

Table of Contents

How Do We Forecast Our Plan With Kanban?	3
Notes	3

How Do We Forecast Our Plan With Kanban?

I've been at a number of places where people think they should choose Scrum over Kanban simply because they think there is no ability to forecast and plan when a team(s) uses Kanban. While there is a general notion that Scrum is a good approach when you are doing more “plan-able” work [What's The Difference Between Kanban and Scrum?](#) this does not mean that Kanban cannot be used when a forecast is required.

Kanban is an empirical practice, and so generates data in the normal course of work that will allow you to prepare forecasts.

Notes

For Kanban we could use cycle time to understand expectations and compare this to upcoming known list of work. Simplistically, for features (same ideas could apply to stories):

Say our project is expected to have 50 features, and we have completed 5 features with a cycle time of 20 days, 25 days, 30, 27 and 21 days, then we know the following information:

1. At our best rate of completion, then the remaining 45 items (assuming normal rules of the game where features are sized at < 1 quarter of work) will be complete in 900 (45 x 20) days. Based on work I've done in the past there less than is a 10% chance of this happening but we can produce “better” view of this probability by doing a simple Monte Carlo analysis.
2. At our average rate of completion (25 days), we will complete the work in 1125 (45 x 25) days. This has a 50% chance of happening. We are reasonably comfortable about this estimate as we are using an average which, pretty much by definition is a 50/50 proposition.
3. If we want a more predictable view of the date, we would could look at the highest cycle time, in this case 30 days. Time to complete remaining work is 1350 (45 x 30) days. We can be reasonably confident this will happen so call it a 85% chance. Again Monte Carlo modeling would improve this.

You might be wondering whether 5 “observations” is enough to give us good data. Basically the idea is that if you have 5 observations like this, then the probability that the next cycle time is beyond the range we already have is 25% (ie chance that we already have all the cycle times that we will actually produce is 75% - see [How Can We Forecast When We Do Not Have a Lot of Data?](#) for thinking process.) In other words, you don't need a lot of data here.

Note that the process here to generate the forecast does not require that we have all stories detailed etc. It requires:

1. Features are roughly same size (greater than 2 weeks less than 1Q work)
2. We guess the right number of features in the project

Naturally we would check these assumptions as we work. In particular we would look at things like effect of dependencies, effect of increasing knowledge (usually more work than expected) and so on.

Last update:

2023/03/10

05:29

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

[FAQ](#), [Kanban](#), [forecast](#), [metrics](#), [throughput](#)

From:

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

Permanent link:

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

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

