**Negative testing**

Negative testing commonly referred to as **error path testing or failure testing** is generally done to ensure the stability of the application.

*Negative testing is the process of applying as much creativity as possible and validating the application against invalid data. This means its intended purpose is to check if the errors are being shown to the user where it’s supposed to, or handling a bad value more gracefully.*

It is absolutely essential to understand **why negative testing is necessary.**

The application or software’s functional reliability can be quantified only with effectively designed negative scenarios. Negative testing not only aims to bring out any potential flaws that could cause serious impact on the consumption of the product on the whole but can be instrumental in determining the conditions under which the application can crash. Finally, it ensures that there is sufficient error validation present in the software.

**Example:**

Say for example you need to write negative test cases about a pen. The basic motive of the pen is to be able to write on paper.

**Some examples of negative testing could be:**

* Change the medium that it is supposed to write on, from paper to cloth or a brick and see if it should still write.
* Put the pen in the liquid and verify if it writes again.
* Replace the refill of the pen with an empty one and check that it should stop writing.

**Practical Examples of positive and negative testing**

Let’s take an example of a UI wizard to create some policies. In the wizard, the user has to enter textual values in one pane and numerical values in another.