

Software Kanban

A Visual Process-Management System for Software Development

Marko Oreskovic, Kevin Rabe, Andreas Ohmer,
Sebastian Müller & Alexander Tkachov

Contents

1	What is Kanban?	2
2	History of Kanban	2
3	Kanban outside of Software Development	2
4	Kanban in Software Development	2
5	The Kanban-Board	2
6	Benefits of Kanban	3
7	Conclusion (vielleicht?)	4
8	Sources	4

1 What is Kanban?

Software Kanban is a process-management system in software development which originated in the japanese automobile industry in the 50s. Usually a Kanban-Board is used for a visual **presentation** (besseres Wort??).

Kanban is giving people permission to think for themselves. It is giving people permission to be different: different from the team across the floor, on the next floor, in the next building, and at a neighboring firm. It is giving people permission to deviate from the textbook.

- David J. Anderson

2 History of Kanban

3 Kanban outside of Software Development

4 Kanban in Software Development

5 The Kanban-Board

6 Benefits of Kanban

6.1 Transparency

Everyone has the same amount of information and is able to see at all times the current progress of the project, the individual tasks and the activities of the team members, making questions like "What is the status of *xyz*?", "What is *Person A* currently doing?" etc. redundant. This saves time for everyone and reduces the amount of miscommunication which could occur.

6.2 Benefits of the Pull

Instead of having a manager *pushing* tasks to the team members, they *pull* the tasks themselves. This has multiple benefits:

- Because the team members decide themselves on which task they want to work, they will choose the task they are most confident to solve. This results in an efficient distribution of the skills of the various team members **among** the tasks.
- Teams only pull tasks when they are done with their current task. This way future tasks do not get delayed when they miss the deadline of their current task. This could happen when a team runs into technical difficulties, which were not identified during the risk management, or when team members are temporarily not able to participate on the project (eg. because of illness.) As a result bottlenecks, where others have to wait for the late team, can be avoided and less time is wasted.
- If a team finishes its task earlier than expected, they can just pull the next available task instead of having to wait. This, again, reduces the waste of time.

However, having no deadline has also its risks: Because the team members are responsible for their own pace of work and there is no pressure from above in this matter, the development time can get high, if the team members decide to not work hard.

This means, if you consider to introduce Software Kanban to your team, you have to make sure, that your team is trustworthy and self-motivated, otherwise the risk mentioned above could occur.

6.3 Ease of Use

6.4 Motivation

7 Conclusion (vielleicht?)

8 Sources

- Epping, Thomas: *Kanban für die Softwareentwicklung*. Springer-Verlag 2011