Software Testing

Manar Elkady

Some of the material are retrieved from a previous course offering by Dr.Soha Makady and Prof. Amr Kamel

Software Testing – Basic Definitions

- Software fault – often referred to as a bug:

A static defect in software (incorrect lines of code)

- Software error:

An incorrect internal state (unobserved)

- Software Failure:

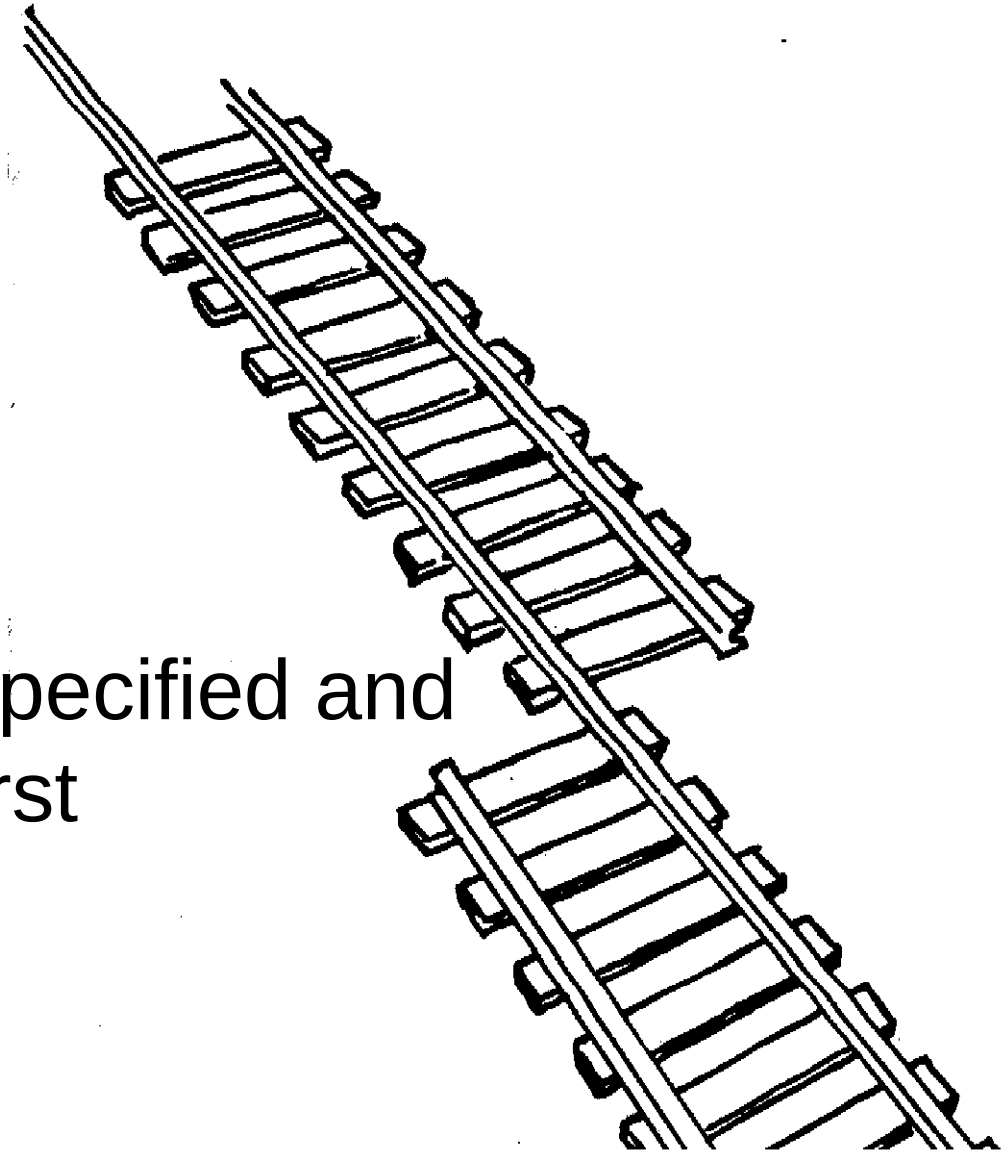
External, incorrect behavior with respect to the requirements or another description of the expected behavior

When do failures happen?

Any examples?

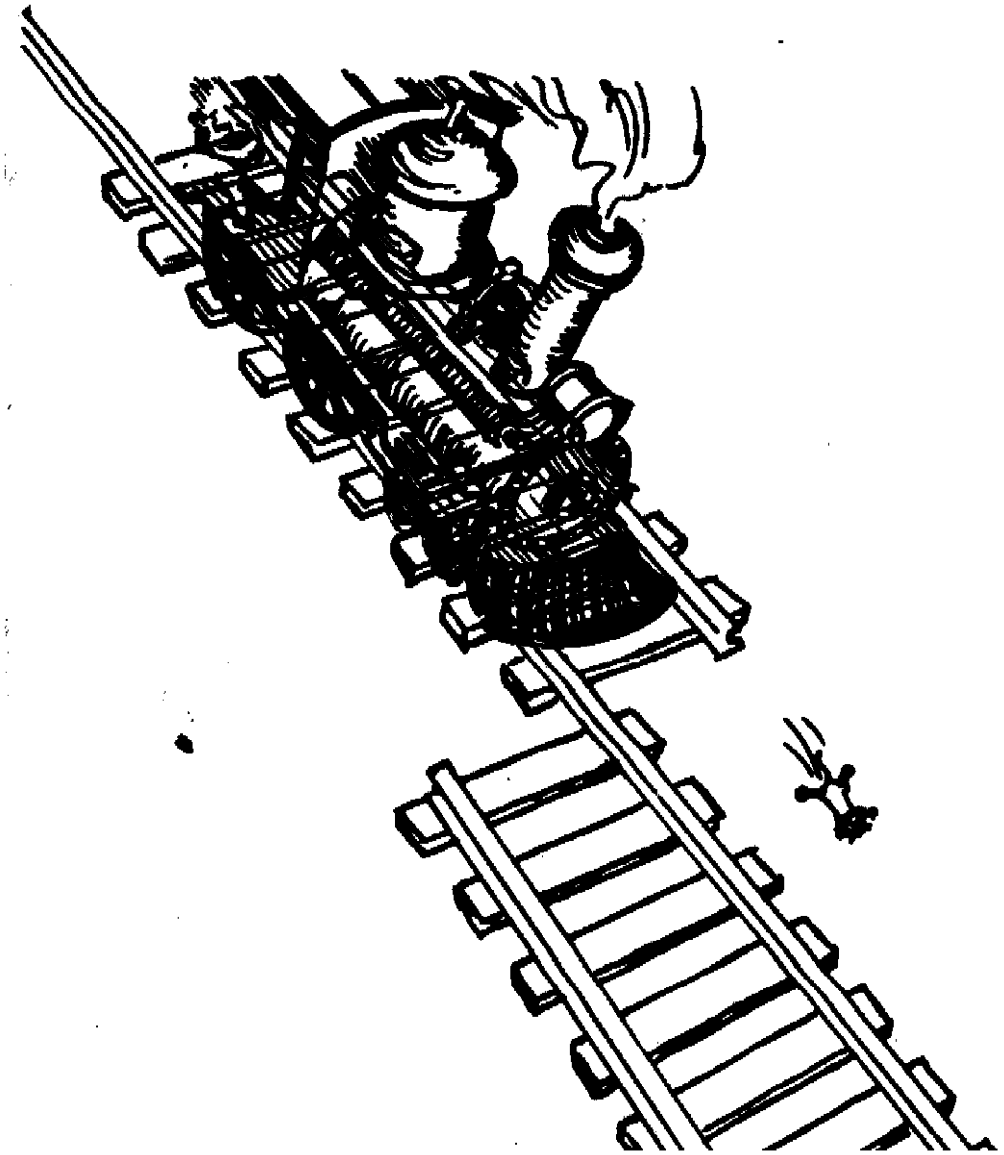
What is this?

- A failure?
- An error?
- A fault?

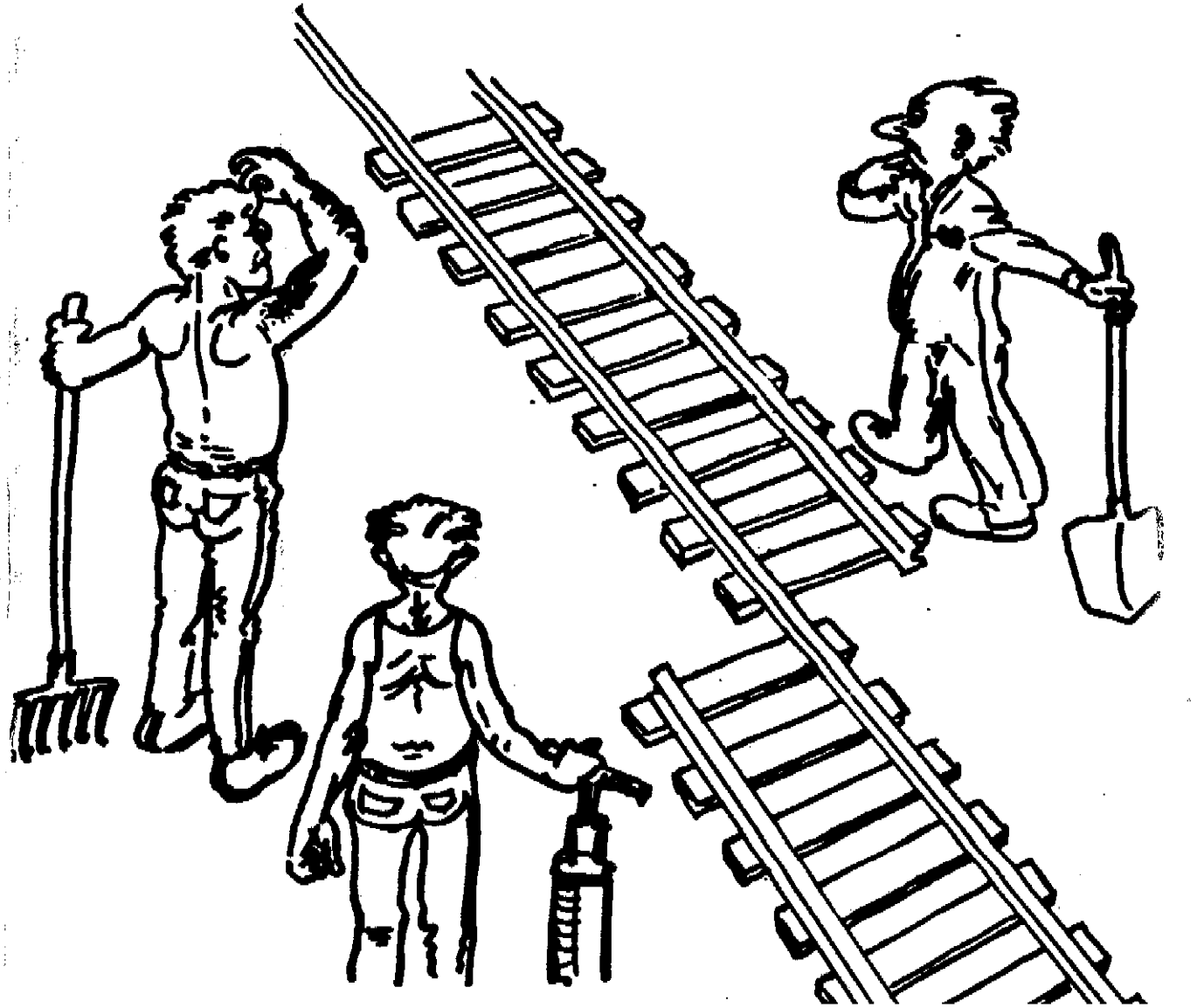


Need to describe specified and desired behavior first

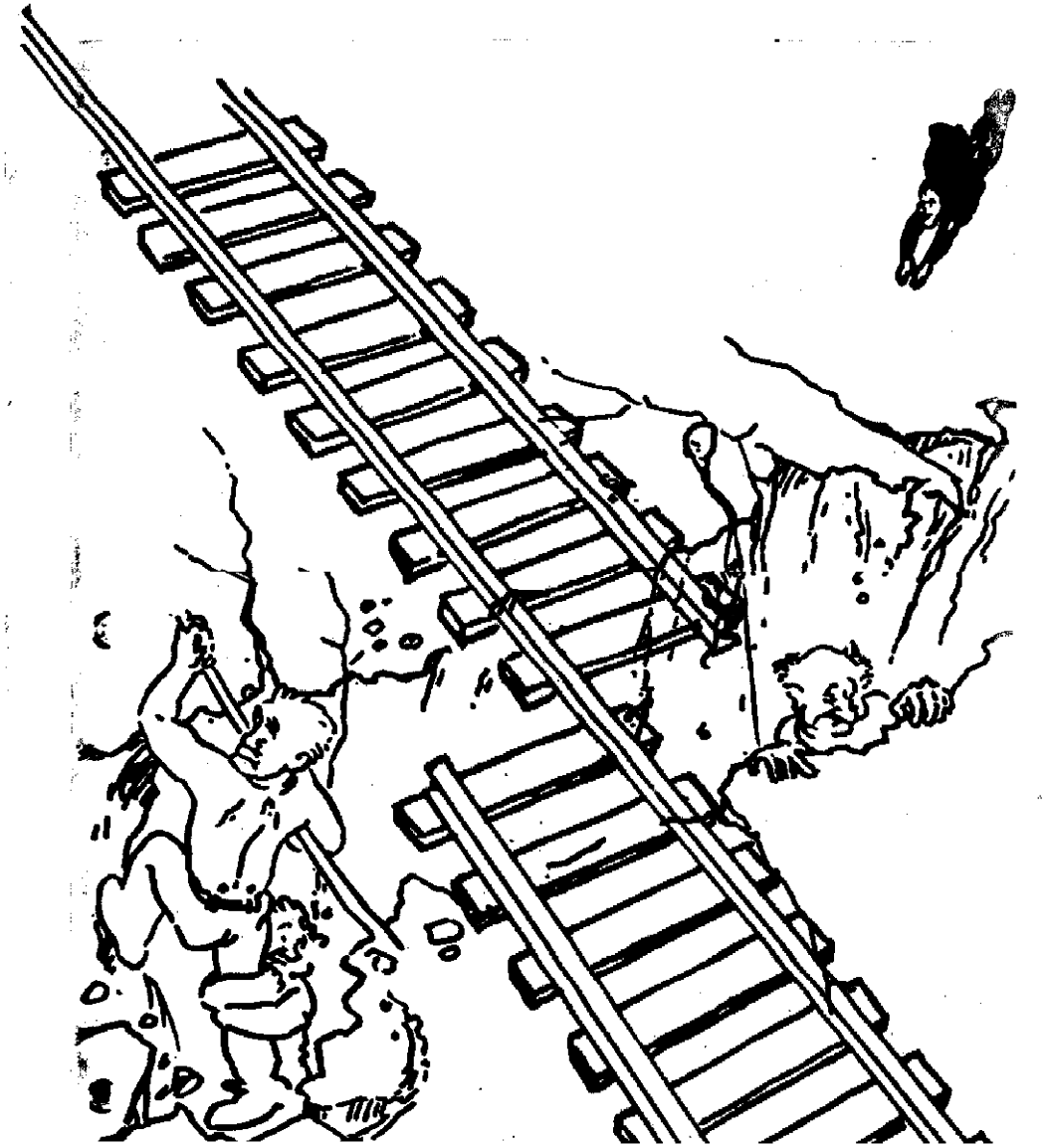
Erroneous State ("Error")



Design Fault



Mechanical Fault



Example: Fault, Error, Failure

- Therac-25 was a radiation therapy machine used to treat cancer.
- Therac-25 had two modes of operation
  - Mode A delivered low doses of high energy.
  - Mode B delivered X-rays (high beams) into a target.
- For mode B, a magnet had to be placed between the beam and the target, in order to correctly focus the X-rays.

Software Testing – Basic Definitions (An Example) Cont'd

- What happened?
- Instead of issuing a focused high beam, full powered radiations hit the patients.
- How did that happen?

1. Therac-25 had a one byte flag initialized to a non-zero value.
2. As the employee is still positioning the light beam, the flag increments.
3. As the employee finishes positioning the beam:
   a. the flag gets set to zero
   b. the magnet is placed
   c. the radiation is allowed to pass.

What is a Program State?

- A ***program state*** is defined during execution of a program as the current value of all live variables and the current location, as given by the program counter.
- The ***program counter*** **(PC)** is the next statement in the program to be executed and can be described with a line number in the file (*PC = 5*) or the statement as a string (*PC = "if (x > y)"*)
- Some code structures like loops have special treatment. The program line ***"for (i=1; i < N; i++)"*** actually has three statements that can result in separate states.

Code Example 1: Fault, Error, Failure

```
/**
 * Counts zeroes in an array
 *
 * @param x array to count zeroes in
 * @return number of occurrences of 0 i
 * @throws NullPointerException if x is
 */
public static int numZero (int[] x)
{
  int count = 0;
  for (int i = 1; i < x.length; i++)
  {
    if (x[i] == 0) count++;
  }
  return count;
}
```

A state is in error simply if it is not the expected state, **even if all of the values in the state, considered in isolation, are acceptable**.

- **Fault?**
- **For [2,7,0], what would the program state be?**
- **Do we have an error state?**
- **Do we have a failure?**

Code Example 1: Fault, Error, Failure

```
/**
 * Counts zeroes in an array
 *
 * @param x array to count zeroes in
 * @return number of occurrences of 0 in x
 * @throws NullPointerException if x is null
 */
public static int numZero (int[] x)
{
   int count = 0;
   for (int i = 1; i < x.length; i++)
   {
      if (x[i] == 0) count++;
   }
   return count;
}
```

- **Fault?**
- **For [0,7,2], what would be the program state?**

Code Example: Fault, Error, Failure

```
public static int numZero (int[] x) {
// Effects: if x==null throw NullPointerException
// else return the number of occurrences of 0 in x
  int count = 0;
  for (int i =1; i <x.length; i++) {
    if (x[i]==0) {
      count++;
    }
  }
  return count;
}
```

Expected State:
x = [2,7,0]
count =0
i =0
PC=first iteration of if

Wrong State:
x = [2,7,0]
count =0
i =1
PC=first iteration of if

x = [2,7,0], fault executed, error, no failure

x = [0,7,2], fault executed, error, failure

Program state defined by: x, i, count, PC

Code Example 2: Fault, Error, Failure

```
public static void isLeap(int year)
{
    if (year % 4 != 0) return false;
    if (year % 400 == 0) return true;
    if (year % 100 < 0) return false;
    return true;
}
```