## Problematic:

Continuous Integration (CI) is a software engineering practice in which code changes are immediately built, tested and reported on once they are committed to the unified code base. The main idea behind this concept is to rapidly detect, identify and correct any potential defect brought on by the code change. CI detects deficiencies early on in development, therefore defects are typically smaller, less complex and easier to resolve. Nowadays its principles are widely applied within most programming models. Vistaprint is most certainly not missing out on this best practice, using Jenkins CI as their CI tool.

Jenkins is an application that monitors executions of repeated jobs, such as building and testing software projects. In fact, Vistaprint makes use of this tool to run different tasks, among which we find the unit testing. Unit testing is also a programming best practice. Simply put, it consists in a software testing method by which individual units of source code are asserted.

In our case, we are using Nunit. Nunit is a unit-testing framework for all .Net languages. Combining both Jenkins and Nunit is utterly possible. In fact, Jenkins is configured to fully support any unit test framework’s execution. However, being Java software itself, Jenkins is naturally better off handling Java based frameworks, most significantly in terms of Reporting and Data Analysis.

Every time a Nunit test is executed by Jenkins, an XML result file is generated. It is then overwritten by the next XML result file generated by the next Nunit test. This is due to the inability of Jenkins to manage and save different Nunit result files at the same time. This means that at any given time, we can have access to only the last generated XML result file. Very useful data may thus be lost, and therefore we would have no overview over the whole testing process.