Scrum and ScrumBan as it’s next evolutionary step … and is it worth starting with Scrum at all

ontheagilepath.net/2015/03/scrum-and-scrumban-as-its-next-evolutionary-step-and-is-it-worth-starting-with-scrum-at-all.html

By Sebastian Radics

Original picture source: shirt.woot.com

In this post I’ll describe the evolution towards ScrumBan as a natural way for the growth of Scrum teams. Let’s in addition ask the questions why to start with Scrum at all.

There is a lot of discussion ongoing and lately I followed an interesting one about Scrum is designed for misuse by Mike Sutton and the great reply Scrum fails, I’m told endlessly by Ron Jeffries.

In the beginning…

In my experience Scrum is a great starting point for young teams – teams that are new to agile software development – where applying an agile method makes sense (e.g. use the Cynefin-Framework to check when to use what method). Scrum provides a simple set of rules and ceremonies including a role for the agile coach – the ScrumMaster – that focus on helping the team understanding agility and how to apply Scrum properly.

Structure helps in the beginning as it removes the trap of too much freedom and missing important pieces. Scrum and it’s parts build a memeplex and removing parts of it can destroy the whole meaning of it.

Scrum’s set of practices, already exercised around the world, like

- cross functional and self organizing teams,
- retrospectives,
- daily standups,
- the focus on the most important topics,
- customer value focus,
- to include the customer/user as early and often as possible

- …and more

combined in a properly described format is of high value to get into thinking and working the agile way.

A group of people become teams and not just a bunch of individuals sitting together in a room. The focus on meaningful outcomes provided by the role product owner supports the intrinsic motivator purpose.

For me this makes Scrum a great starting point to introduce teamwork and agility. It should always be accompanied by professional coaches who really guide you through the heavy change process connected with introducing such a fundamental shift.

The evolution starts…

Over the time when teams get to a more mature state they'll realize that some of the defined boundaries by Scrum (or the quasi standard added features like estimation) are actually hindering in becoming high productive and reaching the optimal flow for delivering high quality product increments. It's time to adjust and add/change/remove artifacts **consciously**.

Changing based on a deep understanding of the original subject to change and supported by data during the change (e.g. by using experiments) is in my opinion really important (e.g. by applying a lean change approach).

Consider other influences in agile like lean, management 3.0 to discover possible extensions and changes – some great teasers are:

- 2 pillars of Toyota production system
- 14 principles the Toyota way
- 5 principles of Lean Thinking
- 7 principles of Lean Software Development
- Deming's 14 principles of management

Based on lean we strive to optimize our flow (by removing all kinds of waste like inventory, delay, rework and overproduction), to deliver highest quality possible and address+discover market needs as fast as possible all done in a joyful way.

I like the 5 guiding questions by Mike Sutton:

VALUE – Do we know the value we seek to deliver and are we consistently delivering the maximum value?
FLOW – Do we understand how we reach that value and are we consistently reducing time and/or increasing the ease by which we reach it?
QUALITY – Do we understand how good our product and workmanship needs to be and are we consistently and demonstrably achieving it?
JOY – Do we know what collectively and individually need to be joyful and are we consistently meeting those needs?

CONTINUOUS IMPROVEMENT – Do we know what we need to improve across VFQJ and are we demonstrably pursuing those improvements?
I learned that changes appear in these areas:
Estimates using Story Points or Ideal days are not longer necessary (by the way so far I did not get the deep understanding of the Story Point concept …beside it's a relative estimation of SoMeThiNg and I really tried to understand it by discussing, reading blogs/books … discovering that there is even no consensus among the leaders in agile) … but good news … **use #NoEstimates** and remove the waste of estimation (and still continue understanding what to build!)

Fixed iteration length for delivering product increments hinder real flow (with the ideal of one piece flow). Using it as a protection mechanism to stop interruptions and keep focus gets replaced by discipline using the guidance:
You may not interrupt work in process and you may not adjust the work plan ~~more~~ LESS frequently than once every n days. (Corey Ladas)
I cannot not really argue why not delivering immediately (maybe waiting for a review), why to create inventory for 2 (iteration length) weeks (done by sprint planning). AHH – velocity … yes important in the beginning but replaced by cycle time (trailing indicator) and minimizing work in process (LEADING indicator).
To continue with flow improvements it's better to deliver immediately when a story is done, to pull in new stories when there is free capacity and to use measurements like the cycle time and WIP.
In a well regulated pull system iterations add no value at all. Iteration is only a transient concept to lead us to the deeper truths of pull and flow. (Corey Ladas)
Prioritization is done on demand, keeping only that much stories prepared in advance to avoid a stalling flow.

Following the approach to first bring your system into a state of control (by finding sources of variation that cause divergence and finding a way to manage them) and start to change something to improve performance.
The ideal work planning process should always provide the development team with the best thing to work on next, no more and no less.

Standups start enumerating on work items and not longer people and are focussed on problems to the flow (it's not a reporting how busy everyone is but a full focus on our product delivery, trusting everyone that they will raise impediments and do a meaningful work).

Burndown charts get uninteresting as soon as variation is under control and cycle time + WIP become key measurements.

The **release cadence gets decoupled from the iteration length** searching for the optimum point between cost of deploying a new release vs. opportunity cost of carrying finished inventory … **with ideal target of deployment on demand.**

And leads to…

Evolution leads to having a ScrumBan environment. Combining the best of the two approaches (Scrum and Kanban). Using retrospectives in intervals (iterations ;-), using dailies, maybe reviews but all considering lean.

Have fun in your journey and evolution! I'm looking forward to hear your steps? Do you see a contra-dictionary way?

To learn more …

- Scrum and Kanban – Making the most of both by Henrik Kniberg
- Lean from the trenches by Henrik Kniberg (or my summary)
- Scrum, Kanban, ScrumBan – a fast overview
- Scrum is designed for misuse by Mike Sutton
- Scrum fails, I'm told endlessly by Ron Jeffries
- Tame the flow by Steve Tendon