# Test oracles

**Zoltan Micskei**

**Budapest University of Technology and Economics**
**Fault Tolerant Systems Research Group**

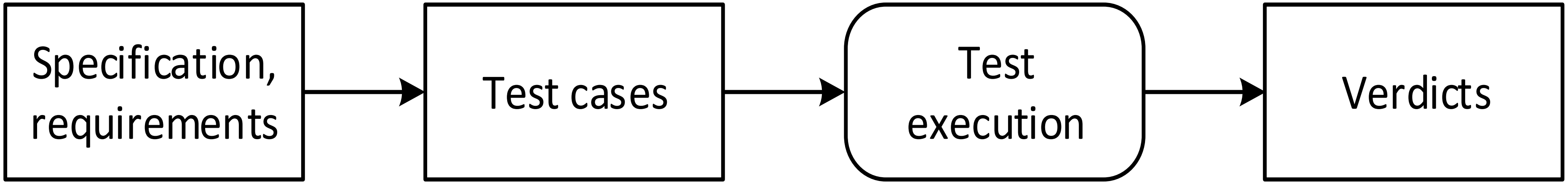
# Main topics of the course

- Overview (1.5)
  - Introduction, V&V techniques
- Static techniques (1.5)
  - Specification, Verifying source code
- Dynamic techniques: Testing (7)
  - Testing overview, **Test design techniques**
  - Test generation, Automation
- System-level verification (3)
  - Verifying architecture, Dependability analysis
  - Runtime verification

# Learning outcomes

- Recall the definition of test oracle (K1)
- Explain different types of test oracles (K2)

# Recap: Basic concepts



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

# EXERCISE: Expected result

1. The tax for a vehicle is based on the volume of the engine. Under 1000 $cm^3$ the tax is 0.1 EUR / $cm^3$, otherwise 0.2 EUR / $cm^3$. Hybrid cars qualify for a 10% discount.
   - Selected input #1: volume = 800 $cm^3$, hybrid = no
   - Expected result: ?
2. The software analyzes data from a particle detector searching for the mass of a new, yet unconfirmed particle.
   - Selected input: ?
   - Expected output: ???

# Challenges with test oracles #1

**“Non-testable programs”**

“Programs which were written in order to determine the answer in the first place. There would be no need to write such programs, if the correct answer were known.”

Source: E. J. Weyuker, “On testing non-testable programs,” Comput. J., vol. 25, no. 4, pp. 465–470, Nov. 1982.

# Challenges with test oracles #2

- The software can pass checking all expected results in the tests, but still be incorrect, e.g.
  - Returns the correct value, but way too slow, does not use authentication, uses hard to read UI…
  - The specification used as oracle also misses important special conditions
- The test fails, but the software behaved correctly
  - E.g. some rare external condition was not considered

**In practice: incomplete, partial, fallible**

*See also*: Test Oracles; Oracle Problem

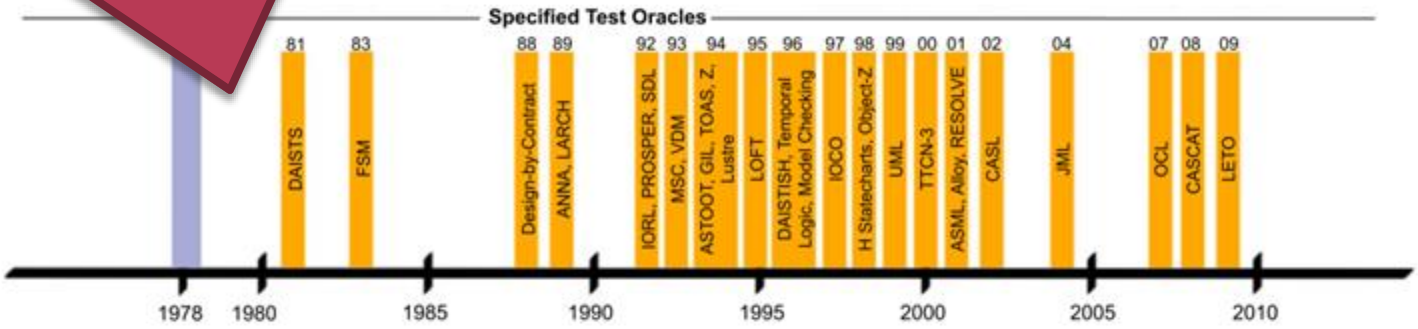
# Types of test oracles

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

*Source*: The Oracle Problem in Software Testing: A Survey

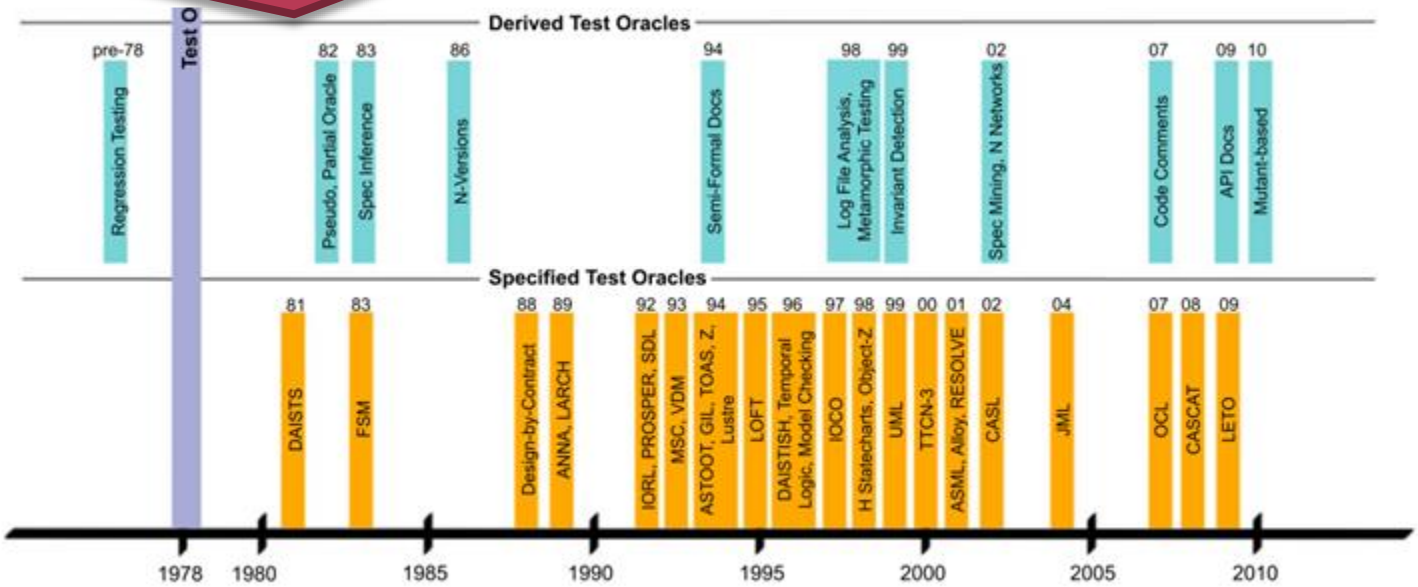
# Specified oracles

- Humans
- Textual specification
- Models (FSM, UML…)
  - Pre/post-conditions (Design-by-Contract)
  - -> see MBT lecture

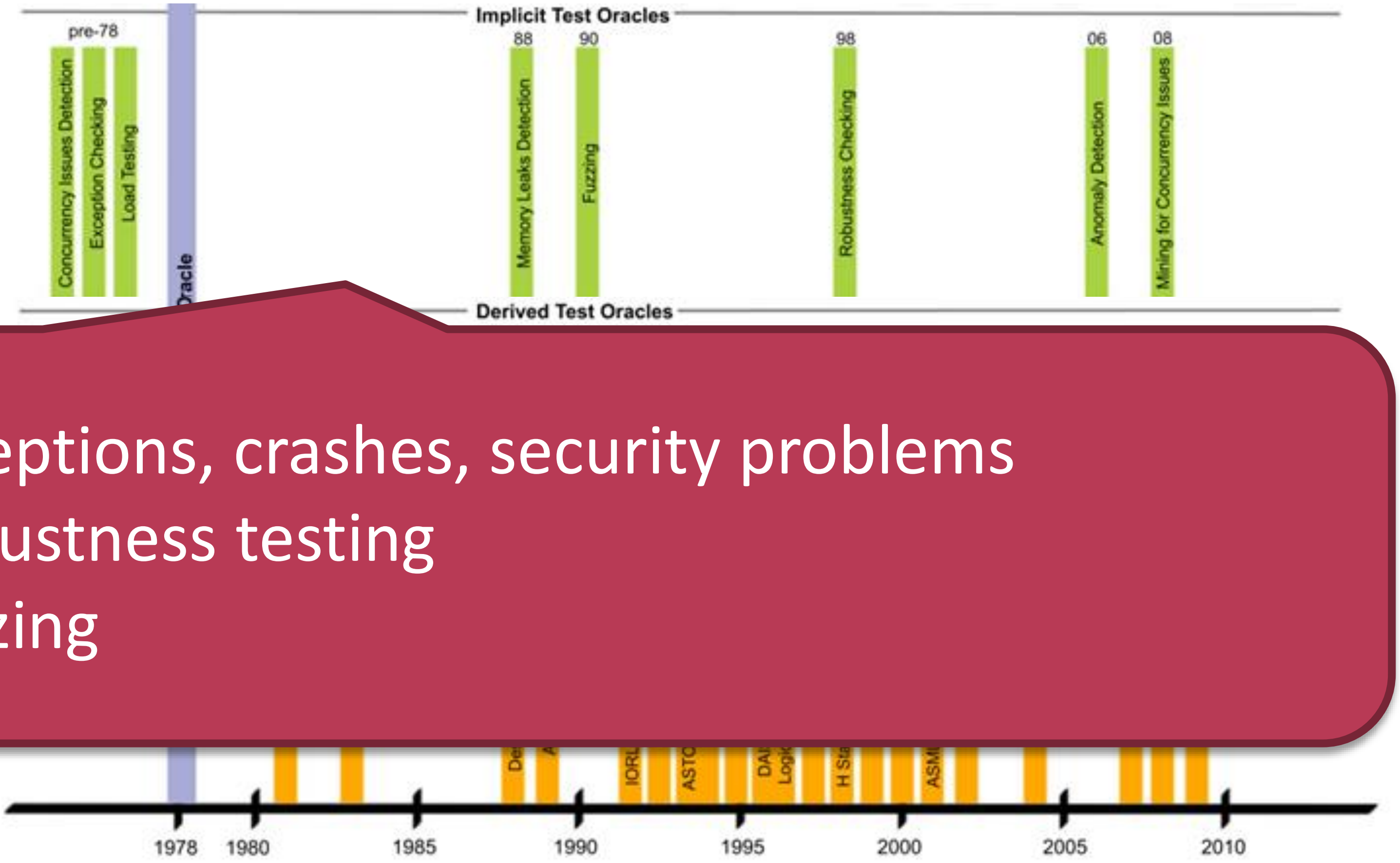


# Derived oracles

- Previous program versions
  - Regression testing
  - Only difference can be detected (not pass/fail)!
- Different implementations (N-version programming)
- Assumptions, validity checks, invertible function



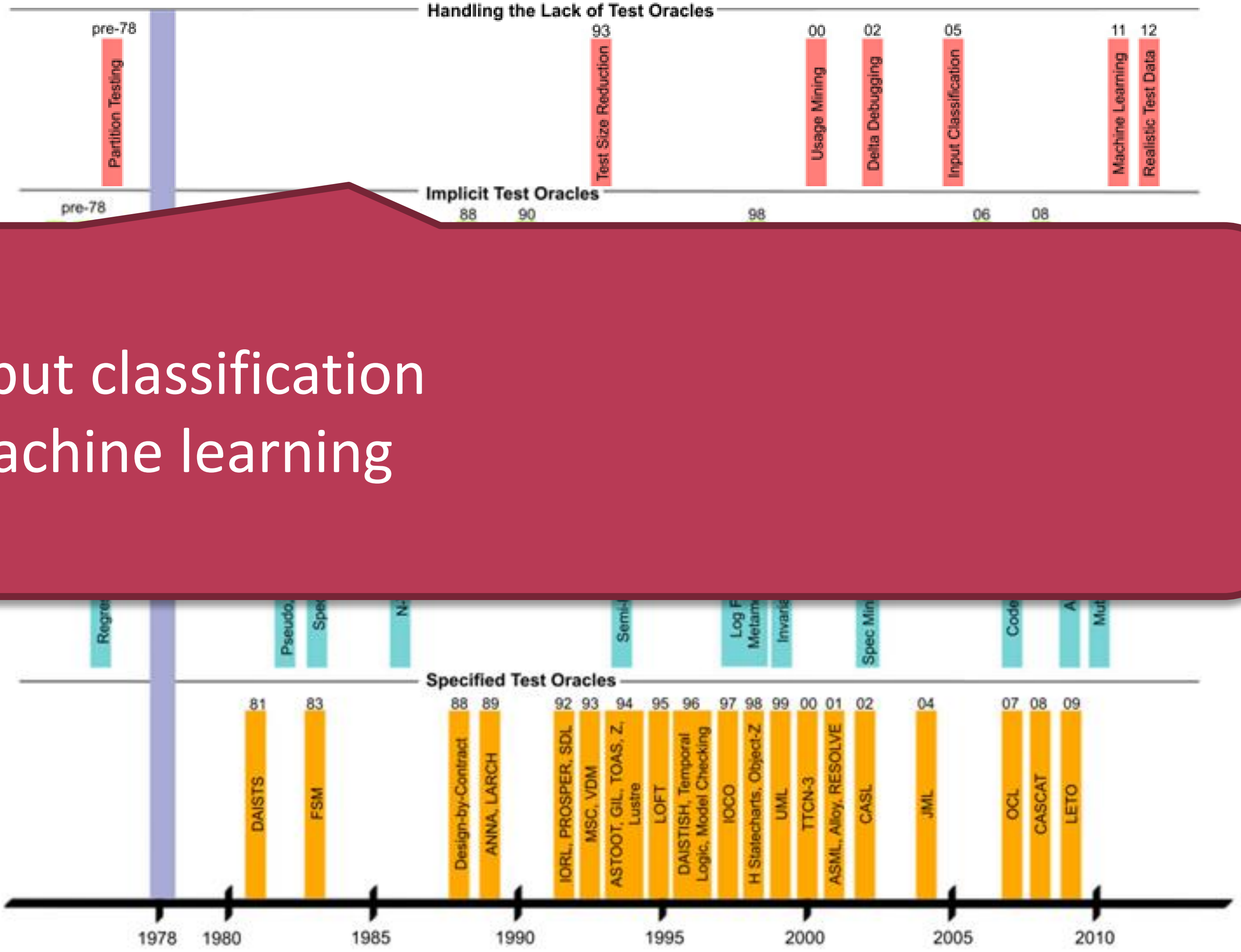
# Implicit oracles



- Exceptions, crashes, security problems
- Robustness testing
- Fuzzing

# Lack of oracles

BARR ET AL.: THE ORACLE PROBLEM IN SOFTWARE TESTING: A SURVEY



- Input classification
- Machine learning