

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?)	5
8	Sources	5

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.

Another positive aspect is that stand-ups/meetings become more efficient because less time has to be used to get everyone on the current status of the project.

6.2 Ease of Use

Kanban-Boards are pretty self-explanatory, only some minor details need to be explained to a beginner. They are also often implemented with physical tools, like a whiteboard or a pinboard. This makes the overall use of it easier, compared to a software implementation on a computer, because there is no knowledge required to open/read/edit a Kanban-Board. Using physical tools also means a higher flexibility because you are not restricted by the capabilities/features of a software application.

6.3 Limits

The use of limits in each section of a Kanban-Board has two benefits:

1. Limiting the amount of work one team member can do means that he is only allowed to work at one task/a few number of tasks at the same time. This requires the individual to focus on the work he is currently doing and it reduces the amount of distractions.
2. Limits prevent a build-up of a backlog for the team members who work on the later parts of the work chain **Begründung hinzufügen!!!**

6.4 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 could choose the task they are most confident to solve. This results in a efficient distribution of the skills of the various team members **among** the tasks.
- Instead of pulling the task they are most confident to solve because of their experience, they could pull a task they are less familiar with. The possible benefits of this will be discussed in a later section.

- 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 not to work hard.

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

6.5 Motivation

The use of Kanban could have the following benefits in regards of the motivation on the team:

- Giving the the team members the freedom of choice by pulling their own tasks raises the morale because the members do not get the feeling of doing something just because someone else told them to do it. Choosing the next task becomes a personal choice.
- The team members can decide whether they want to pull a task which requires experience they already own, resulting in a higher quality of the end-product, or working on a task they are less familiar with, resulting in widening their experience and acquiring new skills.
- Another motivation-rising aspect of Kanban is the Kanban-Board: Team members can literally see how the backlog and the to-do-section are shrinking, the done-section is rising and the individual task are moving from left to the right on the Kanban-Board. This gives a sense of progress and accomplishment.

7 Conclusion (vielleicht?)

8 Sources

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