

Table of Contents

Should We Do WIP Limits Even Though We Are Doing Scrum?	3
Want to Know More	5

Should We Do WIP Limits Even Though We Are Doing Scrum?

Or “What is the benefit of using WIP limits in Scrum?”

I am often asked about using WIP limits in Scrum. It seems like when we do this we will set people up to be idle, that we won't fully utilize the capacity of the Team. Bottom line is that you cannot help feeling that there is something wrong with this WIP idea, when you apply it to Scrum. And I completely understand that as I had the same problem myself.

<BEGIN Theory>

What I found helped me, was understanding that I needed to change my perspective. I needed to understand the issue was not what people were working on that counted, but rather how fast and often I can deliver value, as represented by the User Stories. The phrase I use to capture this notion is:

Follow the baton, not the runners.

We need to focus on how good we are at value delivery (the baton) rather than things like people individually are doing (the runners).

The measures you use to understand whether you are improving are:

- “Cycle time”: or how long does it take you to do a story on average
- “Throughput”: or how many stories you can get done per unit time (e.g. 2 week Sprint or Iteration)

For Scrum Teams the Cycle Time is going to be 2 weeks or less, just because the Sprint (Iteration) is that long, and the throughput will be based on the number of Stories completed (closely related to idea of Velocity).

Now it turns out (as a direct consequence of Little's Law) that there is a mathematical relationship between the amount of work you have in progress (WIP), and the cycle time and throughput. Basically if you reduce WIP, you will typically reduce cycle time, and increase throughput. With Scrum Teams the problem is that there is limited room for improvement as the WIP is pretty much “lets get everything started”.

</END Theory> <BEGIN Practical>

The easiest thing to do to reduce cycle time, improve throughput, and so improve the flow of value, is to limit WIP. What does this mean?

From general Lean thinking the ideal flow is called “single piece flow”. This is the notion that conceptual a single piece of value throughs through your system, which keeps the cycle time to a minimum and so improve overall throughput.

This idea works for manufacturing, but can it be applied to knowledge work? For many years we have

known that for Scrum Teams to be really effective they need to work on a couple of Stories at a time, work them to completion, and then continue with the next stories. Stories are created and worked so that they can be completed in a couple of days, assuming we are 100% focused on the effort.

What does this really mean? It means you are setting the “In Progress” column to 2 (whatever a couple is in your world:-)). What happens when you do that? In order not to violate the WIP limit, the Team has to “swarm” (“pairing” but perhaps with more people involved) on the most important items and get those items to completion.

And that is the point. The Team has to work together to make this happen. WIP limits are used to encourage teamwork and focus. Successfully applied you should see a reduction in your cycle time over time, as well as an improvement in throughput, and therefore velocity.

What stops Scrum Teams from taking advantage of this? Usually it’s the feeling that only certain people can do certain things, and that we have to make sure people are busy. Don’t get me wrong, there is validity in this feeling; we do have specialists. But again this just means you are focusing on the runners, not the baton. By using the WIP limits to encourage “swarming” (or in the simplest case “pairing”) not only will we improve our ability to deliver value (improve out flow efficiency) but you will also:

- Improve focus as you do not try to work on everything at once
- Improve collaboration since people are working together
- Help you understand early bottlenecks, impediments, queues, and dependencies that are holding the team back since you are completing work more often
- Have faster feedback and early validation on completed items reducing risk associated with the question “are we building the right thing”
- Produce higher quality of work since you now have multiple “brains” looking at a single work item which reduce future rework
- Reduce risk of meeting Sprint (Iteration) commitment as you do not wait to deliver everything at the end
- Improve Team resilience by collaboratively developing cross functional skills
- Enhanced trust, transparency and teamwork
- Increased Team motivation of the Team members as they get to see something to Done every few days

Not bad, right? I am not saying this is going to be easy; its a big transition for the Team to make. But I think the benefits suggest a bit of experimentation is probably a worthwhile investment.

Above discussion assumes that we put a limit on the “In Progress” column. We can get even better results by breaking down the “In Progress” into the actual steps required to do the work. For example, breaking a typical development process into “design”, “implement”, “validate”, and “deploy” we could understand where we are seeing bottlenecks (work building up) and, by using WIP limits on each of the states, improve the overall flow of value as measured by cycle time and throughput.

One final comment. The “extreme” version of the approach to “swarming” is called “mob programming”. One person at the keyboard, and everyone else on the Team is helping that person be effective. Check out [A Day of Mob Programming](#) for a video of this approach to working. Basically “single piece flow” for software developers.

</END Practical>

Want to Know More

- [A Day of Mob Programming](#)
- [Why Should We Focus on Doing One Thing To Completion at a Time?](#)
- [What's The Difference Between Kanban and Scrum?](#)
- [The Resource Utilization Trap](#): Henrik Kniberg's excellent demonstration of the problem of focusing on resource utilization in bringing value to our customers

[Scrum](#), [FAQ](#), [HighPerformance](#), [Productivity](#), [WIP](#)

From:

<https://www.hanssamios.com/dokuwiki/> - **Hans Samios' Personal Lean-Agile Knowledge Base**

Permanent link:

https://www.hanssamios.com/dokuwiki/should_we_do_wip_limits_even_though_we_are_doing_scrum

Last update: **2023/03/10 05:29**

