“Agile” is **not prescriptive** which means that it’s not by itself defying exactly HOW to obtain agility. Through the last 40 years there were many approaches (with better and worse results) to creating a concrete guidebook or **framework** that will tell exactly how to obtain agility in organization. Some of those frameworks were formalized even before Agile Manifesto (2001) but in their most updated versions they are in line with manifest’s message.

What is Scrum Ban?

Concept of ScrumBan was first formalised in “Scrumban: Essays on Kanban Systems for Lean Software Development” written by a software engineering practitioner Corey Ladas in 2009.

In his essays Corey stated that adapting Scrum framework is for sure a good **first step** to introduce LEAN concepts in software development. What he also stated is that that **first step** has to be followed by next steps as every LEAN process is continuous. Applying every detail from Scrum Guide (especially from first versions of Scrum Guide) without recognition that maybe this framework has some build-in limitations may stop a team’s ability to develop better products.

What is unique in ScrumBan?

We can remove those built-in limitations of Scrum by applying Kanban “inside” Scrum and slowly recognize which elements/artifacts/meetings of Scrum are bringing the most value and which are a potential waste and could be possibly terminated.

We can specify a few steps that are needed to transform Scrum to Scrumban. At certain points You will have to decide if you want to “break” some rules of Scrum.

1. *Visualize the work.*
2. *Impose WiP limits & add more columns to the sprint backlog.*
3. *Stop Estimating Backlog Items*
4. *Get rid of some “unnecessary” Scrum Events*
5. *Trigger your Sprint Planning aka. stop timeboxing*

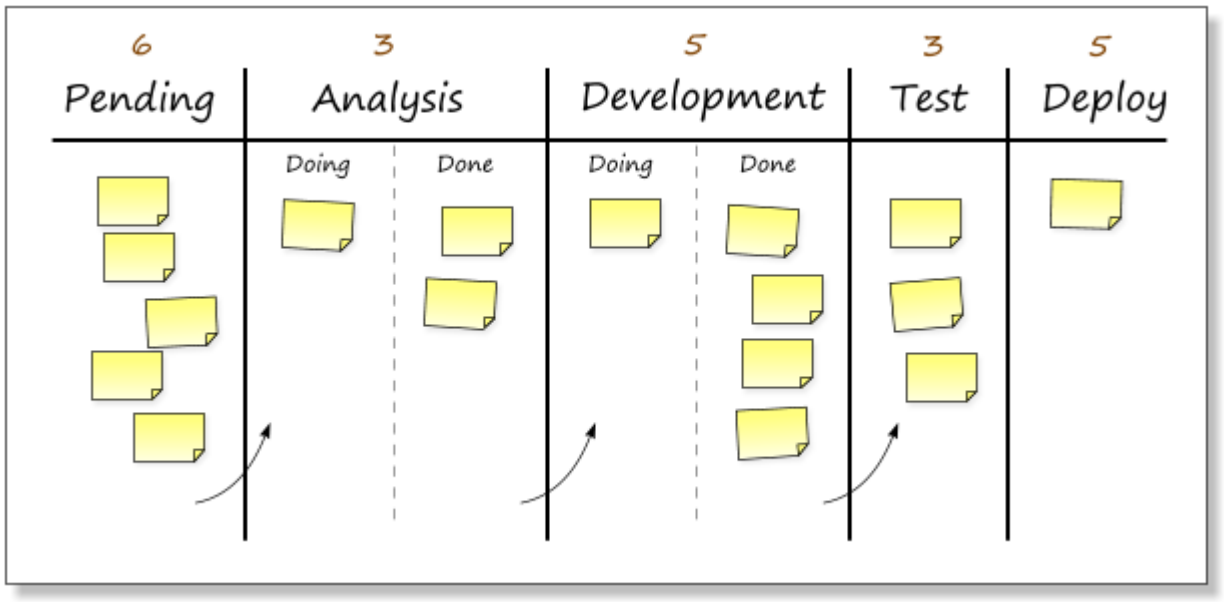
Ad. 1 Visualize the Work

At the beginning you have to be able to see the whole workflow to provide the

necessary amount of transparency & mutual understanding of the development process. Even a basic KanBan board is a great tool to do that.

Ad. 2 Impose WiP limits & add more columns to the sprint backlog.

This step is both most important and most challenging. Work in progress limits it's something that will indicate the Pull System to your current sprint workflow. To manage those "pulls" You can add a separate doing/done column into every work category.



Ad. 3. Stop Estimating Backlog Items (if You are still doing it).

At this point you will have a KanBan board for the Sprint Backlog but how to connect it to the Product Backlog? Sprint planning it's still a thing so you can use that event in the most efficient and valuable way. Estimating items from Product Backlog by using some tools like planning poker etc. could help but is widely misused. In the "Useful links section you can find an Allen Holub's "No Estimates" presentation that shows what i mean by misused).

What may be helpful to easily avoid estimation of backlog items is adding a limit not only to work in progress stepps but also to both Product and Sprint Backlog. For start you can apply two simple rules:

1. *"pulling" not more than X items into Sprint Backlog from Product Backlog*

(during sprint planning)
2. *not having more than Y most important work items **fully described** on the "TOP" of product backlog (items that will potentially be pulled by developer's*

If you are a Scrum purist or steps 1-3 are exactly what you need in your product development process you don't even have to make steps 4 and 5. At this point You have KanBan that is time-boxed and maintained by the Scrum framework. Still nothing stops you from shifting your focus from delivering **more value** to the customer every sprint to delivering value to the customer **more often** by mastering WiP limits.

Ad. 4. Get rid of some unnecessary Scrum Events & meetings

You will always benefit from focusing on the quality of scrum&non scrum events/meetings. After careful assessment & hearing feedback from your team you may end up with one of conclusions:

A. *Effects of that meeting don't have enough value yet. Let's try to make that meeting better next time.*
B. That meeting has no purpose due to the introduction of KanBan. Let's get rid of it and use time in a way that will as a result bring more value.

Ad. 5. Trigger your Sprint Planning aka. stop timeboxing

Final result of KanBan "injection" to Scrum may result in resigning from time-boxed sprints. You can still plan them during Sprint Planning but the moment of that Sprint Planning will not be fixed as before. That moment of next sprint planning will be triggered when for example there will be less than Z work items left in your current Sprint Backlog.

Summary: Pros, cons of ScrumBan

ScrumBan is using the known Scrum framework shell to "inject" KanBan. This strategy results in a minimum amount of organizational effort needed to obtain agility. In theory ScrumBan gives opportunity to benefit from all of both Scrum & KanBan qualities

Pros of KanBan:

- **ScrumBan provides a decent amount of flexibility. If your development process is benefitting from time boxing (for example Your client is constantly switching requirements and You really want to “stop him down”) you can apply Sprints of same length. On the other hand if you have mastered your workflow you can start experimenting with breaking dome Scrum guide rules (triggered sprint planning etc.).**
- **If the Scrum** framework **was already introduced & applied to your development process it will be way easier for the whole organization to understand & adapt to ScrumBan. (and finally KanBan if it’s the goal).**

Cons of KanBan:

- **ScrumBan provides a decent amount of flexibility. If your development process is benefitting from time boxing you can apply Sprints of the same length. On the other hand if you have mastered your workflow you can start experimenting with breaking some Scrum rules (triggered sprint planning etc.).**
- **You have to remember that applying any amount of flexibility into your development process will not cure existing problems by itself. For example If developers are constantly not delivering the amount of value agreed during Sprint Planning you should always carefully analyze what is a root cause of such a problem rather than instantly propose triggered planning or getting rid of estimates as an only solution.**
- **If the Scrum framework was already introduced & applied to your development process it will be way easier for the whole organization to understand & adapt to ScrumBan.**
- **It’s true if your organization was introduced to Scrum with proper attention to transparency/inspection/adaptation and Scrum values. If existing Scrum is already nothing more than a cargo cult any change of current situation will be very hard due to lack of understanding what Agile truly means.**