



INTRODUCTION TO AUTOMATED TESTING



federico.moreira@globant.com

WHY AUTOMATED TESTING?

Find bugs earlier, reduce time to market, improve overall quality!

- Reduce regression testing times by lowering the amount of manual tasks involved in the testing process.
- Provide consistent regression test coverage between test cycles.
- Detect bugs earlier in the process, facilitating bug fixing tasks.
- Provide a testing harness for future upgrades to the application's lifecycle.
- Measure performance between the interaction of different components.

IS AUTOMATION ALWAYS A GOOD IDEA?

Is it always a good idea to automate? Should we automate everything?

To guarantee a successful automation initiative, consider the following:

- The application under test is stable enough and its architecture is less likely to drastically change.
- We have already defined a suite of Test Cases that cover core functionality and are executed periodically.
- We need to define stress or performance tests to understand the system capabilities.
- Certain areas of the application must be validated on different environments and multiple parameters.
- Measure performance between the interaction of different components.

ALWAYS THINK ABOUT ROI

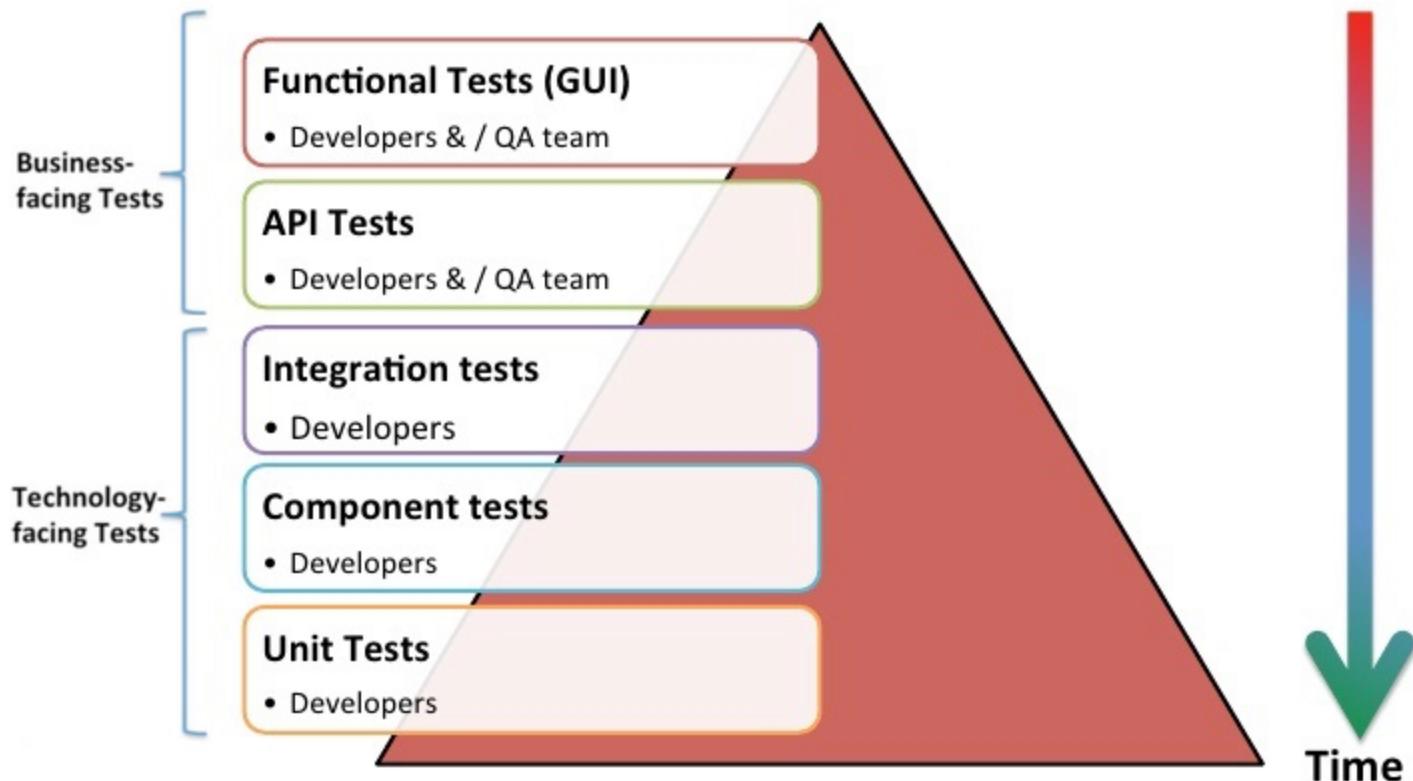
ROI stands for Return Of Investment, our goal is try to maximize the outcome of our tests based on how much we invest on automating them.

We should **avoid**:

- Automating scripts for unstable or soon-to-be deprecated.
- Consider scenarios that require manual intervention and cannot be fully automated.
- Integration tests that are too complex to automate and will not be ran often.
- Ignoring the different testing types focusing only on a single approach



TESTING TYPES



IDEAL TESTING PYRAMID

TECHNOLOGY FACING TESTS

Integration Tests: Verify the communication path between components to detect interface defects.

Component Tests: Limit the scope of exercised software to a portion of the system under test, manipulating it through internal interfaces and isolating said software from other components.

Unit Tests: Exercise the smaller pieces of testable software in the application to determine whether they behave as expected or not.

BUSINESS FACING TESTS

Functional Tests: Exercises the system from a black-box testing perspective that bases its test cases on the specifications of the software component under test.

API Tests: Focuses on determine if the different services provide the correct functionality, reliability, performance and security.



LET'S GO OVER
SOME
EXAMPLES

