

# Tutorial 1

**Swinburne University of Technology**

Software Testing and Reliability (SWE30009)

Semester 2, 2023

Lecturer: Prof Tsong Yueh Chen

Tutor: Dr Hung Q Luu



Part A

Tutorial task

# Testing objectives

Consider the following program:

Input A, B // A and B are integer variables

$C = (A - B) * B$

Output C

Suppose the following testing objectives are applied:

1. One and only one incorrect arithmetic operator
2. Incorrect use of arithmetic operators

# Task

## **Question 1:**

What are the constraints for test cases in order to achieve testing objectives (1) and (2)?

## **Question 2:**

How do you construct your concrete test cases from the constraints of test cases for testing objectives (1) and (2)?



Part B

Revisiting the Lectures

# Why testing

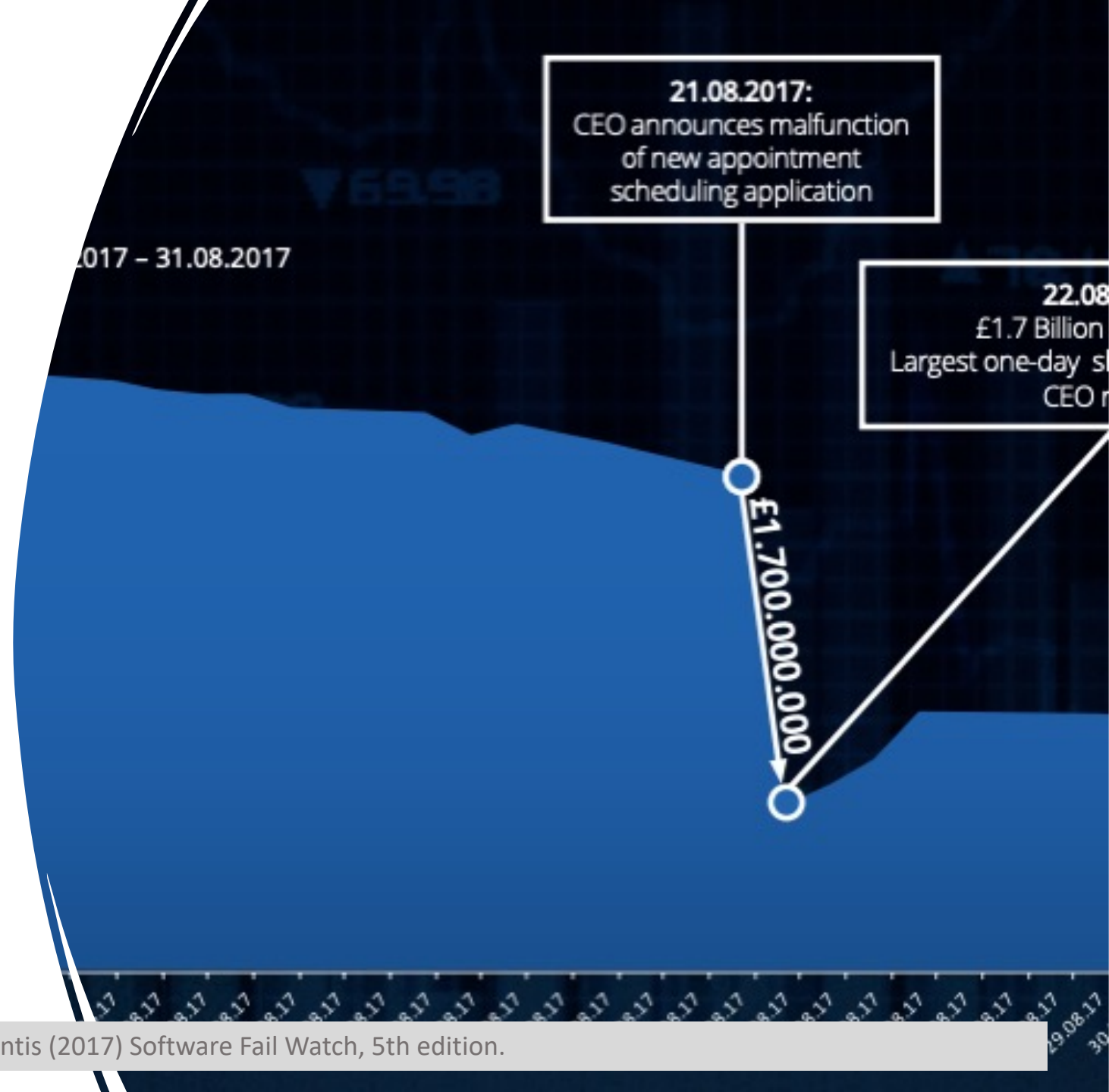
---

- Ariane 5
  - On June 4 1996, the European Space Agency launched Ariane 5 rocket.
  - A bug in software (Ada language) caused the rocket to self-destruct 37s after blast-off.



# Why testing

- Provident Financial lost **£1.7 billion in a single day** and **company £120 million** in profit of the year (2017) – Tricentis (2017).
- Glitch: Malfunction of a new appointment scheduling application.





# Why testing

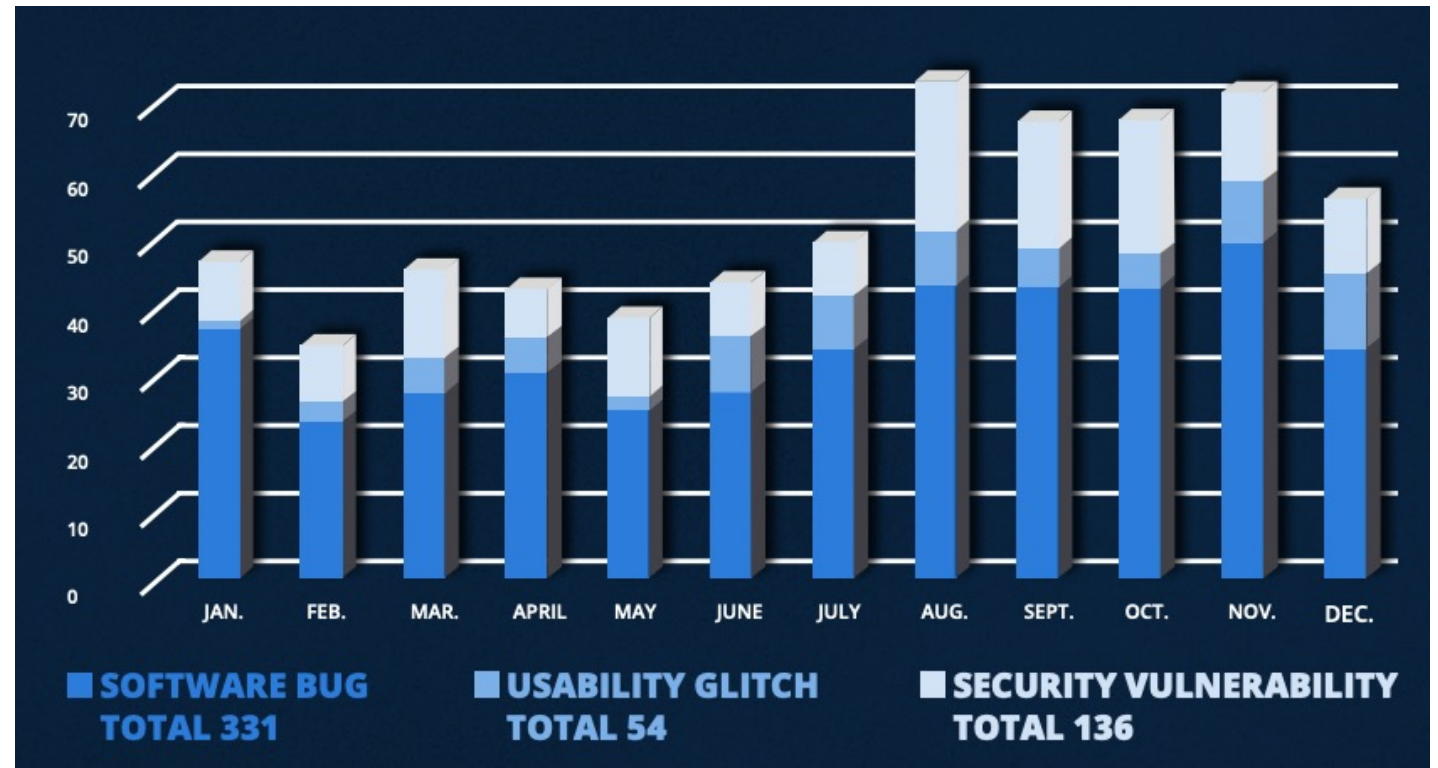
**1,71 trillion dollar**

Tricentis software testing company analyzed 606 software failure from 314 corporations. Results show they affected **3.6 billion people**

## LOSSES FROM SOFTWARE FAILURES (USD)

# 1,715,430,778,504

ONETRILLIONSEVENHUNDREDFIFTEENBILLIONFOURHUNDREDTHIRTYMILLIONSEVENHUNDREDSEVENTY-EIGHTTHOUSANDFIVEHUNDREDFOUR





# Why testing

---

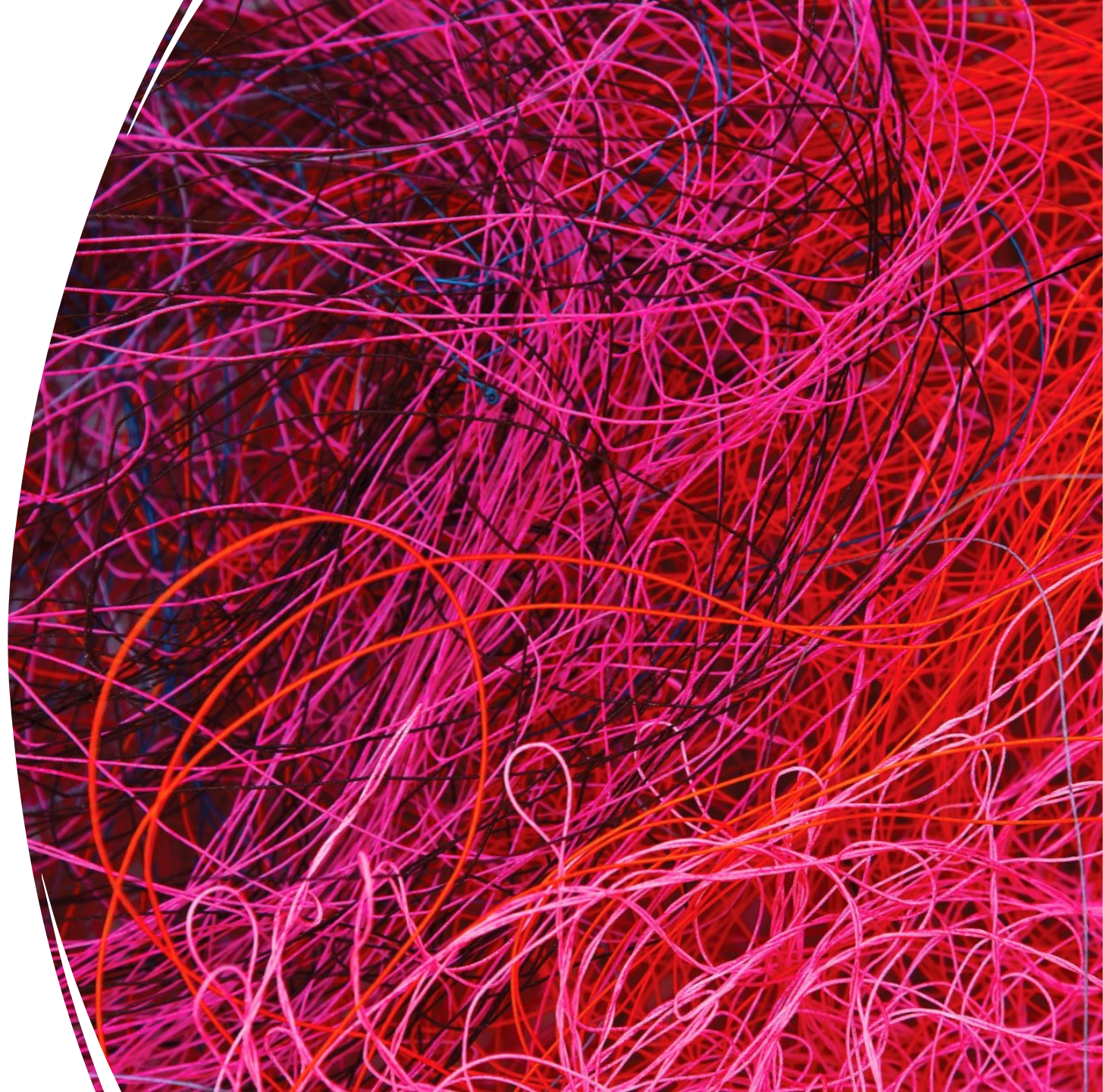
## Cost for debugging



# Discussion

---

- Error
- Fault
- Failure





# Definitions

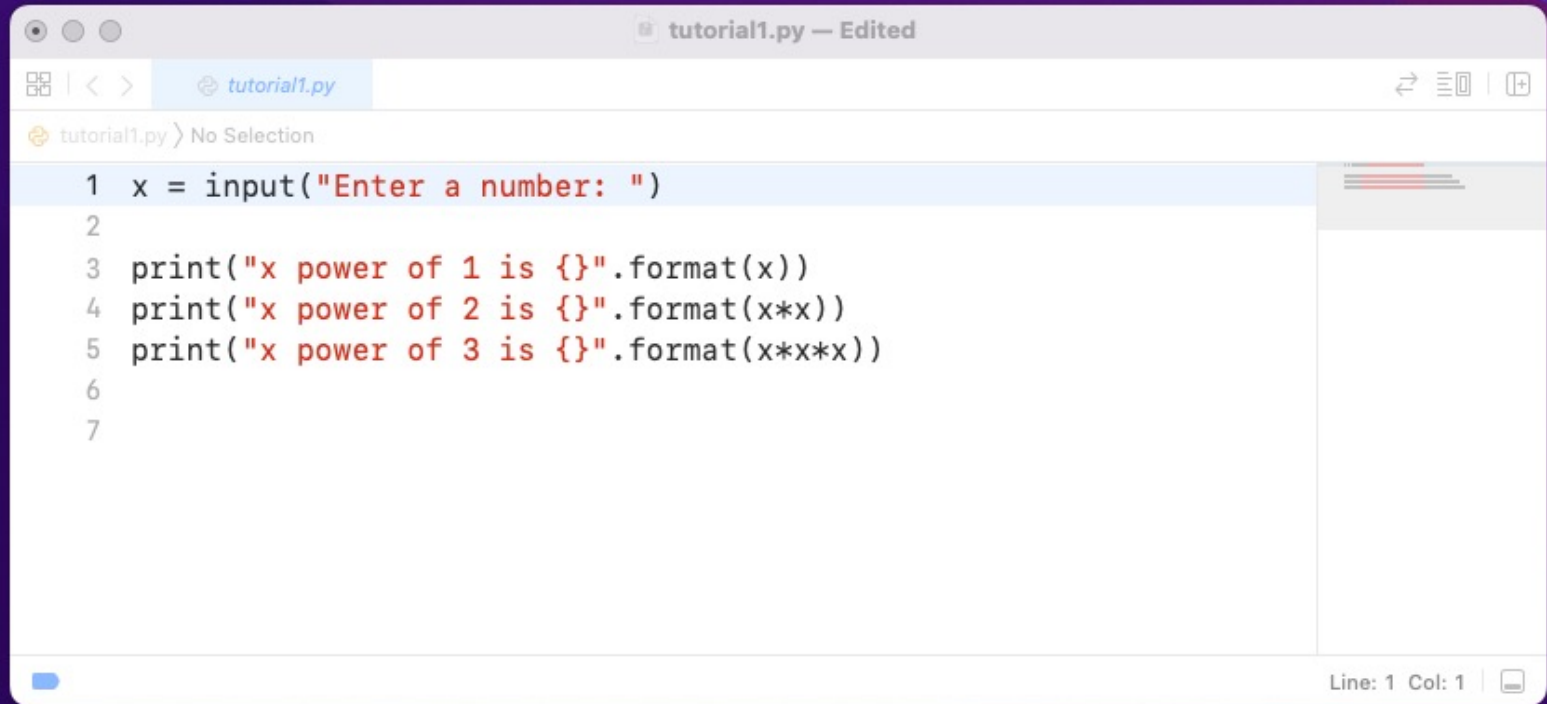
---

- Error
  - A mistake made by the programmer
- Fault
  - An incorrect definition or process in the program
- Failure
  - An observable violation against the specifications



# Example

- Error
- Fault
- Failure



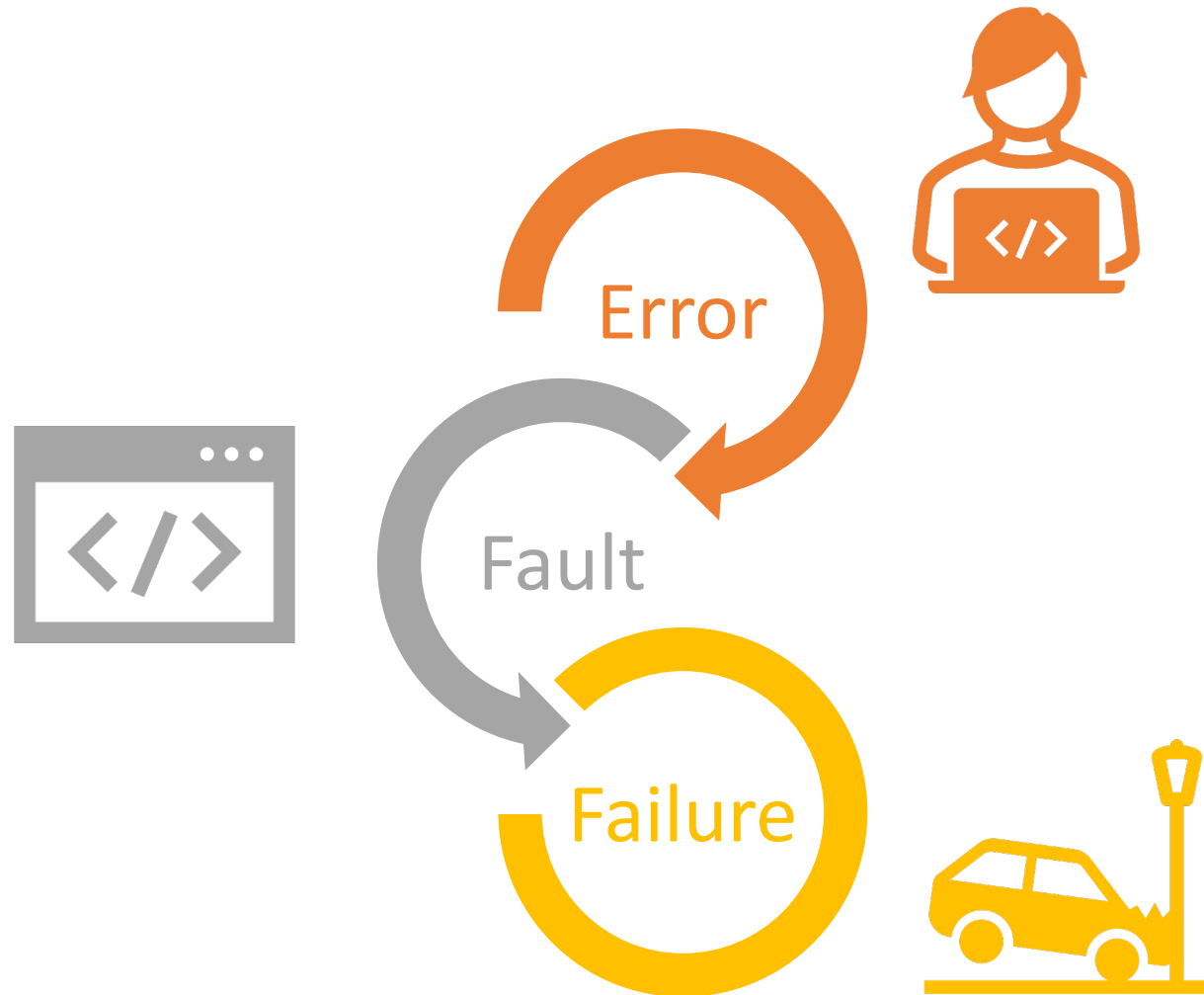
```
1 x = input("Enter a number: ")
2
3 print("x power of 1 is {}".format(x))
4 print("x power of 2 is {}".format(x*x))
5 print("x power of 3 is {}".format(x*x*x))
6
7
```

The screenshot shows a code editor window with the title 'tutorial1.py — Edited'. The code is written in Python and is as follows:

```
1 x = input("Enter a number: ")
2
3 print("x power of 1 is {}".format(x))
4 print("x power of 2 is {}".format(x*x))
5 print("x power of 3 is {}".format(x*x*x))
6
7
```

The code prompts the user to enter a number and then prints the number raised to the powers of 1, 2, and 3. The status bar at the bottom right indicates 'Line: 1 Col: 1'.

# Relations





# Discussion

---

- Test oracle

ORACLE?





# Test oracle

---

- A mechanism or procedure to check whether the output for any input is correct or not





# Example

- Test oracle

Program P

$$x^{100} - 2x^{10} + 1 = 0$$



# Test oracle

---

- Backward substitution and evaluation
- Commonly based on specifications and documentation



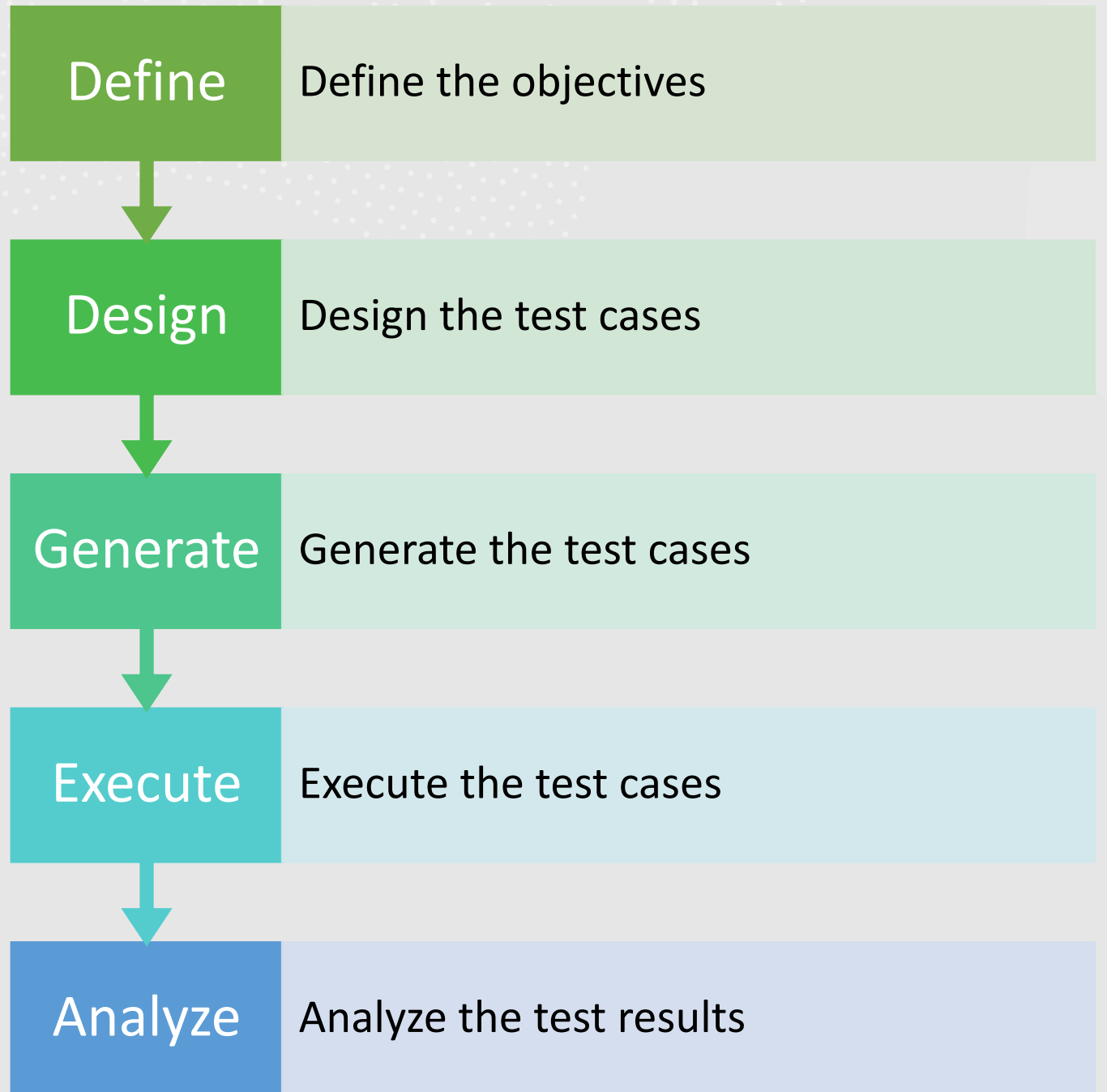


# V&V

---

- Verification
  - Are we building the product **right**?
- Validation
  - Are we building the right **product**?

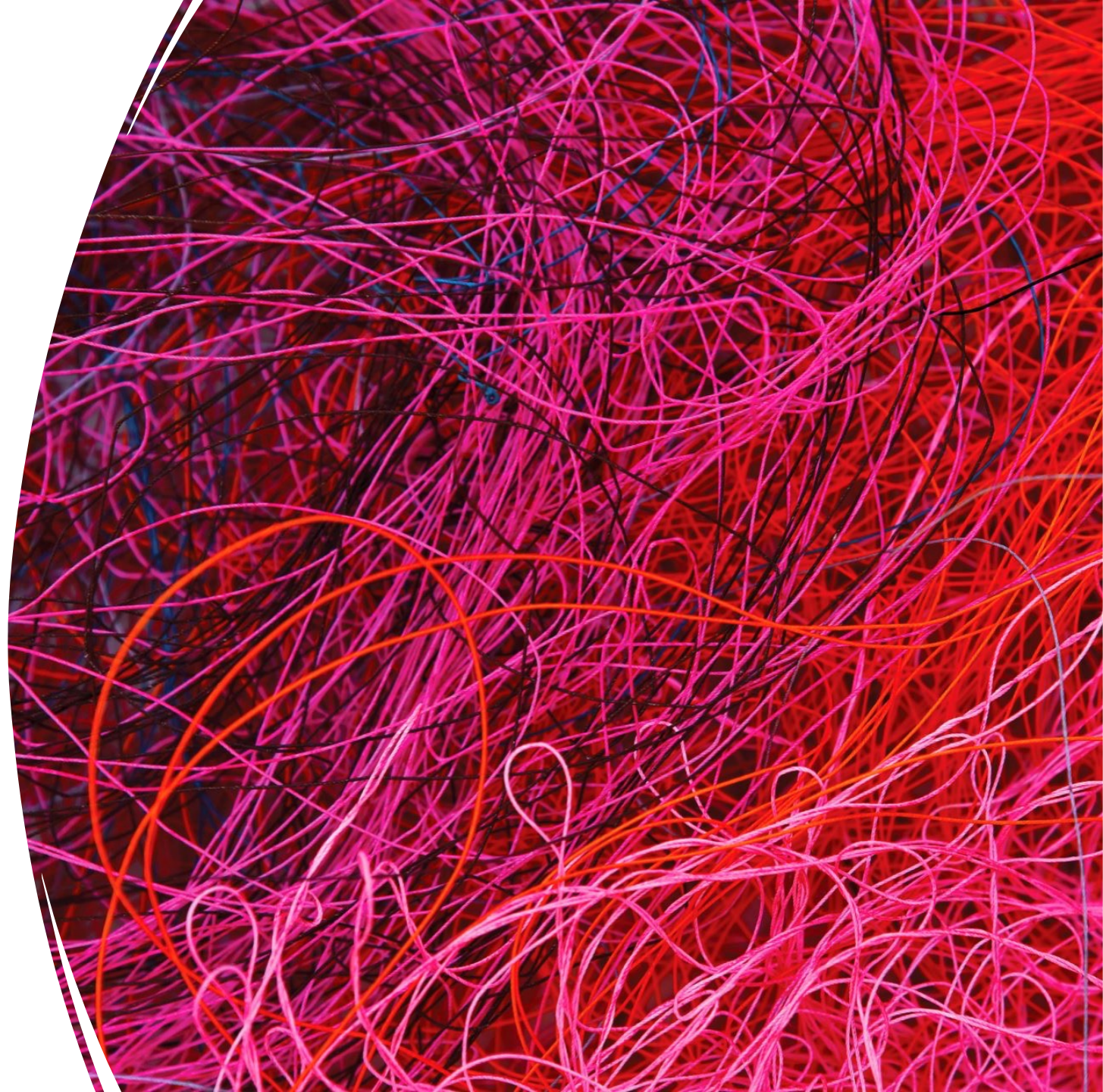
# Testing activities





# Discussion

- Test objectives





# Test objectives

- The aim or purpose of testing



# Example

- Incorrect arithmetic operator
- Incorrect variable definition
- Incorrect data type assignment
- ...

Suppose that we are asked to test the following program

Input A, B // A and B are integer variables

C = A - B

Output C

$A - B \neq$

- $A + B$
- $A * B$
- $A / B$



# Discussion: Which test case is better?

- Test case 1:  $A = 4, B = 2$
- Test case 2:  $A = 3, B = 1$
- Test case 3:  $A = -1, B = 1$
- Test case 4:  $A = 1, B = 1$

$$A - B \neq$$

- $A + B$

- $A * B$

- $A / B$

# Discussion: Which test case is better?

- Test case 1:  $A = 4, B = 2$
- Test case 2:  $A = 3, B = 1$
- Test case 3:  $A = -1, B = 1$
- Test case 4:  $A = 1, B = 1$

Operator error

$$C = A^2 - B^2$$

# Testing objectives

- Frankly speaking, we seek to create one or multiple test cases that can help distinguish “good” program (correctly coded) with “bad” programs (buggy or incorrectly implemented).
- Solution
  - Easy: Trial and error
  - Medium: Automated search
  - Hard: Constraint solving