

Today's Objectives: A set of concepts

- What is a test case?
- Expected outcome.
- Failure and fault.
- Test activities.
- Point of testing.

What is a test case?

In its basic form, a test case is a pair of

<input, expected outcome>

In stateless systems where an input does not depend on previous inputs and expected outcomes, test cases are simple in structure. A compiler is an example of a stateless software.

In reactive systems, where an input depends on the previous input and expected outcome, a test case consists of a sequence of <input, expected outcome> pairs. A telephone switching system and an automated teller machine are examples of reactive systems.

For most of the practical reactive systems, most of the test cases include some form of simple logic involving time, condition, loop and so on.

It may be noted that an expected outcome is a part of the test case.

Expected outcome:

The outcome of a program execution is a complex entity that may include:

- Observable output, such as values produced (integer, text, audio, image) and communicated (messages).
- State change
 - Program state.
 - Database state (due to add/delete/update operations).
- A series of values for it to make sense.

Ideally, the expected outcome must be computed as a part of the test design.

In exceptional cases, where, it is extremely difficult to compute the expected outcome, **observe** the program outcome and **verify** whether or not it is acceptable. In subsequent program runs, **use** the verified outcome as the expected outcome.

Fault and Failure

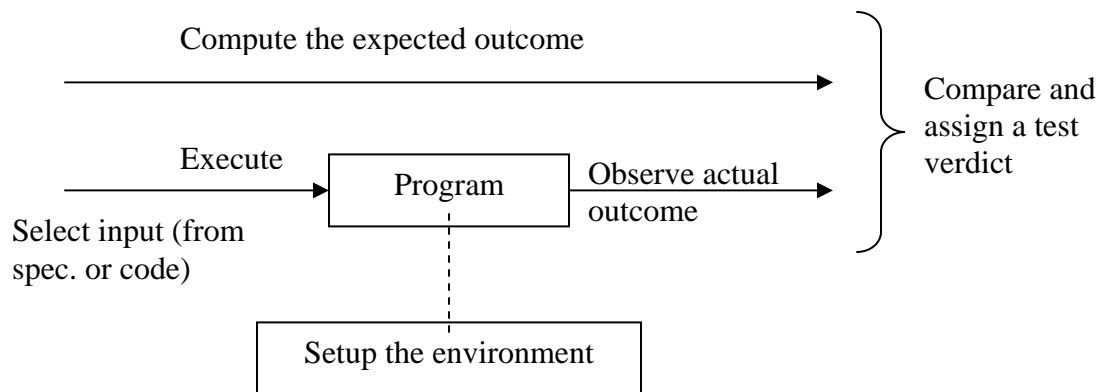
- A failure is said to occur when the program does not do what its end user reasonably expects it to do (deviation of the actual outcome for expected outcome).
- A fault (or error or bug) is the cause of a failure.

Test activities (test plan)

The process of testing a program can be organized as the following seven steps:

1. Identify a feature to be tested.
2. Select input from the specification or from the code.
3. Compute the expected outcome of the program.
4. Set up the execution environment
 - a. Set up the local system (file, resources ...).
 - b. Set up the external system (network, remote partner).
 - c. Sometimes you may have to wait for the physical environment to be set up, for example some weather pattern to occur.
5. Execute the program with the selected input.
6. Observe the actual outcome of program execution.
7. Compare the actual outcome with expected outcome and assign a test verdict. If the actual outcome is the same as the expected outcome we assign a **pass** verdict. If the actual outcome differs from the expected outcome, we assign a **fail** verdict.

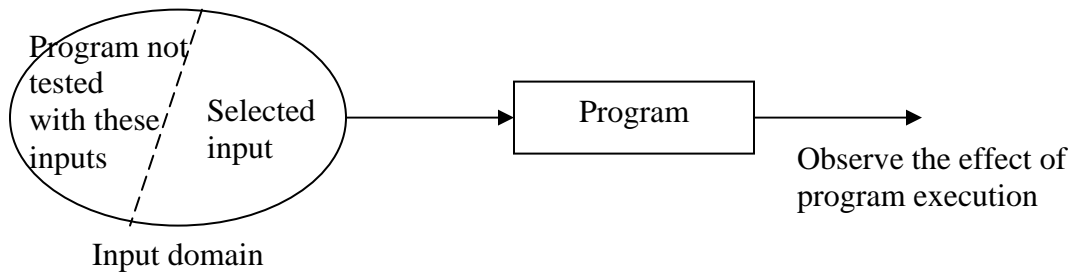
Sometimes it may be necessary to assign inconclusive verdict. Conduct further tests to refine an inconclusive verdict into a pass or a fail.



What is the point of testing (purpose of testing)

1. Phase I: Individual components work to programmer's satisfaction. The entire system works to the team's satisfaction.
2. Phase II: Show that the software system does not work. Here we try to find bugs.
3. Phase III: Reduce the perceived risk of not working to an acceptable level. We express this metric by using the idea of reliability.
4. Phase IV: produce low-risk software without much testing effort.

Testing is an **inference process**. The tester attempts to deduce properties of a program by observing its behavior on selected inputs. This is because it is impractical to test a program with *all* possible inputs. Thus, one must carefully select a subset of the input domain. This is where there is a need for studying *selection criteria* for choosing a subset of the input domain as test inputs. We will study a variety of selection criteria in this course.



What does it mean to test a program completely?

At the end of testing, there are no undiscovered errors. Thus, all problems must be known.

Complete testing is near impossible for most of the system because:

- The domain of possible inputs is too large to use in the test process (valid, invalid input, program state, timing of input).
- The design issues may be too complex to completely test.