Software Engineering

Software Testing, Part II

Srinivas Pinisetty [1]

05 March 2024

[1] Based on material from Wolfgang Aherndt...

Overview of this Lecture

- Focus on Unit Testing
- Terminology: Test case, test set, test suit, oracle
- Introduction to JUnit: a framework for rapid unit testing
- Extreme Testing using JUnit

Examples: System, Unit and Integration Testing

Pentium Bug (-94)

- Wrong result on floating point divisions.
- Missing entries in the lookup table.
- Rarely happened (on system level).
- Easy catch in unit test.

Examples: System, Unit and Integration Testing

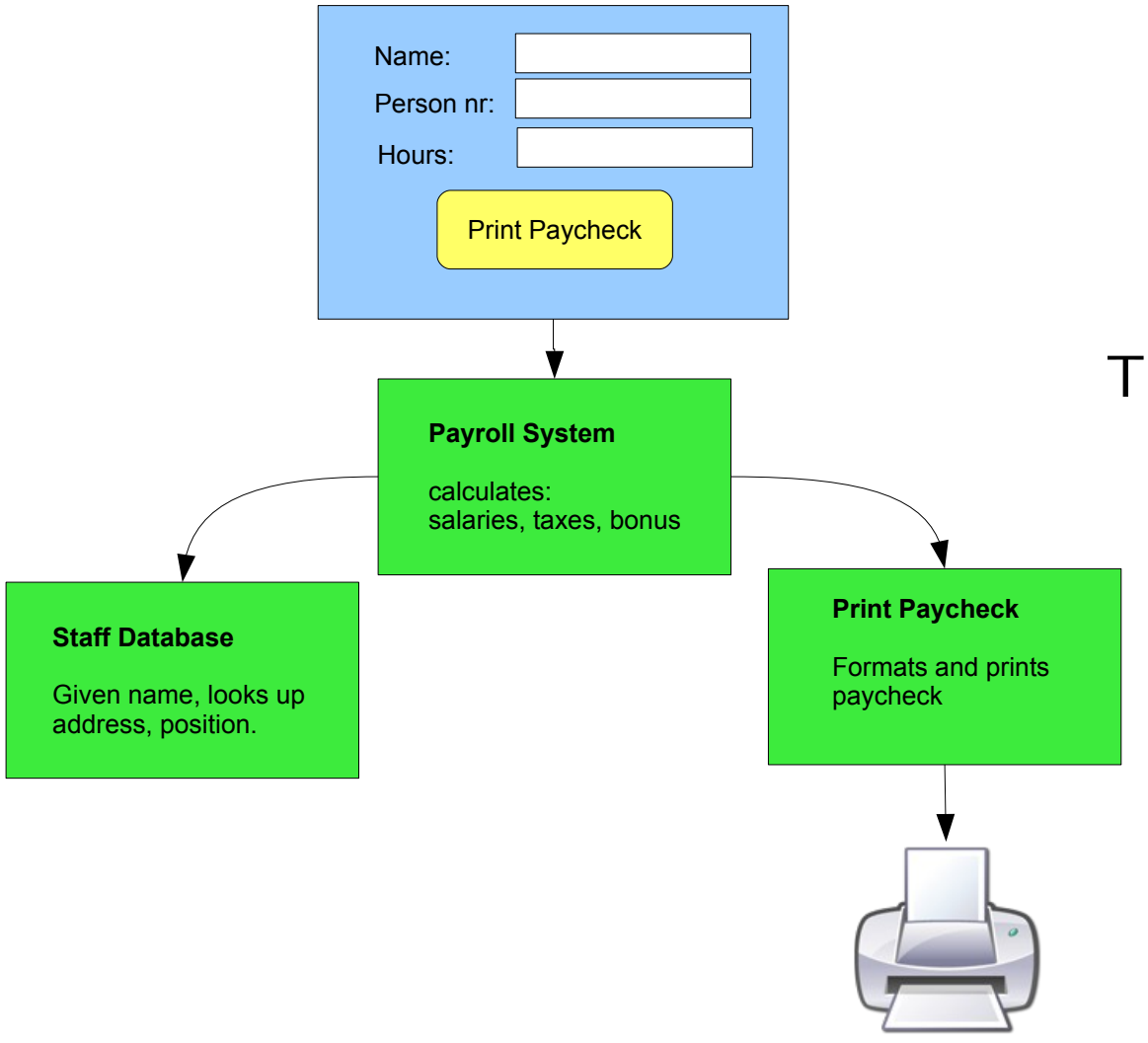
Pentium Bug (-94)

- Wrong result on floating point divisions.
- Missing entries in the lookup table.
- Rarely happened (on system level).
- Easy catch in unit test.

Ariane 5 Rocket (-96)

- Exploded 5 secs. after takeoff.
- Used guidance system from Ariane 4.
- Flight trajectory was different. Lacked system testing.

Discussion: Testing Levels of a System for Printing Paychecks



Think of examples of:

- System Tests
- Integration Tests
- Unit Tests

Some examples of Tests

- System Tests
  - Enter data in GUI, does it print the correct paycheck, formatted as expected?

Some examples of Tests

- System Tests
  - Enter data in GUI, does it print the correct paycheck, formatted as expected?
- Integration Tests
  - Payroll asks database for staff data, are values what's expected? Maybe there are special characters (unexpected!).

Some examples of Tests

- System Tests
  - Enter data in GUI, does it print the correct paycheck, formatted as expected?
- Integration Tests
  - Payroll asks database for staff data, are values what's expected? Maybe there are special characters (unexpected!).
- Unit Tests
  - Does payroll system compute correct tax-rate, bonus etc?
  - Does the Print Paycheck button react when clicked?
  - ...

Regression Testing

Orthogonal to the above testing levels:

Regression Testing

- Testing that is done after changes in the software.
- Purpose:
  gain confidence that the change(s) did not cause (new) failures.
- Standard part of the maintenance phase of software development.

E.g. Suppose Payroll subsystem is updated. Need to re-run tests.

Unit Testing

Rest of testing part of the course: focusing largely on unit testing

recall: unit testing = procedure testing = (in oo) method testing

major issues in unit testing:

1. unit test cases ('test cases' in short)

Test Cases

The science of testing is largely the science of test cases.

What does a test case consists of?

Test case

- Initialisation (of class instance and input arguments)
- Call to the method under test.
- Decision (oracle) whether the test succeeds or fails

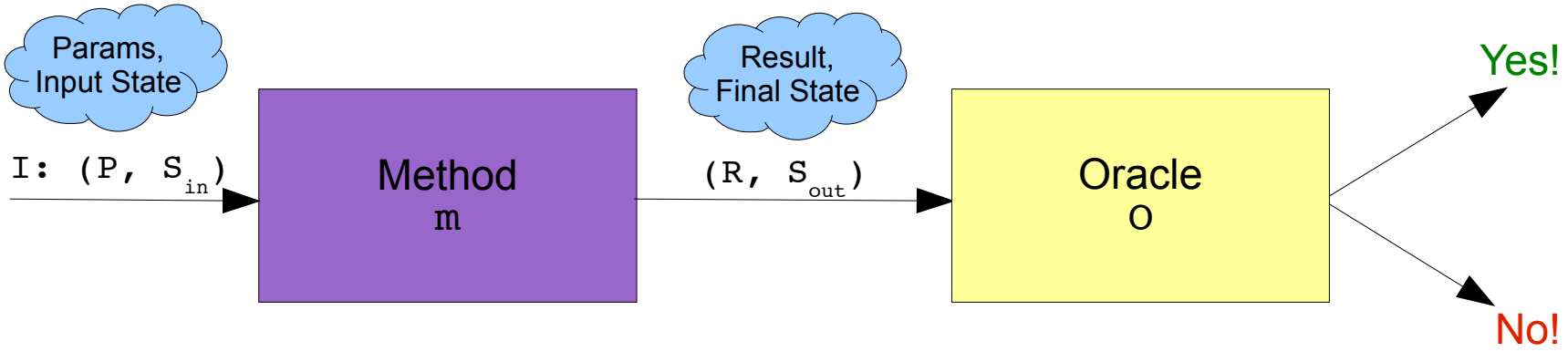
- two first parts seem enough for a test case,
- but test oracle is vital for *automated* evaluation of test

‘Success’ vs. ‘Failure’ of Tests

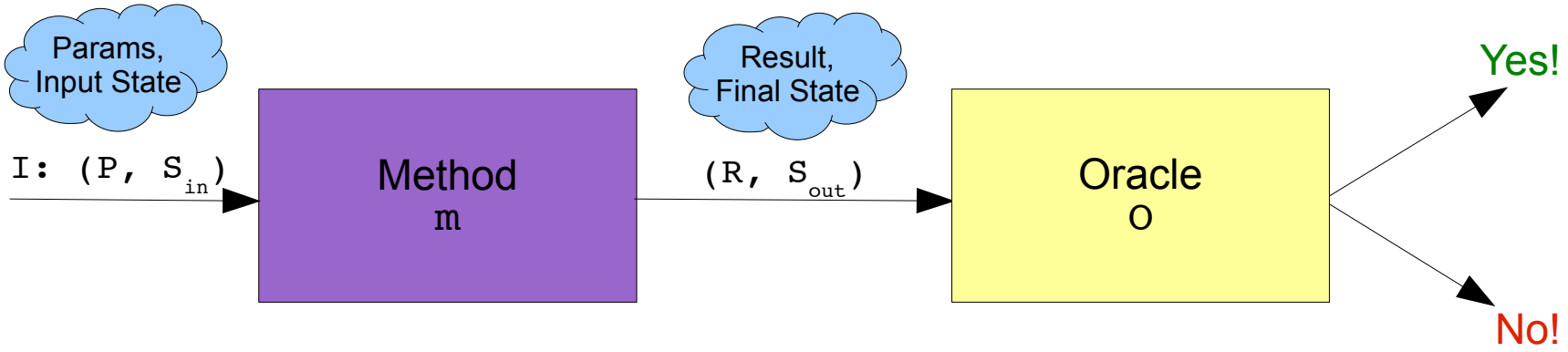
What does it mean for a test to succed?

... or fail?

Test Cases, more precise



Test Cases, more precise



More formally...

A test case is a tuple $\langle \mathtt{m}, \mathtt{I}, \mathtt{O} \rangle$ of method $\mathtt{m}$, input $\mathtt{I}$, and oracle $\mathtt{O}$, where

- $\mathtt{m}$ is the method under test
- $\mathtt{I}$ is a tuple $\langle \mathtt{P}, \mathtt{S_{in}} \rangle$
  of call parameters $\mathtt{P}$ and initial state $\mathtt{S_{in}}$
- $\mathtt{O}(\mathtt{R}, \mathtt{S_{out}}) \mapsto \{\mathit{pass}, \mathit{fail}\}$
  is a function on return value R and final state $\mathtt{S_{out}}$, telling whether they comply with correct behaviour

Test Set

A test set $TS^{\mathtt{m}}$ for a (Java) method $\mathtt{m}$ consists of n test cases:

$$TS^{m} = \{\langle \mathtt{m}, \mathtt{I_1}, \mathtt{O_1} \rangle, \ldots, \langle \mathtt{m}, \mathtt{I_n}, \mathtt{O_n} \rangle\}$$

In general, $\mathtt{O_i}$ is specific for each test case!

Test Suite

A test suite for methods $\mathtt{m}_1, \ldots, \mathtt{m}_k$ is a union of corresponding test sets:

$$TS^{m_1} \cup \ldots \cup TS^{\mathtt{m}_k}$$

Automated and Repeatable Testing

Basic idea: write code that performs the tests.

- By using a tool you can automatically run a large collection of tests
- The testing code can be integrated into the actual code, thus stored in an organised way
- side-effect: documentation
- After debugging, the tests are rerun to check if failure is gone
- Whenever code is extended, all old test cases can be rerun to check that nothing is broken (regression testing)

Automated and Repeatable Testing (cont'd)

We will use JUnit for writing and running the test cases.

JUnit: small tool offering

- some functionality repeatedly needed when writing test cases
- a way to annotate methods as being test cases
- a way to run and evaluate test cases automatically in a batch

JUnit

- JAVA testing framework to write and run automated tests
- JUnit features include:
  - Assertions for testing expected results
  - Annotations to designate test cases
  - Sharing of common test data
  - Graphical and textual test runners
- JUnit is widely used in industry
- JUnit used from command line or within an IDE (e.g., Eclipse)

(Demo)