LIKE A BOSS

QUICK REFERENCE GUIDE

DEVOPS TEST PLANS TERMS

This quick reference guide contains terms encountered when building and managing software Test Plans:

Unit tests

Perform tests of individual components and modules used by your software. Most often written by developers (rather than dedicated testers), unit tests are often cheap to automate, and can be run very quickly by a continuous integration server.

Integration tests

Verify that different modules or services used by your application work well together. Integration tests can validate communication with a database, or make sure that microservices work together as expected. These types of tests are more expensive to run when compared to Unit Tests, as they require multiple parts of the application to be up and running.

Functional tests

Functional tests focus on the business requirements of an application. They only verify the output of an action and do not check the intermediate states of the system when performing that action.

When compared to an integration test which may simply verify that you can query the database, a functional test would test whether or not a specific value from the database is returned.

End-to-end tests

End-to-end testing simulates a user interacting with the software. It verifies that various user flows work as expected and can be as simple as loading a web page, logging in, or much more complex scenarios verifying email notifications, online payments, etc

End-to-end tests are useful, but expensive to perform, and can be hard to maintain when automated. It’s recommended to have a few key end-to-end tests and rely more on lower level types of testing (unit and integration tests) to be able to quickly identify breaking changes.

Acceptance testing

Acceptance tests are formal tests executed to verify if a system satisfies business requirements. They require the entire application to be up and running, and focus on replicating user behaviors. They can also measure performance and reject changes if certain goals are not met.

Performance testing

Performance tests check the behaviors of the system under significant load.

Such tests are usually non-functional, help businesses and developers understand the reliability, stability, and availability of the platform. Netflix's chaos monkey is one such example, a test which randomly kills virtual machine instances running an app. For further reading or to work with the code, visit: https://github.com/Netflix/chaosmonkey

Performance tests are by their nature quite costly to implement and run, but they can help you understand if new changes are going to degrade your system.