Testability definition

According to ISO 9126, testability is defined as:The capability of the software product to enable modified software to be validated.

NOTE - Values of this sub-characteristic may be altered by the modifications under consideration.

Testability concerns

Testability touches upon two areas of concern:

- How easy is it to test the implementation?
- How test-friendly is the requirement?

These two concerns are not independent and need to be considered together.

Testability

Three basic ways to check that we have achieved our goals:

- Executing a test. Give input, observe and check output. A test can be a
  - Black box test
  - White box test
  - Grey box test
- Run experiments
- Inspect the code and other artifacts

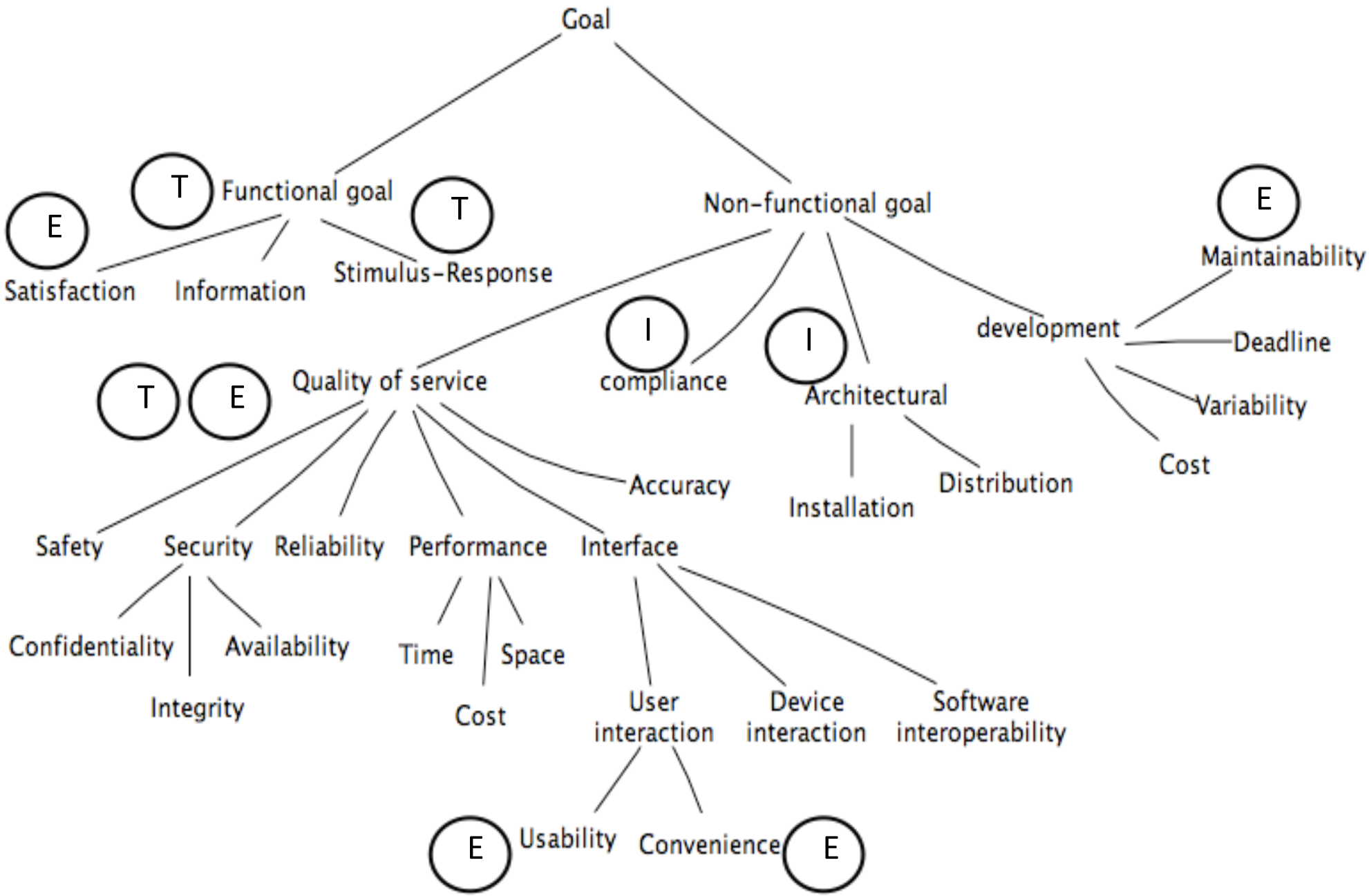
Usually, we will include all of these activities in the term testing

When to use what

The diagram on the next slide is a high level overview of when to use

- T – Tests. Input / output. Involves the computer system and peripherals.
- E – Experiments. Input / output but involves also the users.
- I – Inspections. Evaluation based on documents. They are typically used to validate static nonfunctional requirements

Concrete requirements from high level goals



Testability

In order to be testable, a requirement needs to be stated in a precise way. For some requirements this is in place right from the start:

When the ACC system is turned on, the "Active" light on the dashboard shall be turned on.

In other cases we need to change a requirement to get a testable version.

The system shall be easy to use.

Final comments

That a requirement is testable does not necessarily mean that it is ***easy*** to test.

In order to have testable requirements it is important that

- The ***testers are involved right from the start of the project***. It is difficult to add testability later.
- The tests are an integrated part of the requirement