LEAN/KANBAN ANALYSIS

Kevin Oomen

FONTYS EINDHOVEN

Inhoud

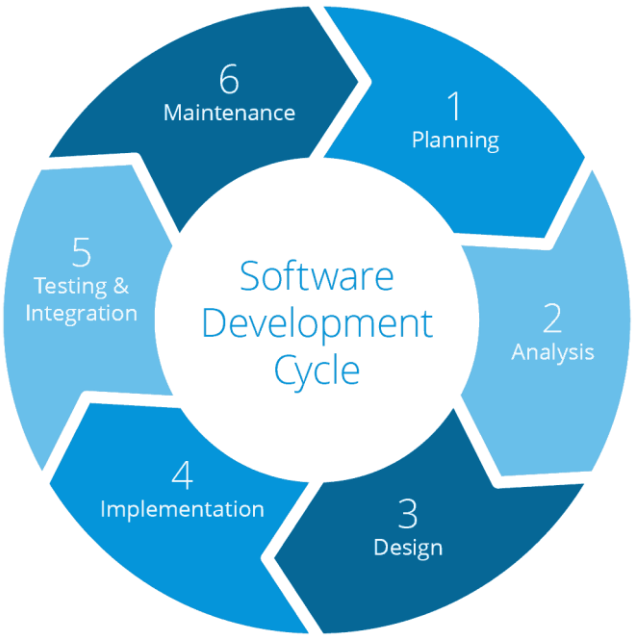
Opening

In this document I will explain what Lean and Kanban are and what the uses of them are. I will do this by performing research on internet. Whenever I state a definition, I will state the source of the definition next to it. Further sources will be stated at the end of the document.

Why use Lean or Kanban?

Before we answer this question, we first need to answer a couple of other questions.

What is Agile software engineering?



Why is this important? Because Lean and Kanban are both agile working methods. So first of we will need to understand what Agile software engineering entails.

Definition

Agile is an iterative approach to project management and software development that helps teams deliver value to their customers faster and with fewer headaches. Instead of betting everything on a "big bang" launch, an agile team delivers work in small, but consumable, increments. Requirements, plans, and results are evaluated continuously so teams have a natural mechanism for responding to change quickly. (Source)

My interpretation

In my interpretation Agile software engineering is a way of helping teams communicate better with each other and their product owner. They do this by delivering small work and evaluate this work with the product owner, so the product owner has a good idea how process is developing.

What is Lean?

Defenition

Lean software development is a translation of lean manufacturing principles and practices to the software development domain. Adapted from the Toyota Production System, it is emerging with the support of a pro-lean subculture within the Agile community. Lean offers a solid conceptual framework, values and principles, as well as good practices, derived from experience, that support agile organizations. (Source)

My interpretation

From what I have gathered, Lean uses a set of principles:

Eliminate waste

This means eliminating as much “waste” as possible. In this context waste is defined as everything that the product owner considers as non-valuable. The key is to identify what is considered waste and what isn’t.

Amplify learning

Because software engineering is an endless cycle of learning and applying what you have learned, In Lean you try to minimize the documentation side of things and replace it for tiny prove of concepts which will make the learning process a lot easier. Which in turn increase the productivity.

Decide as late as possible

It’s important to decide as late as possible because it’s best to decide whenever all facts are present. If you would make decision that are not based on facts, you could possibly create undesirable situations.

Deliver as fast as possible

In a software development team, it’s important to Have good communication with your product owner. A way of doing this is by delivering as fast as possible. It will make the communication between the product owner better because there will be a lot more feedback moments.

Empower the team

It's the reversal of roles were the developers get to make decisions of their own instead of project managers making these decisions for them. The philosophy behind this is that developers know better how to do their own job then the project managers do.

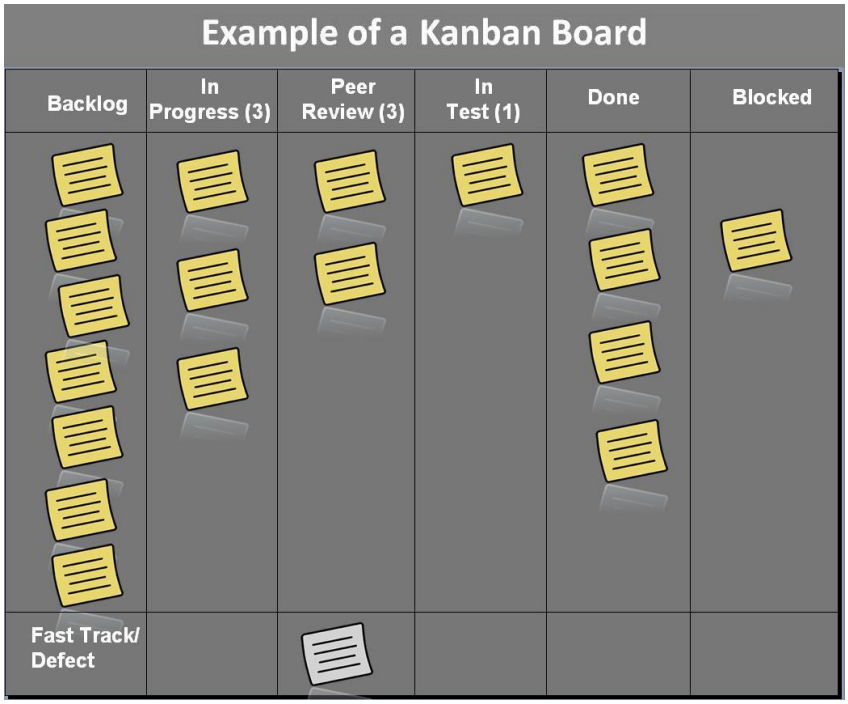
Build integrity in

This means creating the project in rows of functionality's so the project owner can see how the process is going. If first of let's say the back end of the software would be made. The product owner won't have a good vision of how the project is coming along. That's why the project will be build functionality after functionality so the product owner can see good flow of the project.

Optimizing the whole

By dividing big tasks into small tasks, it will be a lot easier to find and terminate bugs in the project that otherwise might be accumulating overtime.

What is Kanban?



Definition

Kanban is a popular framework used to implement agile and DevOps software development. It requires real-time communication of capacity and full transparency of work. Work items are represented visually on a Kanban board, allowing team members to see the state of every piece of work at any time. (Source)

My interpretation

Boards

Kanban uses boards to visualise the workflow and what must be done. These vary drastically with every project, so every board is unique to every project. Working with boards is an easy way to show your team and your product owner how the process is going. It also shows the product owner a sense of work as in it shows that they are physically getting work done instead of digital things the product owner won't necessarily see.

Workflows

In the boards it's important to set boundaries on how many things can be on "in progress" at the same time. This makes it so the board and the team won't be overwhelmed by tasks. It is important to first finish what you started before working on something else.

Metrics

Team velocity

The number of tasks that a team can complete in a sprint.

Lead and cycle time

The average time that it takes to complete a task.

Actionable Agile metrics

Use the previous data to guess more accurately about how the next sprint will be planned.

Conclusion

Lean

Lean is for teams that are focused on creating as little documentation as possible and people who learn by doing stuff. Lean also focuses only on the things that are of value to the product owner. Instead of long documents with lots of technical words for example. These kinds of thing are not important to the product owner and should therefore not be made at all.

Kanban

Kanban is for teams that are focussed on creating a good flow and product owner communication. It will make sure that the flow of the project will ben kept in a Kanban board where the team and product owner can easily see what is going on in the project. Working with a Kanban board also helps the product owner realise that the team is performing physical work by looking at the Kanban board.

Sources

https://en.wikipedia.org/wiki/Kanban_(development)

https://en.wikipedia.org/wiki/Lean_software_development

https://www.atlassian.com/agile

https://www.atlassian.com/agile/kanban