

**Reliable estimation of all relevant activities adds to a realistic planning of software development. Negotiation with stakeholders and managing stakeholder expectations are important planning aspects.**

*Estimation at different levels*

It is difficult to look and think far ahead in time. This makes it hard to estimate lengthy development projects. In Agile/Scrum projects detailed planning and estimation is focused on the next sprint, typically 1-4 weeks in length. It is easier to plan and estimate work reliably for a short time period in the near future. Estimation is usually performed by doing [planning poker](#).

Multiple sprints of one or more teams form a release. Release planning takes care of allocating product backlog items to teams and (roughly) to sprints. In this way the organization works towards consistent releases and the business receives a prediction about when specific software deliverables can be expected.

*Work break down*

All activities needed to deliver (potentially) shippable software are to be planned and estimated. This includes testing activities like: development orientated testing, acceptance orientated testing, regression testing, test set maintenance, test automation, non-functional testing and ('just enough') test documentation. Overlaps and gaps are carefully considered when identifying the various test activities. For development this includes activities like: architectural design, technical documentation, implementation, unit testing, and fixing.

The work for the team is broken down into tasks (like 'programming tasks', 'testing tasks' and 'documentation tasks'). The level of task detail is up to the team but it is a good practice to keep tasks small so they can be completed within a day. This creates flow within the team. The team determines the optimal task order (while respecting stakeholder priorities) and identifies which team member can work on which task.

*Stakeholders*

In traditional (non-Agile) projects a project manager has major influence in the estimation and planning process. In Agile projects experienced teams know their 'velocity' and are able to reliably predict what can be done in each sprint and what the impact is of adopting change requests within a sprint. Proven reliability of estimation and planning builds up mutual trust between the teams and their stakeholders. In all cases agreement on the planning with stakeholders is crucial.

*Stay in control*

Each team keeps track of its progress and reflects on the ability to meet the committed planning. When changes to the planning are needed, time box boundaries of sprints are respected. Rather than shifting the sprint deadline, scope is transferred to a later (extra) sprint or a next release.

## Levels and check points for planning & estimation

The levels for *Planning & Estimation* are typified as follows:

- **Forming:** *Know what to plan and estimate*
- **Norming:** *Measurable, small parts*
- **Performing:** *Time is fixed, scope is flexible*

Please find the checkpoints below.

### Forming

1. Each activity is planned (prepare – define – execute – fix – retest – refactor)
2. Test levels are defined and overlap is minimized (Unit Test – System Test – Acceptance Test)
3. A technique for effort estimation is applied
4. Planning is agreed with the stakeholders

### Norming

1. Estimation is used to estimate team effort (cross functional)
2. The planning meeting is used to elaborate on user stories
3. All work is broken down into measurable, small tasks (preferably 8hrs or less)
4. Burn-down (or up) charts are used to keep track of progress

### Performing

1. Scope is adjusted when estimation seems to be incorrect, since time boxes are fixed
2. Changes on user stories are fit into the planning wherever possible
3. The team(s) decide what is delivered in an iteration
4. Team velocity is used in planning

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