

Defects reduce the business value of the product; therefore careful defect management is essential.

Defects are waste and waste must be managed

The process of software development is never perfect, resulting in defects in the product. Defects reduce the value of the product so they must be fixed as quickly as possible. Insufficient management of defects will make that the defect is longer in the product, reducing the effectivity of the team. Bugs pile up when not resolved quickly, resulting in a backlog of work not done that slows down the team. Cleaning this up in a later phase results in extra effort for the team later: it is better to clean up bugs immediately.

Defect management has many faces

The way in which defects are managed can vary depending on several factors. It is important to have 'just enough formality' in defect management to ensure that it is fit for purpose. Factors allowing for more informal management are (amongst others): co-location, maturity of the team and small sized development. Examples of factors pushing towards more formal management are: high number of defects, complexity of the product or a drive for defect metrics or defect management process in the organization.

Whole team responsibility

The whole team, inclusive the product owner, is involved in defect management. The team finds defects and corrects them as soon as possible. Defects that, for whatever reason, cannot be solved within the sprint are brought to the attention of the product owner, who analyzes the business impact and decides if and when the defect must be fixed.

Levels and check points for defect management

The levels for *Defect management* are typified as follows:

- **Forming:** *Defects are logged*
- **Norming:** *Not all defects are logged*
- **Performing:** *Focus on working software*

Please find the checkpoints below.

Forming

1. All persons involved in logging and/or tracking defects use the same defect tracking tool
2. Defects have a common set of characteristics (e.g. unique id, description, status)
3. All raised defects are followed up

Norming

1. There are simple criteria for deciding when to log a defect
2. Defects that can not be fixed within the sprint have to be added to the product backlog
3. Defects, either logged or not logged, are communicated with the complete team
4. Impact of defects on the product, but also on the sprint needs to be taken into account

Performing

1. The team strives to fix defects found in this iteration as soon as possible (<1 day)
2. The person that logged the defect helps to solve the defect
3. Defects are logged with the minimal set of information needed to follow up the defects

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