Continuous delivery

# Definition:

Continuous Delivery (CD) is a pattern of software delivery where the system is always in a state if being ready to be deployed to production. It maximizes the delivery of new business value by decoupling changes from each other and delivering them to production as soon as they are ready. Vistaprint sees CD as a cornerstone of World-class engineering.

Continuous Delivery is characterized by the use of Continuous Integration (CI) and a high degree of automation. It also has many principles, among which we find: Build only once, Enhance visibility by the use of dashboards and Detect defects as early as possible through testing.

# Continuous Integration:

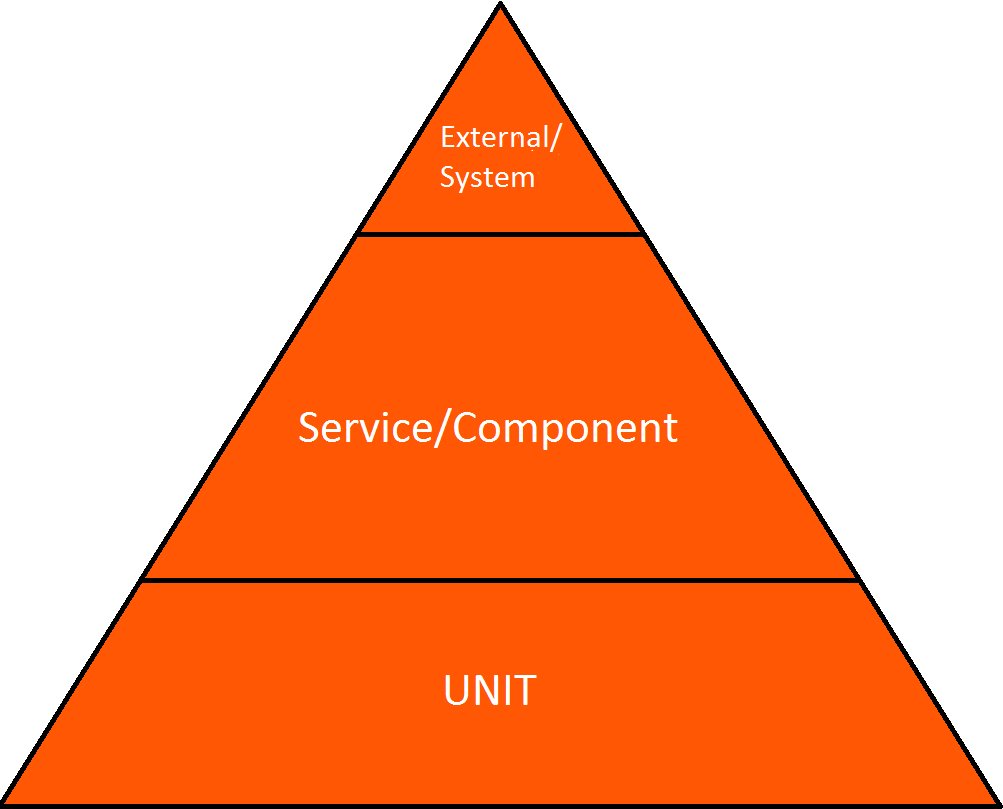
Continuous Integration is the first step of the continuous delivery pipeline. It is primarily based on the principle that delaying integration of changes only delays the detection of defects, thereby making them more expensive than necessary.

More generally, Martin Fowler defined Continuous Integration as a software development practice where members of a team integrate their work frequently. Each of these integrations is verified by automated builds to detect integration errors as quickly as possible. A good way to detect software defects in a faster and more efficient way is to include automated tests in the build process.

Vistaprint never misses a chance to renovate and be up-to-date technology-wise. It has been applying these software engineering principles using tools like Jenkins for CI ant automated testing. Regarding the automated tests, it basically focuses on the Tests Pyramid concept described below.

# Tests Pyramid:

Vistaprint uses a so-called **Test Pyramid** to guide the different squads in the trade-offs they make with regards to testing at different levels of integration.



The Test Pyramid shows three different aspects of testing available when developing and releasing software. It also provides a graphical representation of trading off costs and risks/benefits associated with doing more or less of a certain type of testing.

The first aspect is **quantity of testing**, implied by the horizontal axis. That is, tests belonging to the base of the pyramid are expected to be more numerous than tests shown at the top.

The second aspect is **level of integration**, shown on the vertical axis.Generally,low levels of integration focus on testing of small, isolated units, while high levels of integration focus on testing complex systems with many internal interfaces. Simply put, the bottom two tiers are internal in a single team, while the top layer is external, as it includes system-wide, end-to-end testing.

The third aspect is **test category** or **test objective**, and this is only implied in the diagram. Specifically, each integration tier in the pyramid will generally contain tests devised to satisfy differing objectives. Some of the possible categories are Smoke Testing, Regression testing, Performance testing, Security testing, Build verification testing, and many others.

What we need to retain here is that Vistaprint uses three test types: unit tests, service level tests, and UI (system) tests.