- Failure: A difference from the expected result. This is the problem you observe.
- Fault: The cause of the failure.
- Error: The mistake which caused the fault to occur. e.g, typos.

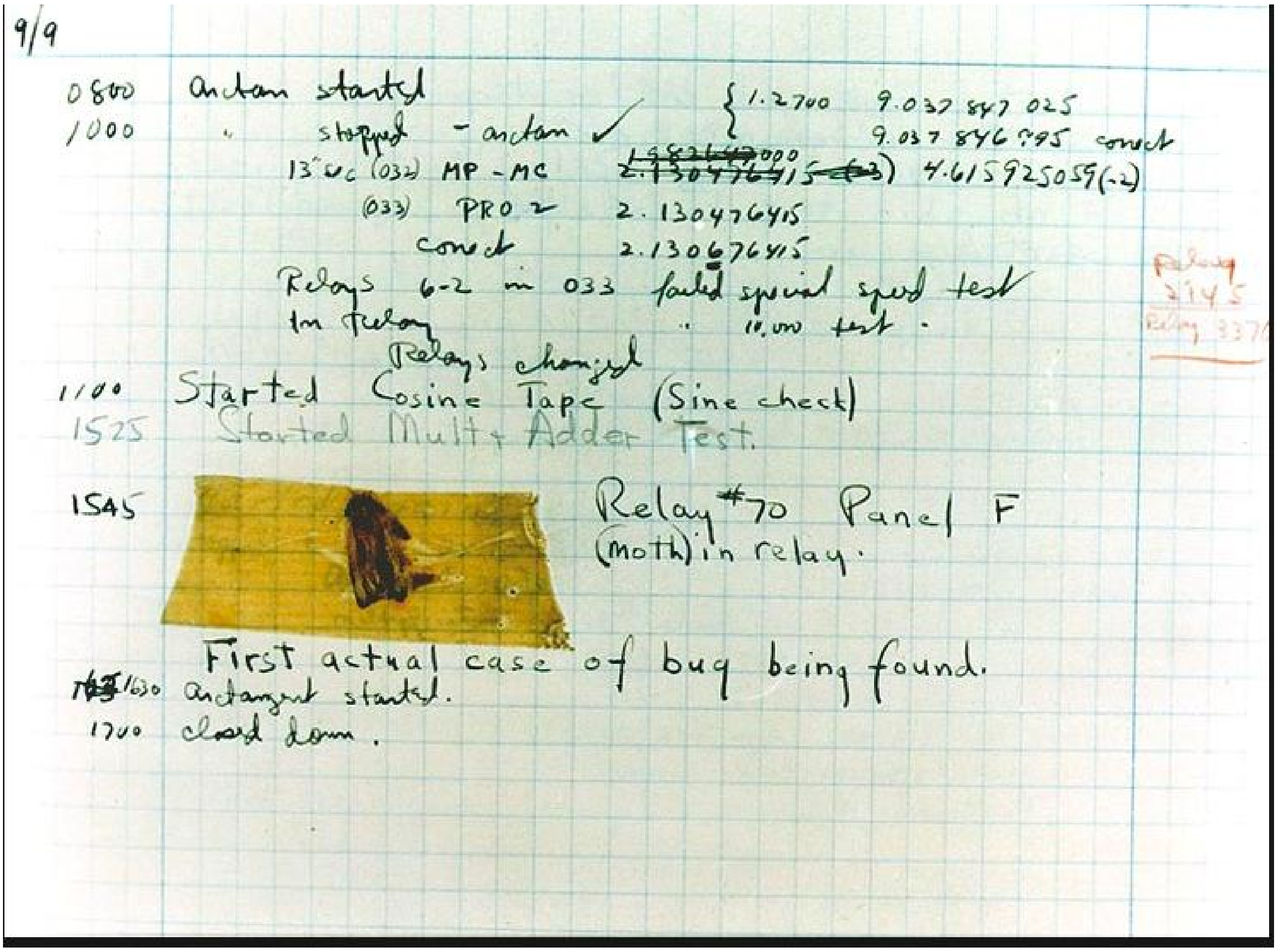
An example of failure, fault and error.

```
pre: param is an integer.
post: returns the product of the param multiplied by 2.

1. int double (int param) {
2.   int result;
3.   result = param * param;
4.   return result;
5. }
```

- A call to double(3) returns 9, but the post condition says it should return 6.
- Result 9 represents a **failure**.
- The failure is due to the **fault** at line 3, ( * is used instead of + )
- The **error** is a typo, ( someone typed * instead of + by mistake).

**Why give three different labels for a "Bug"?**



9/9

0800 Anctan started

1000 " stopped - anctan ✓ {1.2700 9.037 847 025
9.037 846 995 conect}

13" UC (032) MP - MC ~~1.982147000~~ ~~2.130476415~~ ~~(-3)~~ 4.615925059(-2)

(033) PRO 2 2.130476415

conect 2.130676415

Relays 6-2 in 033 failed special speed test
In relay " 11.000 test.
Relays changed

1100 Started Cosine Tape (Sine check)

1525 Started Mult + Adder Test.

1545 Relay #70 Panel F
(moth) in relay.

First actual case of bug being found.

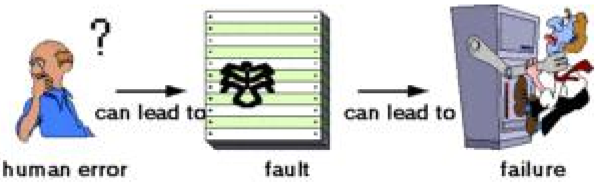
~~1630~~ 1630 Antangent started.

1700 closed down.

Relay 2145
Relay 3370

A page from the Harvard Mark IIelectromechanical computer's log, featuring a dead moth that was removed from the device

https://www.testingexcellence.com/error-fault-failure-software-testing/



## Error, Fault, Failure

What is the difference between **error**, **fault** and **failure** in software testing?

**Error –** a human action that produces an incorrect result. This is also sometimes referred to as **Mistake**.

**Fault –** a manifestation of an error in software, also known as **Defect** or **Bug**.

**Failure –** a deviation of the software from its expected delivery or service.

An error is something that a human does, we all make mistakes and when we do whilst developing software, it is known as an error. The result of an error being made is a fault. It is something that is wrong in the software (source code or documentation – specifications, manuals, etc.). Faults are also known as defects or bugs.

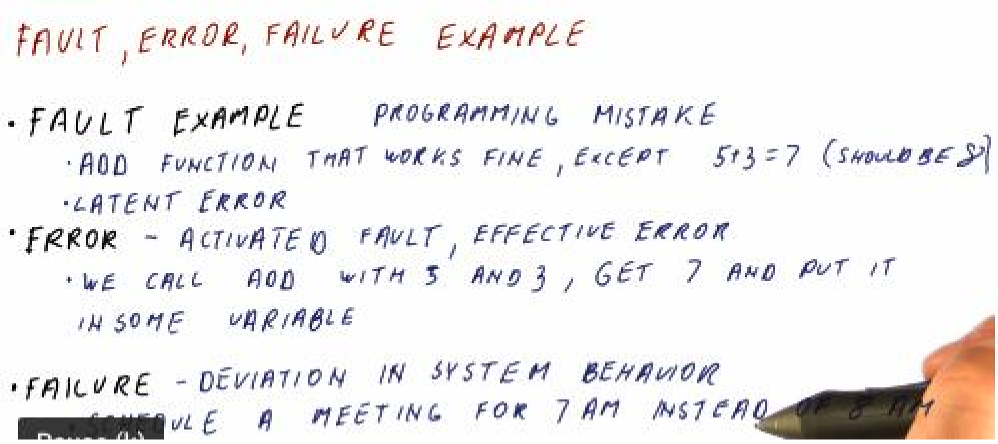
When a system or piece of software produces an incorrect result or does not perform the correct action, this is known as a failure. Failures are caused by faults in the software. Note that software system can contain faults but still never fail (this can occur if the faults are in those parts of the system that are never used). In other words, failure is the manifestation of one or more faults (bugs).

### Reliability

Another term that should be understood is reliability. A system is said to be reliable when it performs correctly for long periods of time. However, the same system used by two different people may appear reliable to one but not to the other. This is because the different people use the system in different ways.

**Reliability**: the probability that the software will not cause the failure of the system for a specified time under specified conditions.

The definition of reliability therefore includes the phrase ‘under specified conditions’. When reporting on the reliability of a system it is important to explain under what conditions the system will achieve the specified level of reliability. For example, a system may achieve a reliability of no more than one failure per month providing no more than 10 people use the system simultaneously



FAULT, ERROR, FAILURE EXAMPLE

- FAULT EXAMPLE PROGRAMMING MISTAKE
  - ADD FUNCTION THAT WORKS FINE, EXCEPT 5+3=7 (SHOULD BE 8)
  - LATENT ERROR
- ERROR - ACTIVATED FAULT, EFFECTIVE ERROR
  - WE CALL ADD WITH 5 AND 3, GET 7 AND PUT IT IN SOME VARIABLE
- FAILURE - DEVIATION IN SYSTEM BEHAVIOR
  - SCHEDULE A MEETING FOR 7 AM INSTEAD OF 8 AM