MaTeLo implements a **Model-Based Testing** approach in a *user-friendly environment*. Starting from **application usages**, **business requirements** or **user stories**, testers design models able to automatically generate **optimized test suites** based on **risk** analysis, **coverage** and expected results. Test suites can be exported either to **automatic execution tools** or to **test management tools** for manual execution.
**Requirements Traceability**

Business requirements or User Stories are created into MaTeLo or imported from a requirements manager tool (HP ALM QC, Doors, Test Link,...). Requirements are assigned to the model transitions. Generated test cases steps are linked to requirements. Traceability is maintained when exporting test cases into the test manager. Requirements defects are often found when model is designed and raised early to business analysts.

**Dynamic Stimulation and Verification**

For each test step, stimulation data is generated from equivalence classes. Output test data set (Test Oracle) can be computed from treatment functions or from data sources (files, DB, Web Services, ERP...), or from external tool calculator (Matlab, Scilab) or by using Python functions.
MaTeLo offers high-performance algorithms based on risk analysis to automatically generate functional test suites which can be exported either to the test manager (QC, TestLink,...) for manual campaign or to the test automation tools.

Automated Test Execution

Generated automation scripts are described with generic keywords immediately usable in most automation tools (QTP, Selenium, TestStand...). They describe all input contexts and expected results needed. The execution of instantiated scripts drives the System or Software Under Test and automatically generates test reports.
AGILE MODEL-BASED TESTING

Agile testers design usage models from user stories. Every day they improve their models representativity thanks to the scrum meeting outputs. While the agile team designs, develops and runs unit tests, the agile testers design an application usage model, then generate functional test suites and automation scripts. Code and script tests are both available to be executed at the end of each sprint.

MaTeLo, based on eclipse technology provides a friendly GUI, entirely configurable, and covers the global Model-Based Testing process. For advanced users, MaTeLo provides APIs in order to develop specific functions, connectors or even custom test cases generation algorithms.

HAVE FUN TESTING!