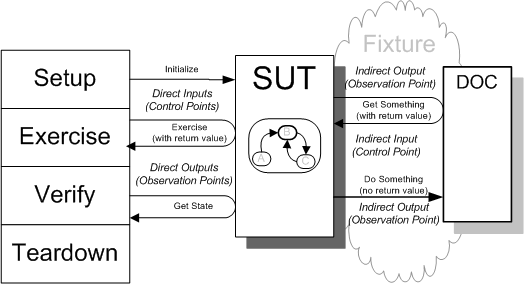
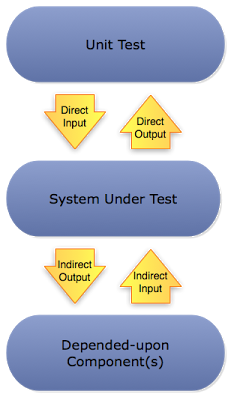
* Input has Control Points. A *control point* is how the test asks the [**system under test (SUT)**](http://xunitpatterns.com/SUT.html) to do something for it.
* Output has Observation Points. An *observation point* is how the test inspects the post-exercise state of the [**system under test (SUT)**](http://xunitpatterns.com/SUT.html). 

**Mock Object:**An [observation point](http://xunitpatterns.com/observation%20point.html) to verify the [indirect outputs](http://xunitpatterns.com/indirect%20output.html) of the SUT as it is exercised. Typically, it includes the functionality of a Test Stub in that it must return values to the SUT, but the emphasis is on the verification of the indirect outputs.

**Test Stub: r**eplace a real DOC on which the SUT depends so that the test has a [control point](http://xunitpatterns.com/control%20point.html) for the [indirect inputs](http://xunitpatterns.com/indirect%20input.html) of the SUT.

**Test Spy:** a more capable version of a Test Stub. Implements an [observation point](http://xunitpatterns.com/observation%20point.html) that exposes the [indirect outputs](http://xunitpatterns.com/indirect%20output.html) of the SUT so they can be verified.