

Test automation is an essential enabler for agility

Constant change

In agile development, code is developed incrementally and iteratively. This means that code is constantly changed, either because of the wish to add or change functionality or because of the need to optimize the existing code (refactor) to keep it maintainable and scalable. The constantly changing code is subject to regression risk. Functionally that works in a previous release can be broken by refactoring. If rework is done often, regression testing must also be done often. If regression testing is done manually, this claims a lot of capacity from the agile team. Also throughput times for doing a regression test will be time consuming. Only by automating the regression test, lead times can be kept short so there is a quick turn around of the regression test results.

Frees up time for high quality manual testing

High quality testing is more than checking if a product meets its specifications. A product can meet all its specification and still not add the expected business value. To investigate if added functionality also adds the expected business value, high quality exploratory testing must be done. This type of testing can only be done by humans. Therefore, the human capacity in agile teams must be reserved for exploratory testing as much as possible. This requires automation of as many tests as possible so they can run automatically.

Includes black box functional and acceptance tests

Not only the low level unit tests must be automated, also the higher level black box functional tests and the acceptance tests must be automated as much as possible. Traditionally these tests are lengthy and involve a lot of hands on testing. Acceptance tests by nature have a checking character; upfront acceptance criteria are defined and during the acceptance test it is verified that the product meet this set of acceptance criteria. This makes them perfect for automation.

Certain tests cannot be done manually

There is a class of tests that is impossible to do manually. Typically, performance testing can only be executed by an automated setup. This also is true for tests that require timing critical manipulation of data, massive data sets or long periods of monitoring specific program metrics.

Levels and check points for test automation

The levels for *Test automation* are typified as follows:

- **Forming:** *Test automation provides added value*
- **Norming:** *Test automation at different test levels*
- **Performing:** *Maintainable and continuous*

Please find the checkpoints below.

Forming

1. The organization acknowledges the importance of test tooling and allocates sufficient budget
2. Test cases are automatable by describing them using 'inputs - actions - expected results'
3. Decisions on what to automate are based on risks and return on investment

Norming

1. Automated unit testing is part of the development process
2. Tests are automated to save time for manual testing
3. Test cases are designed for automation
4. Test automation is seen and treated as software development

Performing

1. Automation is modular and maintainable
2. Requirements are set up to be executable
3. Teams are allowed to try different tooling to reach their own optimal setting
4. Test automation is used to support continuous integration

[Terug naar TI4Agile](#)