



# Light-Weight Processes for Increased Agility

## Origin

There is a natural tendency to refine and augment systems of guidelines over time. As a backlash to this type of heavy-weight processes, the lightweight process movement emerged in the late 1990s. There, the focus was on empowerment of developers, reduction of “red tape”, and focusing on the product rather than the process.

## Core Idea

Light-weight process propose to de-emphasize non-coding activities as typically occur in the early life-cycle phases (e.g., requirements, architecture) in favor of the actual coding. They also reduce iteration durations to allow for frequent evaluations and adjustments of the project course.

## Benefits

Clearly, frequent re-assessments allow for quicker response to changed requirements (thus agile). This avoids delivering the wrong product. Highlighting process improvement activities increases productivity. However, probably the largest contribution, though, is motivating developers through empowering them..

## Limitations

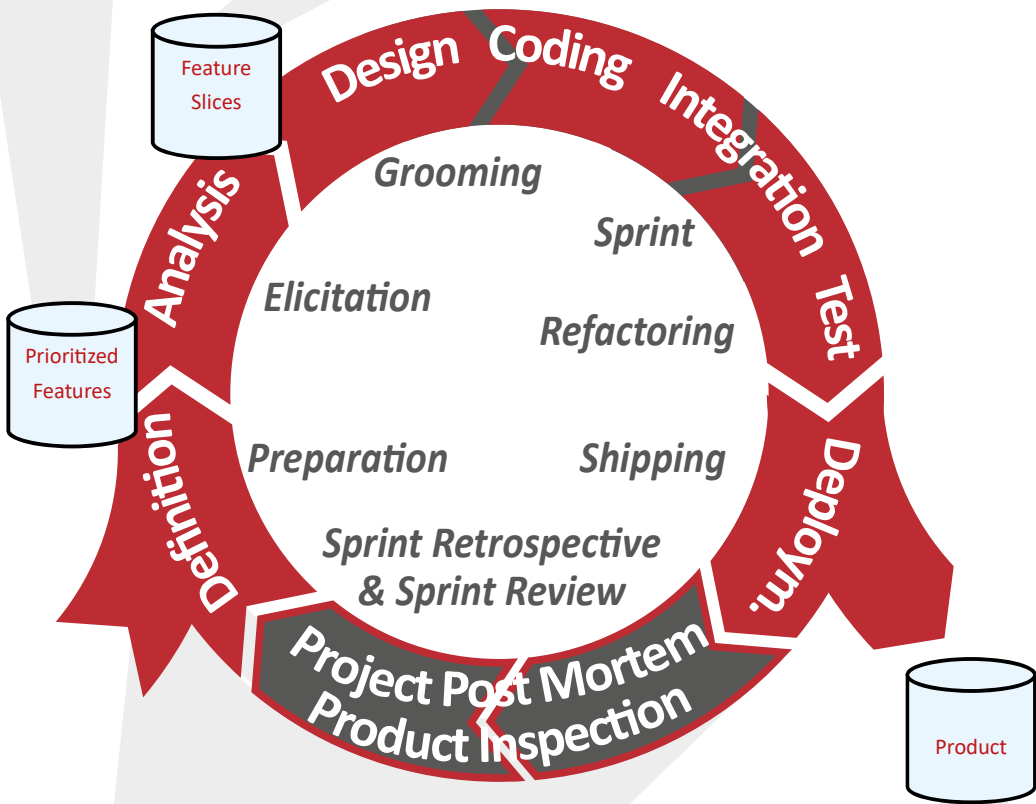
The benefit of agility is also the greatest limitation: highly flexible processes lack the rigidity needed to guide large projects. Also, every iteration has fixed cost so that the overall overhead is actually larger than in sequential processes. Finally, light-weight processes can’t ensure regulation compliance as needed in some industries.

### Product Backlog

A prioritized set of required features and qualities, often expressed as scenarios (**user stories**) is created by the product owner.

### Sprint Backlog

A set of feature slices, often expressed as sub-stories. Supposed to be completable within a sprint. The process clarifying, refining, and restructuring the sprint backlog is called **grooming**.



### Scrum Master

A group facilitator and process manager who is responsible for

- removing impediments for development team
- shielding the team from outside influence
- moderating reviews and estimation



### Product

The prototype system. After a sprint, the product is inspected (**sprint review**), and maybe refactoring is added as a new item in the sprint backlog. The past sprint is also reviewed from a process point of view (**sprint retrospective**), for exploiting process improvement opportunities.

## XP

Extreme Programming was the first lightweight development method to become popular. It re-popularized long standing proposed practices such as “Pair Programming (PP)” and “Test First” and claimed improved quality and productivity, both of which failed to materialize. Other benefits like knowledge sharing by PP are demonstrable, though.

## SCRUM

In the mid 2000s, Scrum became popular as a project management approach. In comparison to XP, Scrum is fairly vaguely defined, and every project interprets it in a different way, impeding learning. Core elements are public displays, group dynamic techniques, and a strong focus on process improvement.

## Indications

Light-weight processes are advisable when one or more of the following apply:

- 1) the requirements are unknown, unknowable, or highly volatile.
- 2) project size is fairly small (up to 10 people), and
- 3) the systems is rich in atomic features (as are often found in the MIS domain).

## Counter Indications

Light-weight processes are not advisable if one or more of the following apply:

- 1) the requirements are unknown or highly volatile.
- 2) project size is large (20+ people, geographic distribution),
- 3) the systems is rich in crosscutting concerns
- 4) system needs to comply with regulations or has high criticality.