Princeton University
COS 333: Advanced Programming Techniques
A Software Testing Taxonomy

Internal testing. Designing your code to test itself. Performed by *programmers*.

Checking for function/method failures. Checking function return values. Considering exceptions thrown by functions/methods.

Validating parameters. At the leading edge of each function/method making sure parameter values are valid.

Checking invariants. At the leading (and sometimes trailing) edge of each function/method, checking aspects of data structures that should not vary.

Leaving internal testing code intact. There will always be another round of development! Using asserts.

External testing. Designing code or data to test your code.

White box external testing. External testing with knowledge of the structure of the tested code. Performed by *programmers*.

Boundary (corner case) testing. Testing with input values at the boundaries of the input domain, or with input values causing output values to be at the boundaries of the output domain. Generalizing… Testing with both normal and unusual data.

Statement (coverage) testing. Testing to make sure that each statement is executed at least once.

Path testing. Testing to make sure that each logical path is followed at least once.

Black box external testing. Testing without knowledge of the structure of the tested code. Performed by *quality assurance engineers*.

Use case testing. Testing driven by use cases derived during requirements analysis.

Stress testing. Testing with a large quantity of data or a large variety of (random) data.

Field external testing. External testing performed by *customers*.