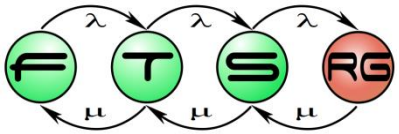
Introduction to testing

Zoltan Micskei, Istvan Majzik

Budapest University of Technology and Economics
Fault Tolerant Systems Research Group



Main topics of the course

- Overview (1)
  - V&V techniques, Critical systems
- Static techniques (2)
  - Verifying specifications
  - Verifying source code
- **Dynamic techniques: Testing (7)**
  - Developer testing, Test design techniques
  - Testing process and levels, Test generation, Automation
- System-level verification (3)
  - Verifying architecture, Dependability analysis
  - Runtime verification

Dynamic techniques

WHAT: code or other executable artefact

HOW: with execution

USING: testing, runtime verification…

WHAT IS TESTING?

Learning outcomes

- Recall different definitions of testing (K1)
- Explain goals and basic concepts of testing (K2)

Definition of testing (1)

„Testing is an activity performed for evaluating product quality, and for improving it, by identifying defects and problems.”

Source: IEEE, „Software Engineering Body of Knowledge” (SWEBOK) 2004
URL: http://www.computer.org/portal/web/swebok/

Definition of testing (2)

„An activity in which a system or component is executed under specified conditions, the results are observed or recorded, and an evaluation is made of some aspect of the system or component.”

Source: IEEE, "IEEE Standard for Software and System Test Documentation," *IEEE Std 829-2008* , 2008

Definition of testing (3)

„The process consisting of all lifecycle activities, both static and dynamic, concerned with planning, preparation and evaluation of software products and related work products

- to determine that they satisfy specified requirements,
- to demonstrate that they are fit for purpose and
- to detect defects.

Source: International Software Testing Qualifications Board (ISTQB), URL: http://istqb.org/

Definition of testing (4)

"Testing is the process of evaluating a product by learning about it through exploration and experimentation, which includes: questioning, study, modeling, observation and inference, output checking, etc."

Source: James Bach, Micheal Bolton. Exploratory Testing 3.0

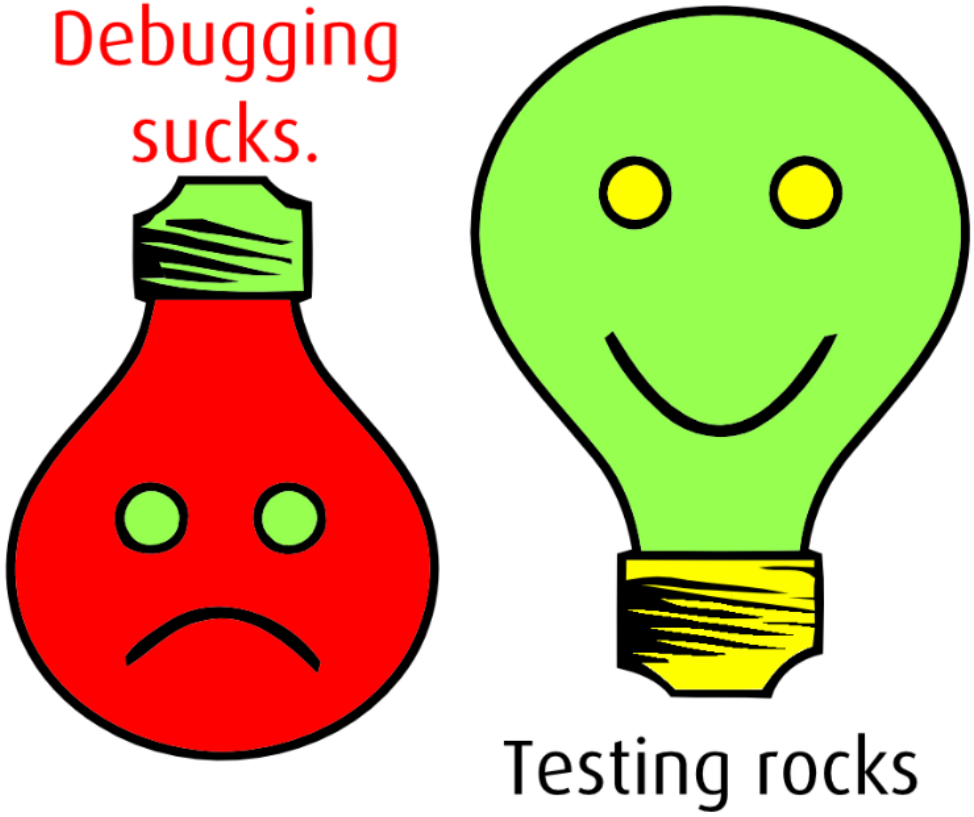
Debugging vs. Testing

Debugging

- Finding cause of the bug

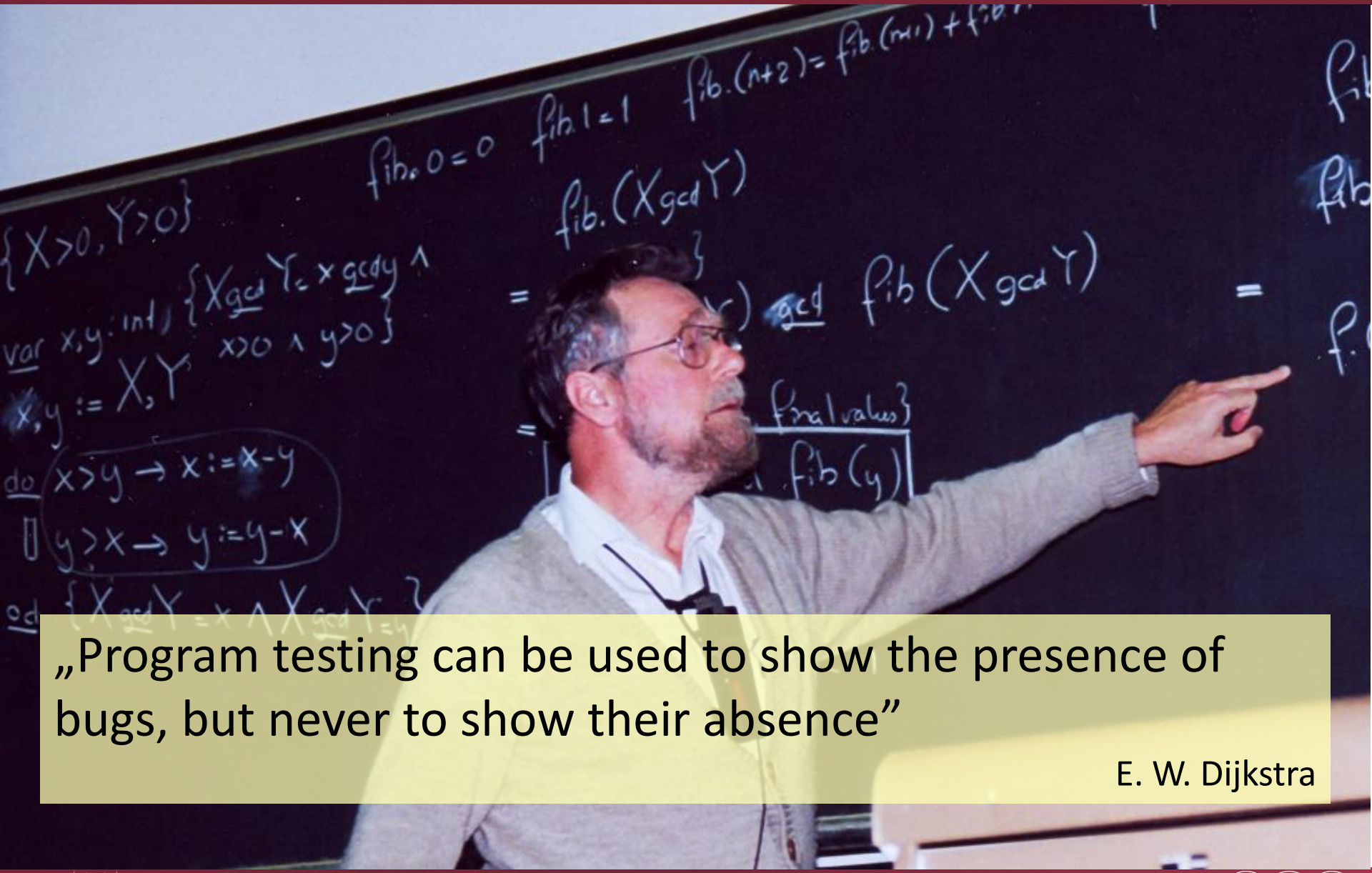
Testing

- Finding the bug



Source: Google Testing Blog

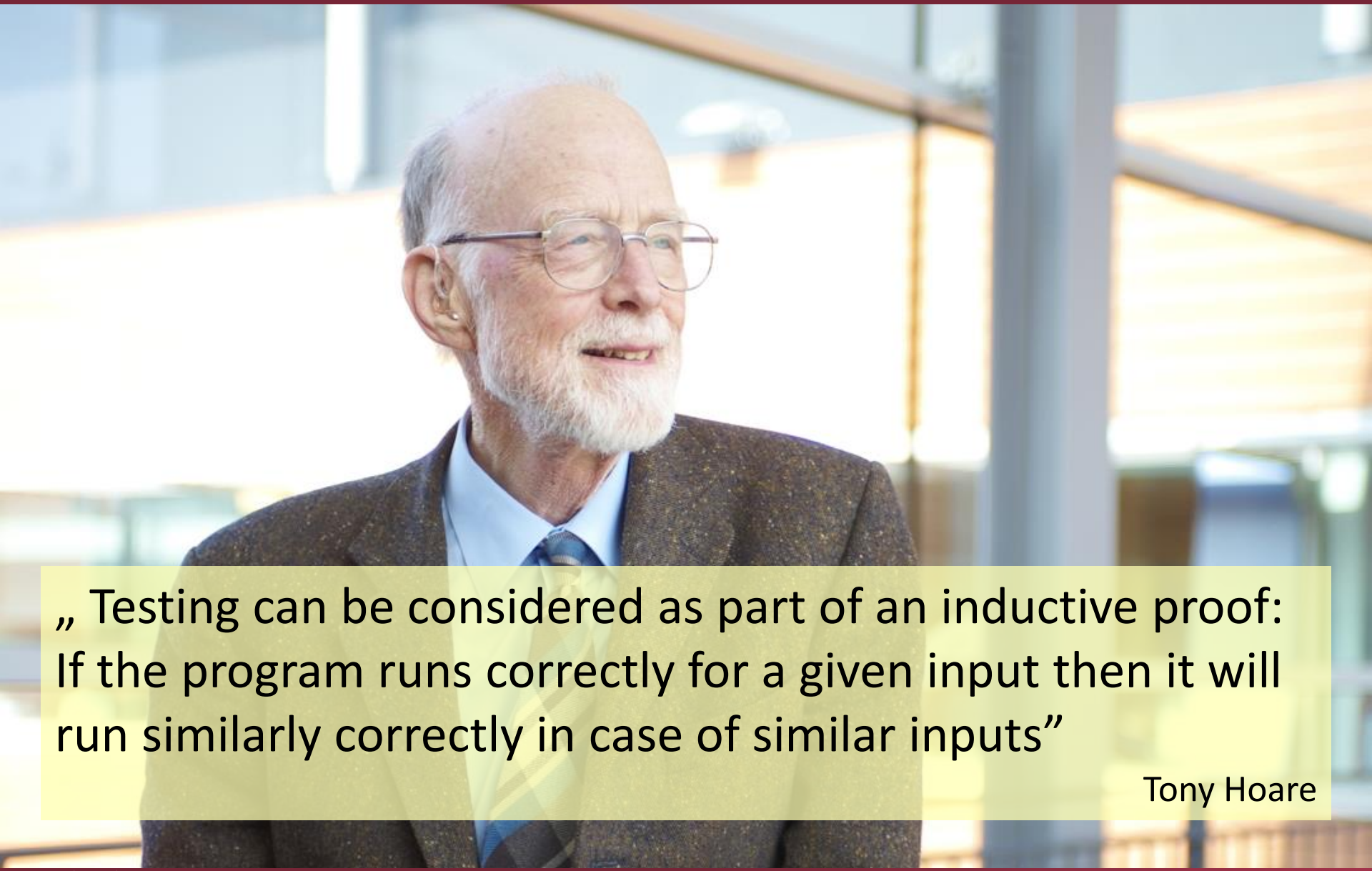
Quotes (1)



„Program testing can be used to show the presence of bugs, but never to show their absence”

E. W. Dijkstra

Quotes (2)



„ Testing can be considered as part of an inductive proof: If the program runs correctly for a given input then it will run similarly correctly in case of similar inputs”

Tony Hoare

Quotes (3)

„More than the act of testing, the act of designing tests is one of the best bug preventers known.”

Boris Beizer

Possible goals of testing

Testing “approaches”

Test-as-information-provider

- Test-last
- Independent test team
- Separate test phase
- Fixed releases

Test-as-quality-accelerant

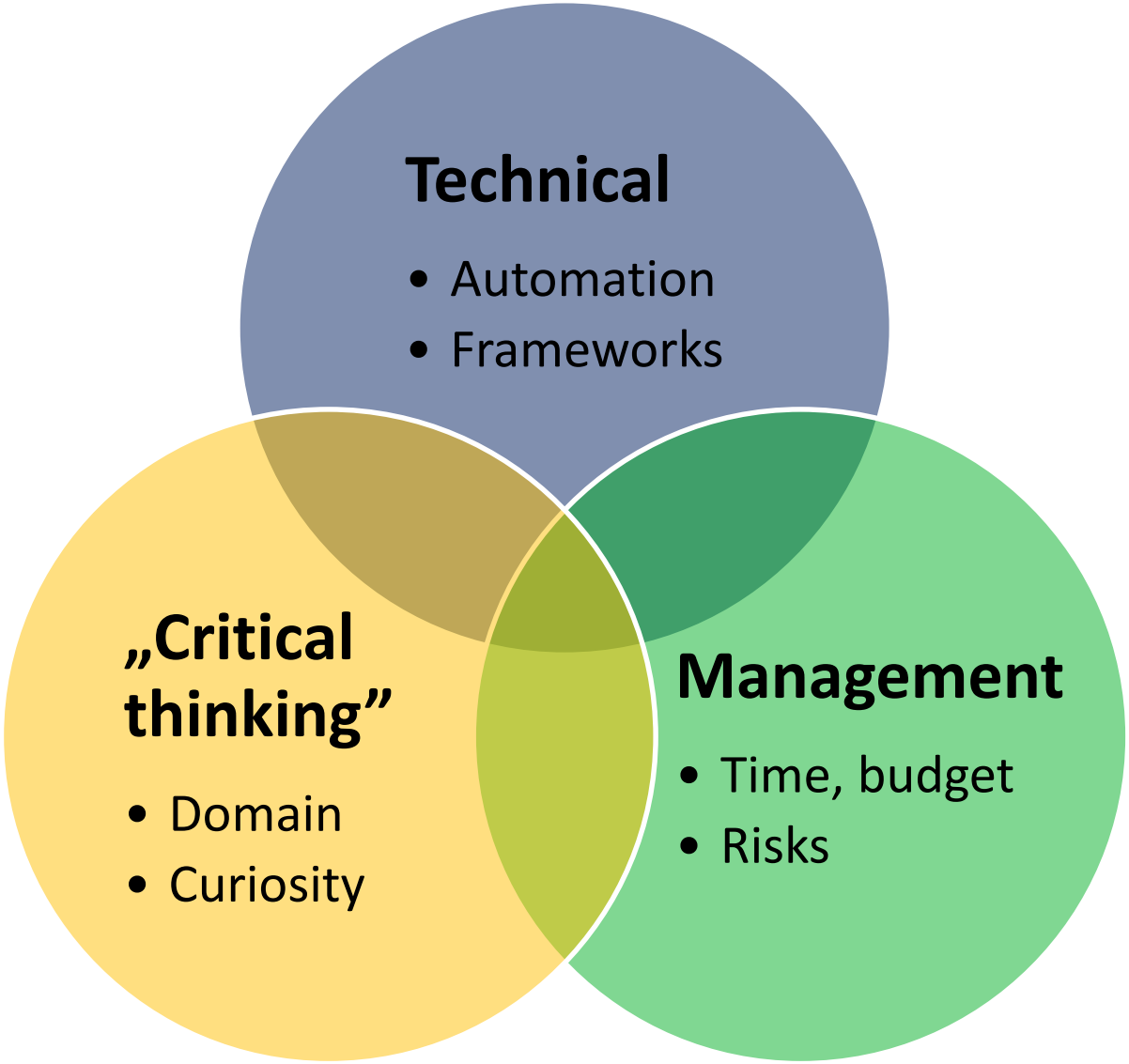
- Test-always
- Testers quality assistants
- Developers write tests
- Release often/always

Source: Alan Page. “Two new… schools”

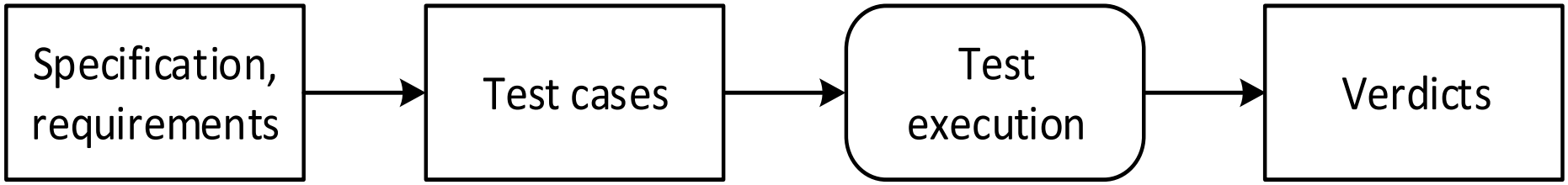
7 testing principles (ISTQB)

1. Testing shows presence of defects
2. Exhaustive testing is impossible
    1. *How many tests for a function with 3 int parameters?*
3. Early testing
4. Defect clustering
5. Pesticide paradox
6. Testing is context dependent
7. Absence-of-errors fallacy

Viewpoints in testing



Basic concepts



- SUT: system under test
- Test case
  - a set of test inputs, execution conditions, and expected results developed for a particular objective
- Test suite
- Test oracle
  - A principle or mechanism that helps you decide whether the program passed the test
- Verdict: result (pass / fail / error / inconclusive…)

Test oracles

In practice: incomplete, partial, fallible

Specified	Implicit	Derived	…
Human	Crash	Previous version	
Model	Exception	Other program	
..	…	…	

Source: The Oracle Problem in Software Testing: A Survey

See also: Test Oracles; Oracle Problem

Problems and tasks

- Test selection
  - What test inputs and test data to use?
- Oracle problem
  - How to get/create reliable oracle?
- Exit criteria
  - How long to test?
- Testability
  - Observability + controllability