



The Ultimate Agile Guide

Based on experiences by Eylean

About this book

After being part of the Agile world for a while, we at Eylean feel it is time to share our experience and advice. Thus as we celebrate the fifth birthday of Eylean Board, we decided to write all of our knowledge up in an eBook for you to read and learn.

In this eBook we present on our experiences. Thus instead of the well known descriptions and facts, you will find our take on Agile as well as tips & tricks on how to make sure your journey is not short-lived.

The focus is on the three most popular Agile practices - Scrum, Kanban and Scrumban. Deconstructing them and giving you a clear comparison so that it is easy to choose the right one.

We hope you enjoy and let's dive in.

Eylean Team



4

What do we call Agile?

8

Going for Kanban

12

How do they stack up?

When it comes to Scrum

6

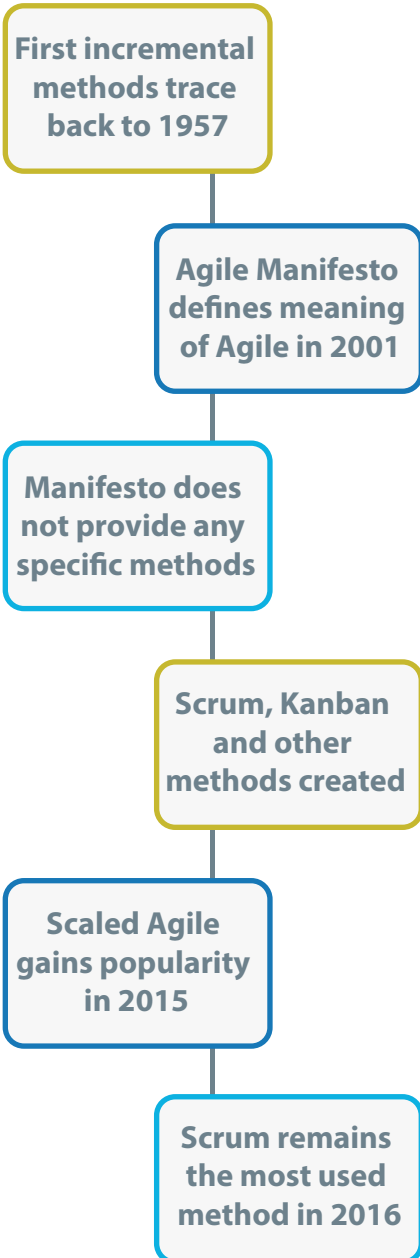
Mix it up with Scrumban

10

Tips & Tricks

14

What do we call Agile?



Agile is one of the newer project management approaches being used today. Originally it was started by the software development teams that needed more flexibility in their processes and thus the 4 values and 12 principles of Agile Manifesto came about.

Over time the methods that were first created for developers gained traction and attracted a more various group of industries and users. Many companies saw the value Agile could bring and therefore it is now applied by financial institutions, marketers, sales, R&D, governments and others.

With the users the number of Agile methods grew as well. The original ones such as Scrum and Kanban, were reinvented, merged and completely rethought by various people using the core principles. The fact that no specific method is defined in the Manifesto allowed for a variety of them and also created a search for perfection in more places than one.

Despite this constant search, the three practices associated with Agile the most, remained the same over the years. The overall favorite Scrum, laid back Kanban and a merge of the two - Scrumban.

The Agile method hierarchy

Agile

Agile
Modeling

ASD

AUP

Crystal
Clear

DSDM

FDD

Lean

Scrum

XP

Kanban

DAD

LeSS

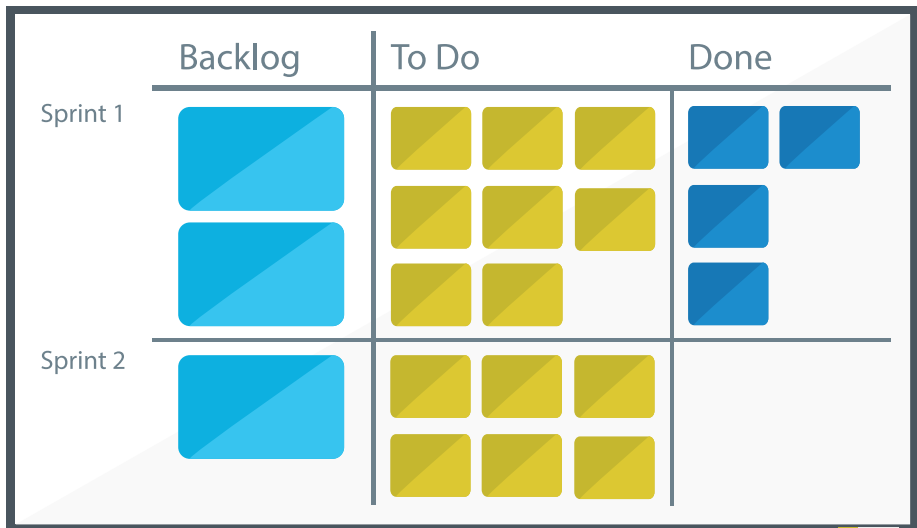
IXP

Nexus

Scum-
ban

Enterprise
Scrum

When it comes to Scrum



Scrum is one of the most popular as well as one of the oldest Agile methods and is usually the first association people have with Agile. Compared to other practices Scrum is very descriptive and thus a common first choice. However, due to the descriptiveness, it can also be restrictive and some teams find it holds them back instead of enabling the progress.

Due to this, Scrum works best with teams that are used to clear processes and have a good understanding of authority. It is also often used as a stepping stone from the traditional project management into other more loose Agile frameworks.

Even though all of the Agile methods work in a similar fashion, when choosing Scrum, teams have to be prepared for a longer adoption period. Scrum carries several roles, three types of meetings, User Stories, Story Points and Burndown charts. All of that takes time to understand and take over.

Despite this, Scrum is successfully used in various fields from development to accounting and with the new scaled approaches can also serve large teams just as well as the small ones.

Running a Scrum project

1 Plan the Backlog

The very first thing to do in a Scrum project is to fill up the Backlog. For that you will need to get all of the user stories from your client, prioritize and add them to the board.

2 Plan the first Sprint

Once the Backlog is full, choose one or more user stories, define tasks for each of them and estimate the story points they will take to complete.

3 Daily operations

Assign tasks to the team and start working. Update the board and hold 15 minute standup meetings daily to discuss progress and issues.

4 Organize a Sprint Review

At the end of the sprint, hold a review meeting for the stakeholders. Present finished work to get it approved or denied and returned to the next sprint.

5 Hold a Sprint Retrospective

To end a sprint, call a Retrospective meeting. Sit down with your team, look back on the sprint and discuss what went well and what should be improved in the future.

6 Take on the next Sprint

After that, repeat the steps - review the backlog, choose User Stories and run Sprints with your team up until you have a finished product.

7 Complete your project

Once all of the Sprints are done and the client has a finished product, hold a final retrospective. Take a final look back on the project, discuss successes, failures and future improvements.

Choose the smallest task and evaluate it first - then evaluate the rest in comparison to the first one

The team members can choose to update the board throughout the day or just before the Daily Scrum

Make sure the team delivers new value to the end user or project at the end of each sprint

Retrospectives will help the team to find their groove in the beginning and to improve later on

New User Stories can be added or priorities changed during the project so update the backlog for each sprint

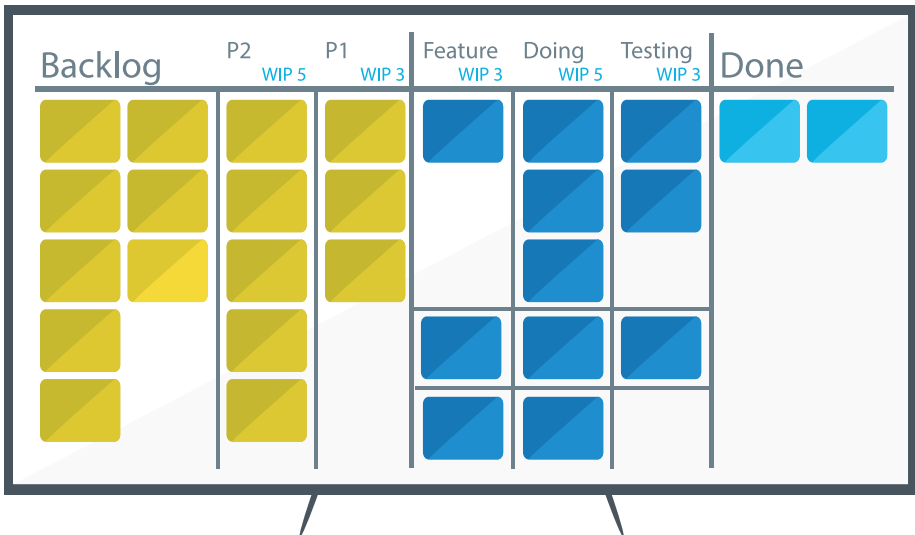
Going for Kanban

The origins of Kanban trace back to the manufacturing system by Toyota and its Kanban cards. Since then, these principles have been successfully adopted to fit as an Agile framework and now ensure teams have just the right amount of tasks in the backlog as well as the effectiveness of processes by limiting the work in progress (WIP).

Compared to Scrum, Kanban is a relatively loose method as it does not have extensive processes or roles and only monitors the team via the WIP limits. Therefore, Kanban is often seen as more adaptable and laid back, suitable for teams that have changing processes and do not want precise rituals.

Due to the lack of descriptiveness Kanban is rarely the first choice when starting out with Agile. However, teams already using one Agile method or another, are more likely to switch over to Kanban in search of flexibility and process influenced planning and reviews.

Compared to others, Kanban has a medium adoption period. While it does not carry many rules or restrictions, some teams struggle with the lack of descriptiveness and take time to fully take in the new process. So it will be up to the team to determine how much time is needed.



Start practicing Kanban

1 Understand the existing process

To start doing Kanban, you have to build your board. Analyze your processes and create a board that reflects each step with a column (Doing, Testing, etc).

2 Add WIP limits

To control the amount of work your team will be doing at any given time, add Work In Progress (WIP) limits to the priority and process columns.

3 Prioritize tasks

Once the board is set up, start adding tasks to your backlog. Since they are not prioritized by Sprints, use priority columns (P1, P2, etc.) to highlight which tasks are the most important and which can wait.

4 Daily meetings

Organize short daily or weekly meetings to keep track of what is going on and to troubleshoot any problems your team may be having.

5 Review reports

Review the lead and cycle time diagram as well as the cumulative workflow diagram frequently to make sure your team is working effectively and making progress.

6 Set up a planning trigger

Evaluate how much time you need for planning and monitor the number of tasks are left in the backlog. Once the time needed to finish remaining tasks equals the planning time, hold a planning meeting.

7 Present your work

Organize review meetings for stakeholders and present the progress your team has made. This will give you valuable feedback and direction.

Start with a simple Kanban board and add process columns to reflect your teams' everyday processes

If you are unsure about WIP limits, simply add 1 or 2 to the number of people in your team

To prioritize effectively, higher priority column task limit should be smaller than lower priority – $P1 < P2$

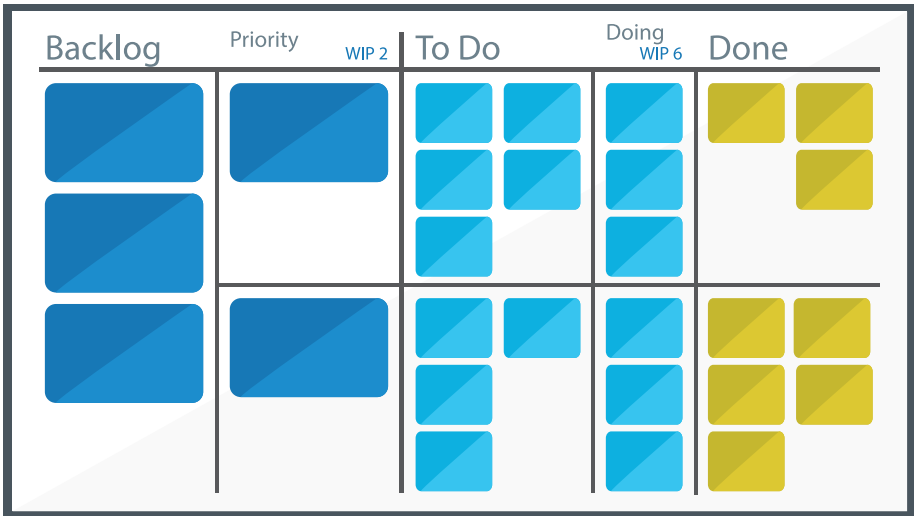
WIP does not allow taking on a new task when stuck and instead forces the team to look for solutions

Reviews can be held in agreed intervals or when the team feels they have completed an incremental change

Mix it up with Scrumban

The newest approach of the three takes the best of both worlds and mixes them to create a completely new method - Scrumban. It was first created as a way to transition from Scrum to Kanban and is now recognized as completely separate way to apply Agile.

While mixing together two different methods, one can expect that the result is rather complicated and Scrumban is just that. No matter if you have previous experience with Scrum, Kanban or both, you should not expect to understand Scrumban right away - it will take some time.



Scrumban is best suited for teams that find both Scrum and Kanban are not completely right for the job. One is too strict and the other too loose, while Scrumban sits right in the middle. As a result it is often chosen by both previous practitioners of Scrum and Kanban to enhance the results.

Inevitably, due to complexity of the method, the time it takes to adopt is longer. However, if the team is transitioning from Scrum or Kanban, some concepts will be familiar. Scrumban is a great choice for any team that needs stability in regular iterations as well as flexibility in planning.

Starting your Scrumban project

1 Create your board

Start off by taking a standard 3 column Scrumban board (To Do, Doing and Done) and making it your own with additional process and priority columns.

2 Come up with a plan

To plan in the long term Scrumban uses 1 year, 6 and 3 months planning buckets. Put the furthest plans in the 1 year, closer plans to 6 months and current plans to the 3 months bucket.

3 Start an iteration

Overview user stories in the 3 months bucket, decide which ones the team will be completing in the next iteration and add them to the board.

4 Prioritize work

Let your team know which tasks are the most important by numbering them in the prioritized order or by using priority columns from Kanban.

5 Run the iteration

Begin an iteration by letting team members pull and complete the planned tasks. Hold daily meetings and review the progress in the reports.

6 Set up a planning trigger

To make sure your team never runs out of tasks, set up a planning trigger based on team velocity. And continue to run iterations until the project deadline.

7 Freeze and Triage

Once the project deadline is coming close, freeze all of the current features and stop your team from working on them. Then triage and decide which should be finished and which will be abandoned and not completed.

Think about process steps your team works through and replicate them in the form of a board

The 3 months bucket holds your most immediate plans and is used for iteration planning

Choose the priority system that is most understandable for your team - do not use both

In Scrumban tasks are pulled by team members themselves not assigned to them by the leader

Decide which features will add the most value to the end user and complete them, forget the rest

How do they stack up?

	Scrum	Kanban	Scrumban
Rules	Very descriptive	Not descriptive	Descriptive
Board	Reset for each sprint	Used continuously	Used continuously
Roles	SM, PO and the team	Specialized team	Specialized team
Iterations	1-4 week sprints	Need based	Need based
Planning routines	Sprint planning	On-demand & release planning	On-demand & bucket planning
Task estimation	Done before each sprint	During planning (optional)	Optional
Task assignment	Assigned to the team	Taken by team members	Taken by team members
Prioritization	By refining backlog	By priority columns	By priority columns
Task limits	Limited by sprints	Limited by WIP	Limited by WIP
Meetings	Planned and mandatory	On-demand and optional	On-demand and optional
Performance metrics	Burndown chart	Lead & cycle time, cumulative flow	Lead & cycle time, Average cycle time
New tasks in a live iteration	Not allowed	Allowed	Allowed

Choose your Agile method

1

Team size



Up to 8 people



Up to 10 people



Up to 10 people



Several teams



Mid-sized company



Large company

2

Adoption time



A few weeks



One week



Up to a month



A month



More than a month



Up to two months

3

You deliver



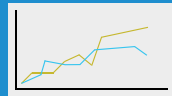
Projects



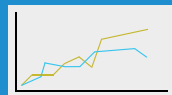
Products



Products



Projects



Projects



Projects & Products

Your method



Scrum

Kanban

Scrumban

SoS

LeSS

SAFe

Tips & tricks

1. If a team is new to agile, never implement all Scrum practices at once. Approach it iteratively - each month employing a new practice until you know them all. Be strict, motivate the team and keep up with routines.

2. If your team is well motivated and experienced, go full Scrum from sprint 0.

Sprint
0



3. Do not stick with one methodology - use Scrum for the project and Kanban for support.



4. Keep teams small and delegate ownership of both process and project.



5. Use Scrumban for small teams working on high uncertainty conditions - especially startups.



6. If your company is new to Agile - use consultants. It will save time and make on-boarding less painful.



7. To know if you are getting better or worse - start measuring your process before implementing Agile. Possible measures: (man hours, quality of output, number of defects, time to delivery, etc.).

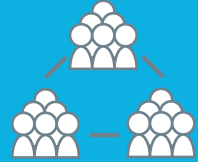
8. Kaizen in kanban is not only about production process it is also about kanban itself.



9. Multiple teams in one company should use unified tools unless their work differs drastically.

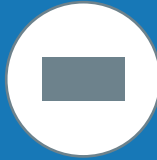


10. Scrum in small projects do not bring large benefits, smallest projects should be at least 9-12 months.



11. Strengthen your knowledge before implementing Agile methodologies, wrongly implemented ones cost both time and effort.

12. Ultimate principle - fail fast and iterate over again.



13. In Scrum, make sure task size is less than 8 man hours.

14. In Kanban use floating WIP's to maintain healthy planning triggers.

WIP

 **< 8 hrs**



15. Task board is just a single tool among other very important tools in agile methodologies. Tools are maintained by strict practices .

16. Scrum master's main job is to shield and support the team only intervening if necessary.



17. Clear and visual communication is a core value of any team.

Brought to you by:

