

**Although working software is the primary measure of progress, there are other metrics that will help to provide the right software.**

*Different categories of metrics are collected*

Collecting data about all kinds of things and categorizing the types of data provides more insight. Automatically gathering data does not cost much, so when gathering is started at first, later on it is possible to select what is useful enough to keep collecting it. The aid of [tooling to do static code analysis](#) is essential. Your context defines which specific metrics are most useful.

*Product and process are adjusted based on the metrics*

Gathering data is not enough, it needs to be interpreted and then lead to actions. When useful metrics have been selected, target values are defined after which the path to reach the targets is planned. This can lead to actions to improve the product and/or the process.

## Levels and check points for metrics

The levels for *Metrics* are typified as follows:

- **Forming:** *Metrics are collected*
- **Norming:** *Metrics are analyzed and used*
- **Performing:** *Trends are analyzed and used*

Please find the checkpoints below.

### Forming

1. Developers are aware of the value of metrics
2. Code statistics are collected (LoC, #classes, ...)
3. Code quality metrics are collected (cyclomatic complexity, maintainability index, ...)
4. (Unit) test coverage is measured (code coverage, branch coverage, ...)

### Norming

1. Which metrics are gathered is based on conscious decisions
2. Target values for the metrics are defined
3. Metrics are used to improve the quality of the code

### Performing

1. Trends over metric data are analyzed
2. Processes are adjusted based on the trends
3. Target values for the metrics are regularly evaluated and adjusted when necessary

---

[Terug naar I4Agile](#)