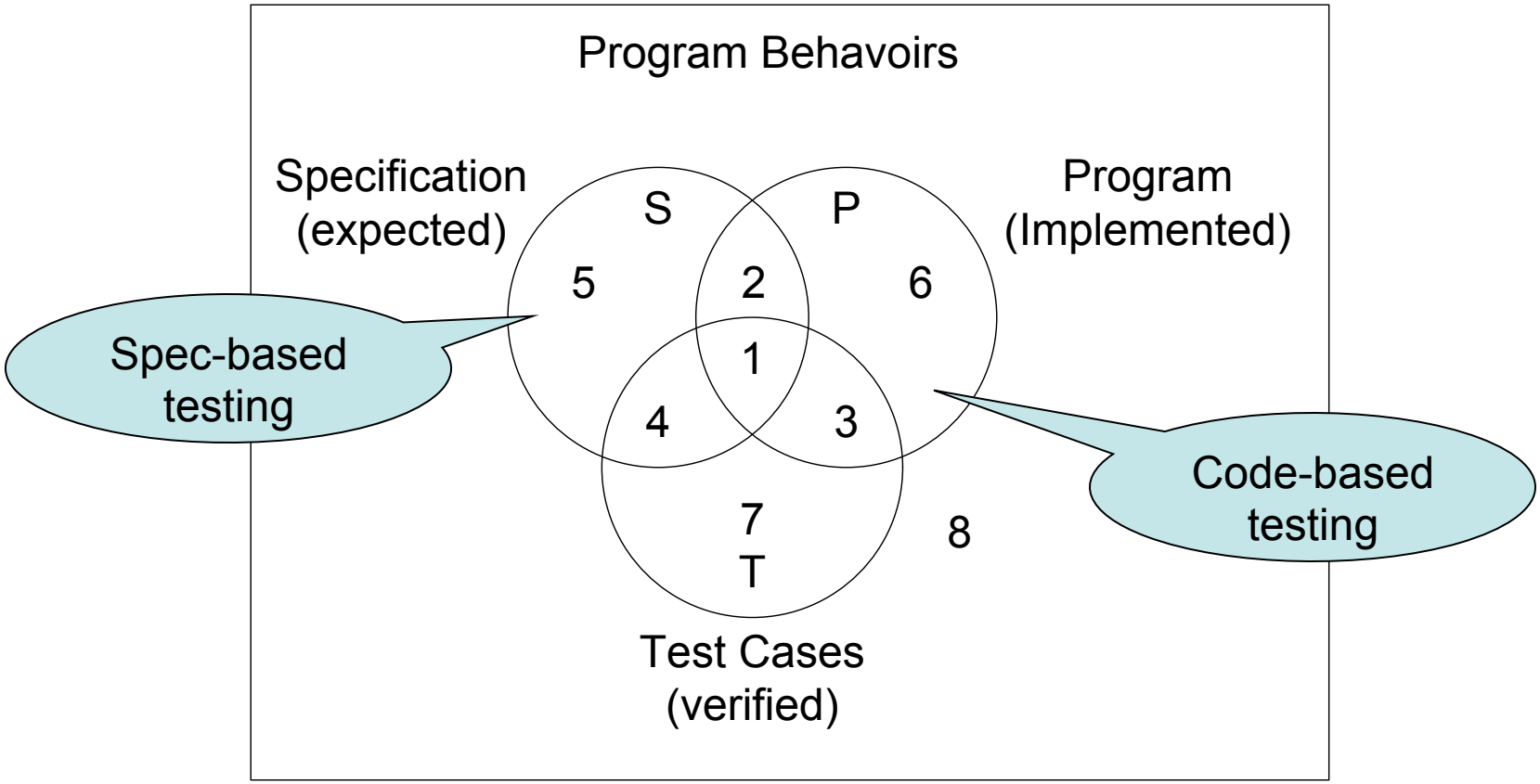
Universe of Discourse—Program Behaviors



About Correctness...

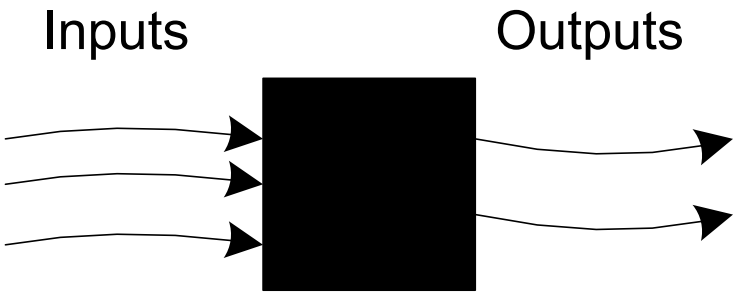
- Impossible to demonstrate
- A term from “classical” computer science
  - “proofs” derived from code
  - Not derived from specification
  - Can only prove that the code does what it does!
- Better viewpoint: a relative term—program P is correct with respect to specification S.
- Bottom Line: do the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

Testing and Program Behaviors



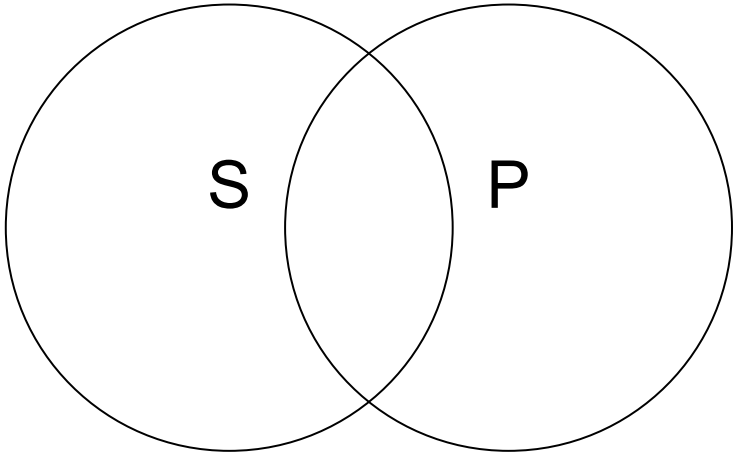
Synonyms—Metaphors gone wild

- Specification-based testing
  - Functional testing
  - Black box testing
- Code-based testing
  - Structural testing
  - White box testing
  - Glass box testing
  - Clear box testing



An engineer's Black Box: a system understood in terms of its inputs and outputs

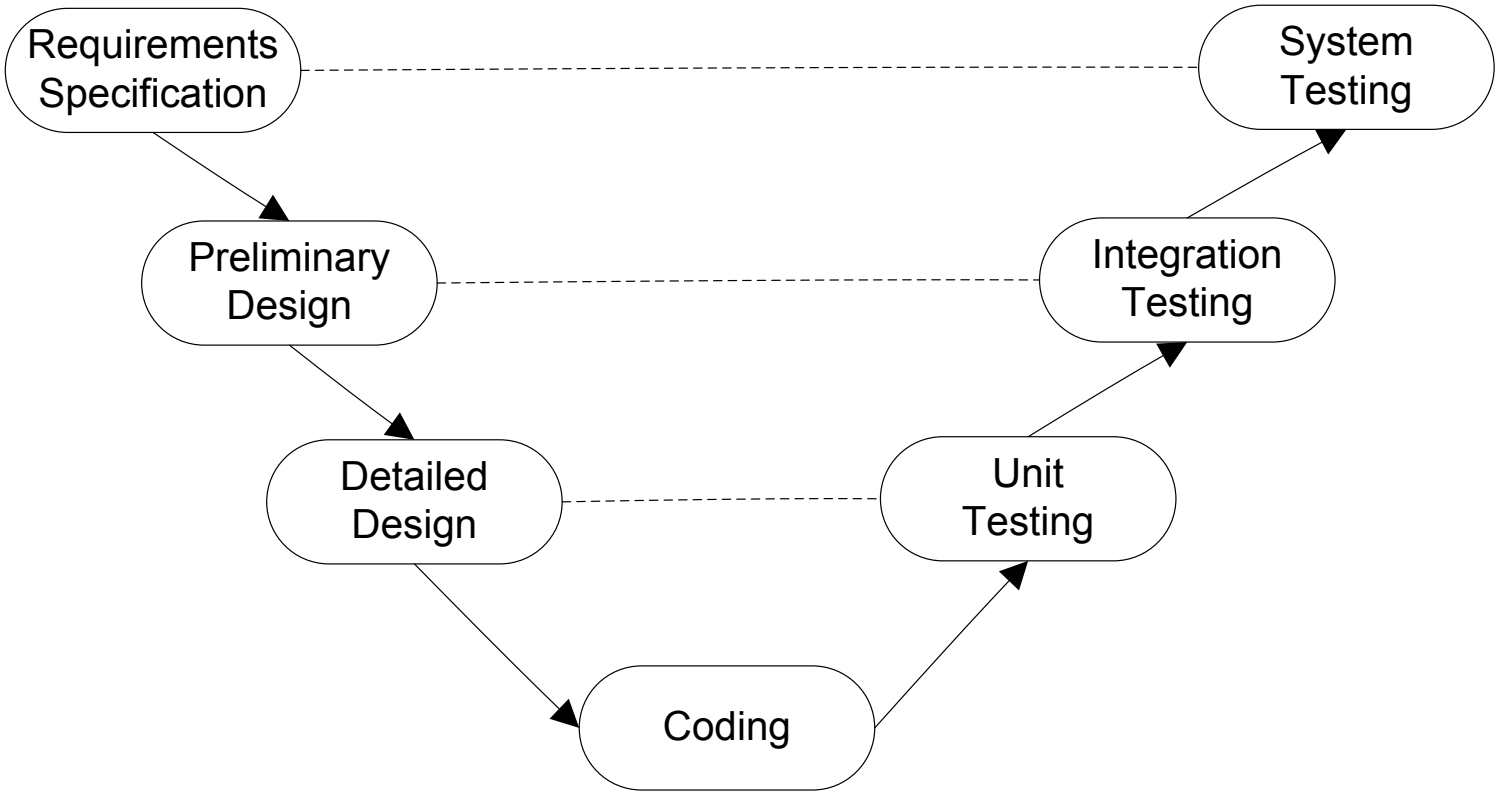
Program Behavoirs



Spec-based
Functional
Black Box
(establishes confidence)

Code-based
Structural
White/Clear Box
(seeks faults)

Levels of Testing (as seen in the V-model)



Content of a Test Case

- Test case identifier (usually a short name for test management purposes)
- Name
- Purpose (e.g., a business rule)
- Pre-conditions (if any)
- Inputs
- Expected outputs
- Observed outputs
- Pass/fail?

Testing as an Experiment

- With respect to the Error, Fault, Failure, Incident framework, a test case is an experiment that...
  - is designed to anticipate an error
  - that is realized as a fault
  - that causes a failure
  - and is recognized by comparing expected and observed outputs.
- In this framework, test cases are intended to reveal incidents.

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