

# Software Bugs

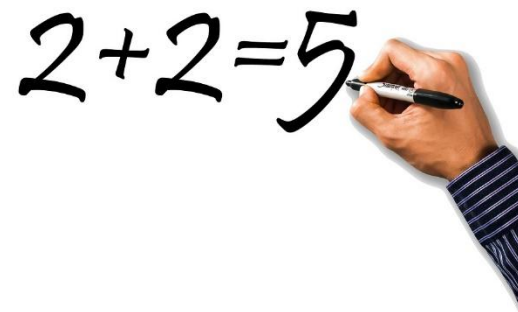
States of existence

# States of bug existence

- Error
- Fault
- Failure

# What is a error?

People make errors. A good synonym is mistake. When people make mistakes while coding, we call these mistakes bugs. Errors tend to propagate; a requirements error may be magnified during design and amplified still more during coding.



# What is a defect(fault)?

A fault is the result of an error. It is more precise to say that fault is the representation of an error, where representation is the mode of expression.

# Omission faults

Faults can be elusive. An error of omission results in a fault in which something is missing that should be present in the representation. This suggests a useful refinement; we might speak of faults of commission and faults of omission.

# Commission faults

A fault of commission occurs when we enter something into a representation(source code) that is incorrect.

# Fault(defect) example

An accounting program had an incorrect instruction (fault) in the formatting routine that inserts commas in large numbers such as “\$4,500,000”. Any time a user prints a number greater than six digits, a comma may be missing (a failure).



# What is a failure

A failure occurs when the code corresponding to a fault executes.



Your PC ran into a problem and needs to restart. We're just collecting some error info, and then we'll restart for you. (0% complete)

If you'd like to know more, you can search online later for this error: HAL\_INITIALIZATION\_FAILED



# How many failures result from a single fault?

That depends on:

- The location of the fault
- How long the remains before it is removed
- How many people are using the software

The comma-insertion fault led to millions of failures because it was in a frequently used piece of code, in software that has thousands of users, and it remained unresolved for more than a year.

# Software testing goal

A goal of testing is to execute the program with inputs that cause all faults to reveal themselves as failures. What does this require?

Consider this program:

```
public int square(int x) {  
    int square = x + x; //should be x*x  
    System.out.println("Squared x:" + square);  
    return square;  
}
```

# Ideal fault conditions

- Reachability condition
- Necessity condition
- Propagation condition

These three conditions are called ideal fault conditions.

# Reachability condition

To provoke a failure, the program's inputs must cause the faulty statement to be executed. Those inputs satisfy a reachability condition. In this case simply invoking the method will do the job.

# Necessity condition

Next, the faulty statement must produce a different result than the correct statement. This is called necessity condition. It describes the inputs that cause an incorrect internal state. In this example if we pass  $x = 2$ , program will return correct value 4.

The necessity condition for this fault is that  $x \neq 2$  and  $x \neq 0$ .

# Propagation condition

Finally, the incorrect internal state must propagate so that it becomes visible in the program's results.

In this case print method is taking care for this. This is not going to happen very often by itself. It is our responsibility to make sure propagation happens for real.

# Software Testing History

When and how everything started?

# First Bug Report

In 1947, Grace Murray Hopper was working on the Harvard University Mark II Aiken Relay Calculator (a primitive computer). On the 9th of September, 1947, when the machine was experiencing problems, an investigation showed that there was a moth trapped between the points of Relay #70, in Panel F. The operators removed the moth and affixed it to the log. The entry reads: "First actual case of bug being found."





# First Bug Report

Picture of the bug report created back in time.

9/9

0800 Andam started  
 1000 " stopped - andam ✓


1300 (032) MP - MC ~~1.582147000~~  
 (033) PRO 2 2.130476415 (2) 4.615925059(-2)  
 correct 2.130676415

Relays 6-2 in 033 failed special speed test  
 in relay " 10.00 test.

Relays changed

1100 Started Cosine Tape (Sine check)  
 1525 Started Multi Adder Test.

1545



Relay #70 Panel F  
 (moth) in relay.

First actual case of bug being found.

1630 Andam started.

Relay 2142  
 Relay 3370

# Bugs in real life

Systems in trouble

# The Pentium chip's math error (1993)

Thanks to a programming error, Intel's famous Pentium chip turned out to be pretty bad at math. The actual mistakes it made were fairly minute (beyond the eighth decimal point) and limited to certain kinds of division problems. But the irony—oh, the irony!—of a computer chip that made math errors made the problem blow up into the mother of all public relations disasters. After playing down the severity of the problem, causing even more public backlash, the company finally agreed to provide anyone who asked with a fixed chip.

# AT&T hangs up its long-distance service (1990)

[www.pragmatic.bg](http://www.pragmatic.bg)

For nine hours in January 1990 no AT&T customer could make a long-distance call. The problem was the software that controlled the company's long-distance relay switches—software that had just been updated. AT&T wound up losing \$60 million in charges that day—a very expensive bug.

# The Mars Climate Orbiter in space (1998)

NASA's \$655-million robotic space probe plowed into Mars's upper atmosphere at the wrong angle, burning up in the process. The problem? In the software that ran the ground computers the thrusters' output was calculated in the wrong units (pound-seconds instead of newton-seconds, as the NASA-Lockheed contract had specified). Fortunately software programs for subsequent missions to Mars have gotten the measurements right.

# Windows locks out non software pirates

For 19 hours on August 24, 2007, anyone who tried to install Windows was told, by Microsoft's own antipiracy software (called Windows Genuine Advantage) that they were installing illegal copies. If you'd bought Windows Vista, you discovered certain features shut off as punishment. The bug this time was both human and traditional: Someone accidentally installed a buggy, early version of the Genuine Advantage software on Microsoft's servers.

# Apple Maps gives us directions to nowhere

In its rivalry with Google, Apple decided to get rid of the much-adored Google Maps app that had always come on new iPhones—and to replace it with a new map app that Apple had written itself. But in Apple's Maps whole lakes, train stations, bridges and tourist attractions were missing or mislabeled. The Washington Monument moved across the street. Riverside Hospital appeared in Jacksonville, Fla., even though it had become a Publix supermarket 11 years earlier. In the app's 3D view bridges and dams seemed to melt into the water and Auckland, New Zealand's main train station was in the middle of the ocean.

# Questions

