

Agile work asks for agile environments

No environment no work

Waste in the form of waiting for environment availability or set up must be minimized. When the team has no access to an environment or the environment is unstable or difficult to operate, the ability to work effectively and efficiently is negatively impacted. As development and testing are parallel activities in agile development, having dedicated environments is key.

Multiple environments

The short increments of agile development require efficient work. Multiple independent environments will facilitate parallel work. This enables automated tests to run undisturbed in their own controlled environment while e.g. development and manual testing are performed in an other environment.

Fit for purpose

Non functionals and end-to-end testing put their own special demands on environments. End-to-end environments are often complex and exist of multiple, connected systems. It is essential that the capabilities of the environment are sufficient to perform all required actions by the team. This includes sufficient, synchronized data sets throughout the system landscape.

Under team control

The environment(s) must be under full control of the agile team so they have on-demand access to the environments to perform the actions they see fit. This is especially challenging for end-to-end environments as parts of the systems may be under control of other teams or even external parties. No matter who manages the environment, the data in it is important and needs to be able to be updated and changed often. Workable solutions must be found to solve these challenges.

Levels and check points for environments

The levels for *Environments* are typified as follows:

- **Forming:** *There is an environment*
- **Norming:** *The team is stakeholder*
- **Performing:** *The environment is flexible*

Please find the checkpoints below.

Forming

1. The environment is managed (setup, maintenance, version management, etc.)
2. The environment is sufficiently representative for the tasks
3. The data used in the environment can easily be refreshed

Norming

1. The required environment is always available to the team
2. Changes to the environment are communicated to the involved teams
3. A version management system is used for the environment(s)

Performing

1. Testing takes place on all relevant environments
2. The environment supports the use of continuous integration
3. The team manages its own environments
4. A mechanism to efficiently support testing in different configurations is in place (e.g. virtualization)

[Terug naar Context Driven Testverbetering](#) | [Terug naar TI4Agile](#)