## **Lean and Kanban Software Development**

[Lean Software Development](http://www.poppendieck.com/index.htm) is an iterative agile methodology originally developed by Mary and Tom Poppendieck. Lean Software Development owes much of its principles and practices to the Lean Enterprise movement, and the practices of companies like Toyota. Lean Software Development focuses the team on delivering Value to the customer, and on the efficiency of the “Value Stream,” the mechanisms that deliver that Value. The main principles of Lean methodology include:

* Eliminating Waste
* Amplifying Learning
* Deciding as Late as Possible
* Delivering as Fast as Possible
* Empowering the Team
* Building Integrity In
* Seeing the Whole

Lean methodology eliminates waste through such practices as selecting only the truly valuable features for a system, prioritizing those selected, and delivering them in small batches. It emphasizes the speed and efficiency of development workflow, and relies on rapid and reliable feedback between programmers and customers. Lean uses the idea of work product being “pulled” via customer request. It focuses decision-making authority and ability on individuals and small teams, since research shows this to be faster and more efficient than hierarchical flow of control. Lean also concentrates on the efficiency of the use of team resources, trying to ensure that everyone is productive as much of the time as possible. It concentrates on concurrent work and the fewest possible intra-team workflow dependencies. Lean also strongly recommends that automated unit tests be written at the same time the code is written.

[The Kanban Method](https://www.versionone.com/what-is-kanban/) is used by organizations to manage the creation of products with an emphasis on continual delivery while not overburdening the development team. Like Scrum, Kanban is a process designed to help teams work together more effectively.

Kanban is based on 3 basic principles:

* Visualize what you do today (workflow): seeing all the items in context of each other can be very informative
* Limit the amount of work in progress (WIP): this helps balance the flow-based approach so teams don ‘t start and commit to too much work at once
* Enhance flow: when something is finished, the next highest thing from the backlog is pulled into play

Kanban promotes continuous collaboration and encourages active, ongoing learning and improving by defining the best possible team workflow. See how VersionOne supports [Kanban software development](https://www.versionone.com/Product/Kanban-Software-Kanban-Boards).