On Measurements – applied actionable agile metrics for predictability – Part I

ontheagilepath.net/2018/12/on-measurements-applied-actionable-agile-metrics-for-predictability-part-i.html

By Sebastian Radics

With this post I'm going to share my insights when working with Actionable Agile metrics. It is the start of a series of posts dedicated to measurements applying Daniel S. Vacanti's [1] approach on measurements in agile environments.

Today let's have a deeper look at the cycle time scatterplot diagram.

Background for the Cycle time scatterplot diagram

Let me first describe some background. Some months ago I supported in splitting a team of 10 people in two teams five members each.

Expectations where that the teams overall delivery will improve. Based on the assumptions that there will be less communication overhead, faster decisions, more fun and motivation.

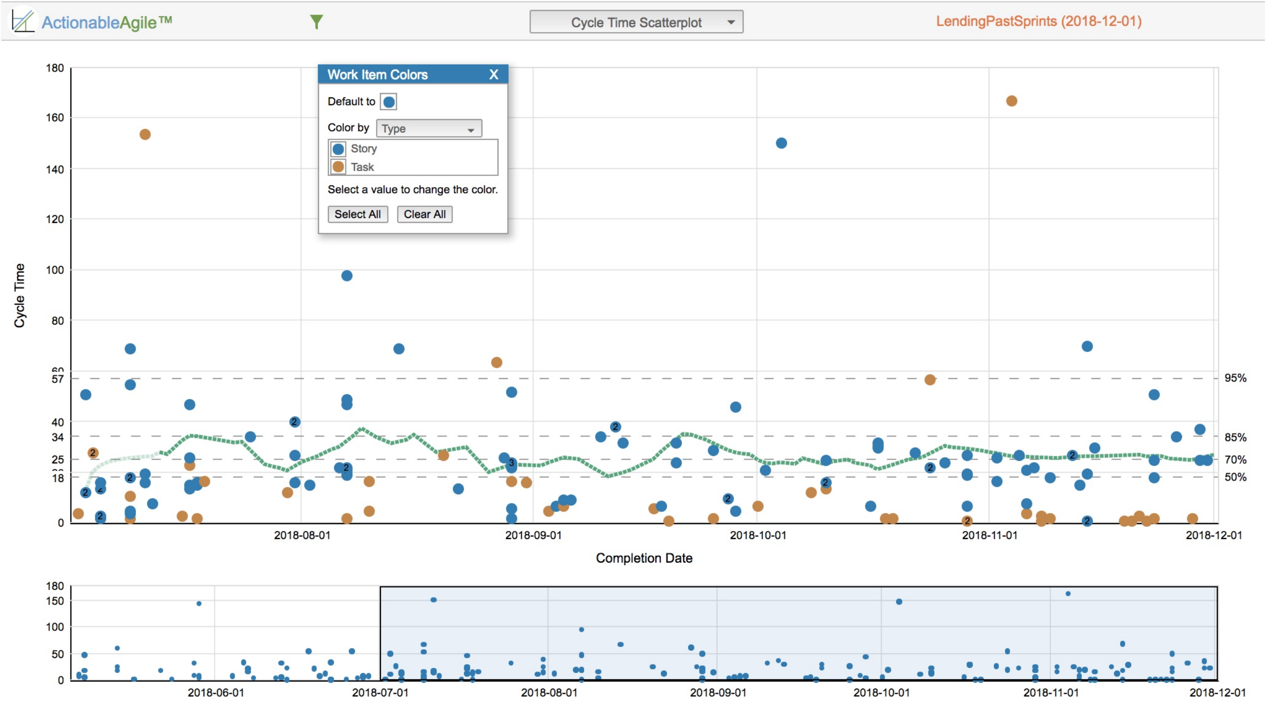
So the measurement needed has to provide data to take a decision whether the team split worked and the expected faster and more reliable delivery is reached or the change (maybe) failed or at least did not lead to expected results.

In addition we had a low predictability of the team's delivery. Sprints sometimes closed with 12 finished items, the other time with 0 finished items. So an additional goal of the team split was to improve predictability. And with more predictability to enable proper delivery date forecasting.

Let's welcome the cycle time scatterplot diagram on stage. Looking for years to finally use that visualization of a team's cycle time development, now I got the chance to connect the dots and install the Actionable Agile metrics [2] plugin JIRA.

Fundamentals about Cycle Time Scatterplot diagrams

Before going into concrete examples I'll explain fundamentals of working with the cycle time scatterplot diagram. To get a visual impression please have a look at the following picture.



Default to ●

Color by Type

Select a value to change the color.

What is that diagram about?

It shows the development of cycle times for items mapped on a timeline. Cycle time in that sense describes the timespan between an items arrival in our process and its departure. It implies that only completed items are shown [3].

In addition the diagram shows percentiles (dashed lines) that provide an easy to grasp overview on the distribution of all cycle times. It does not take averages in to consideration and values the fact that the cycle time distribution does not have to follow a normal distribution [4].

Percentiles by example

The 85% percentile shows the maximum cycle time for 85% of all items. Let’s assume the 85% percentile is 30 days. 85% of all items where completed within 30 days. Phrased differently one can say that an item was being processed between O and 30 days with a probability of 85%.

Seen this way one can use it as input for forecasting delivery dates and when observed over a time span as an indicator for delivery behavior changes.

It gets even more interesting when we consider more percentiles e.g. the 50% and 95% percentiles in relation to our 85%.

Wide distances between percentiles e.g. between the 50% and 85% can indicate problems with predictability. Lets for example assume the 85% percentile shows 30 days and the 50% percentile shows 10 days. So the distance between 85% and 50% is 20 days.

It means that you can for one item assume that there is a 50% probability that it is delivered between 0 and 10 days and a 85% probability that it can be delivered between O and 30 days. And this is really a huge difference.

The goal of improved predictability could be to get the 50% and 85% percentile more closer or understand the underlaying pattern and introduce groups if items with similar predictability patterns.

An additional perspective can be taken when looking at the percentile development over time. Let’s ask questions that provide valuable insights about predictability development: How did the 85% percentile develop in a 2 months timeframe? How did the distance between 50% and 85% develop? We will explore these questions next.

This was just a brief explanation for the cycle time scatterplot diagram. I highly recommend getting into much more details by following the sources listed at the end of this post.

The cycle time scatterplot applied to the team split

Let’s apply the theory on the concrete example of the team split.

Assumptions

Assumptions that indicate a successful team split.

- The cycle time for items decreases as the necessary amount of syncs is reduced
- Predictability of item delivery improves. This improvement is partially connected with the team split as this enables faster process improvements. In addition every team member is better connected with the item delivery. Potential social loafing is reduced and visibility on individual contribution is increased [5].

Timeframes to consider

In order to observe changes it is helpful to consider different time frames. The first one spans the time of some months before the team split until the team split and shows the „old” team’s cycle time development.

The 2nd time frame spans the time from the 1st sprint in two teams until today and shows the cycle time development with both teams. It shows the time frame after the team split.

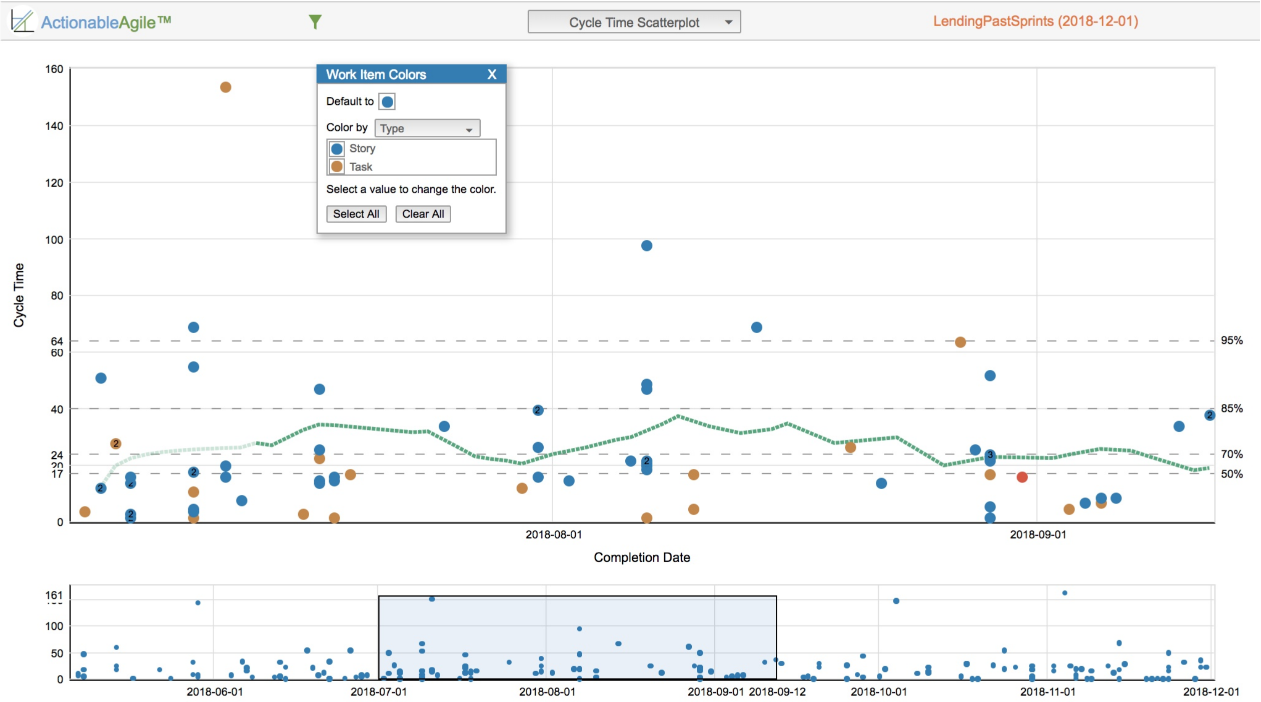
The 3rd time frame starts with start point of frame one and ends with end of frame two and covers the complete range. Based on this frame one can see the overall development, spot patterns and visualize by zooming out.

Interprete with What – So What – Now What

To analyze the data I apply the liberating structure *What? -So what? – Now what?* [6] It helps to separate my observations (What?) from my data interpretation (So What?) and my conclusions (Now What?).

Timespan 1 – Before the team split

Let's look at each graph, starting with timespan span one before team split.



Cycle Time scatterplot that shows the cycletime development in the timeframe before the team split

A brief overview

- Dashed lines show the percentiles (right side is the percentile name; left side the percentile value in days).
- The dots show individual items. Blue dots show completed stories, orange dots show completed tasks. If a dot contains a number it combines several items that where completed on this day with the same cycle time.
- The horizontal axis shows the timespan.
- The vertical axis the cycle time in days.
- Below the diagram one can see the timespan zoom from 1st of July until 12th of September (team split happened on 12th of September and my influence in that team started on 1st of July)

What?

- 85% percentile = 40 days (there is an 85% probability that an item can be completed between 0 and 40 days … as 85% of all items where completed within that range so far)
- 95% percentile = 64 days
- 70% percentile = 24 days
- 50% percentile = 17 days
- spread 50-85% = 23 days
- spread 85-95% = 24 days

There are some extreme outliers too with a maximum of 160 days cycle time. A trend is not visible.

I can observe gaps on the timeline when no items get completed and ares where many items get completed within a short timeframe.

So what?

The wide spread of 23 days between 50-85% percentile strongly impacts predictability. The message that it takes between 0 to 40 days to deliver an item in comparison to up to 23 days for 50% of the items does provide a tendency to either over or under commit to the customer. For me it just feels strange to communicate the 85% range knowing that for many of the items we deliver much faster.

The gaps and bulks of items completion are an indicator for batch processing of items. As we work with sprints it can be connected with a typical sprint closing pattern where in the beginning of a sprint, things are started to be processed in parallel and nothing is completed for some days. Toward the sprint end items are completed and delivered.

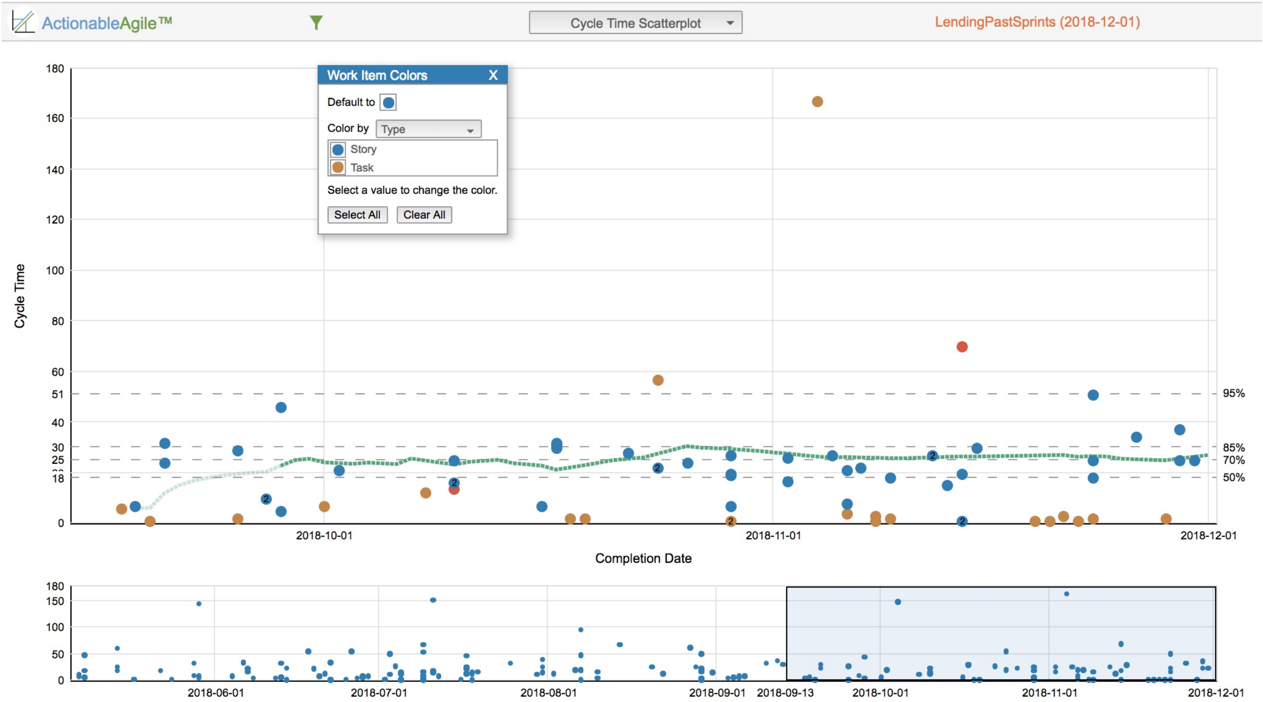
This kind of behavior to start many things in parallel is also supported by having bigger team sizes and it tends to create sub groups that work (maybe) together on an item.

Now What?

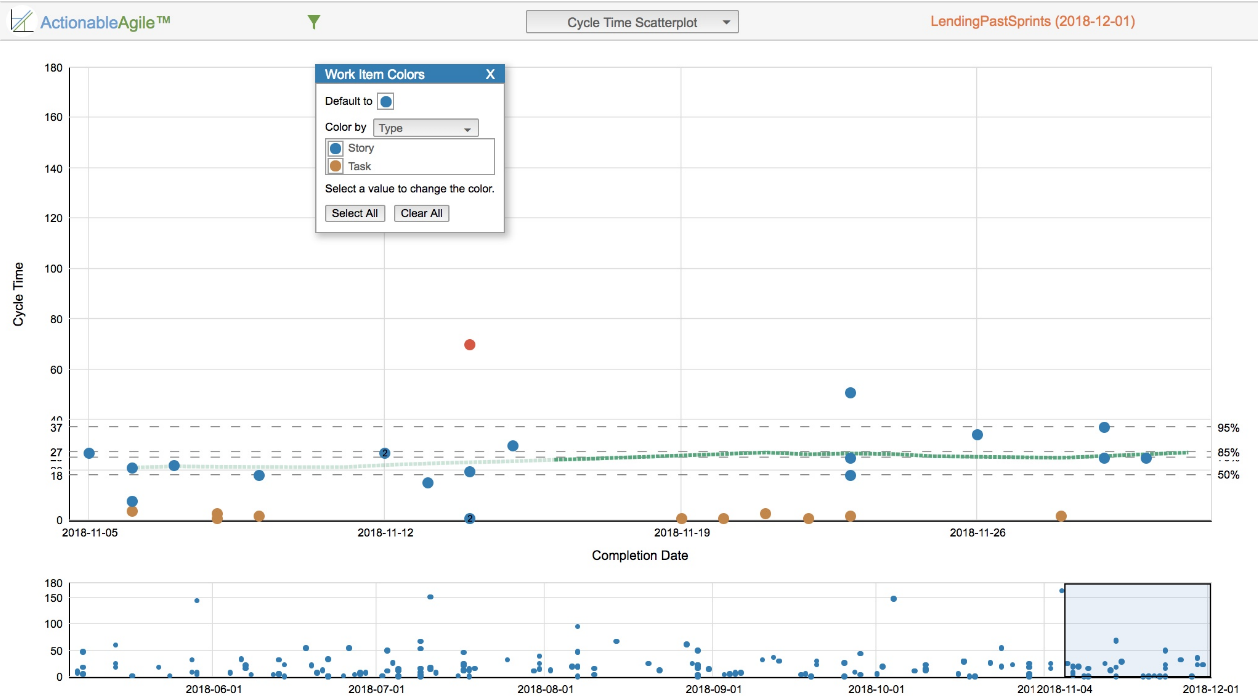
Based on that data and on concrete daily working behavior observations and reflections in the teams retrospective we derived splitting the team in two teams.

Timespan 2 – Predictability development after the team split

Let's have a look on the cycle time scatterplot after the team split.



Full timespan after team split



What?

For the full timespan after the team split:

- 85% percentile = 30 days
- 95% percentile = 51 days
- 70% percentile = 25 days
- 50% percentile = 18 days
- spread 50-85% = 12 days
- spread 85-95% = 21 days

and just considering the last month:

- 85% percentile = 27 days
- 95% percentile = 37 days
- 70% percentile = 25 days
- 50% percentile = 18 days
- spread 50-85% = 9 days
- spread 85-95% = 10 days

So What?

We improved predictability. The distance between the percentiles (e.g. 50-85%) decreased from previously 23 days to best 9 days. Our 85% percentile went down from 40 days to 27 days. Now we can communicate that it takes up to 27 days to finish an item with 85% probability (what previously took up to 40 days). An improvement of 32,5%.

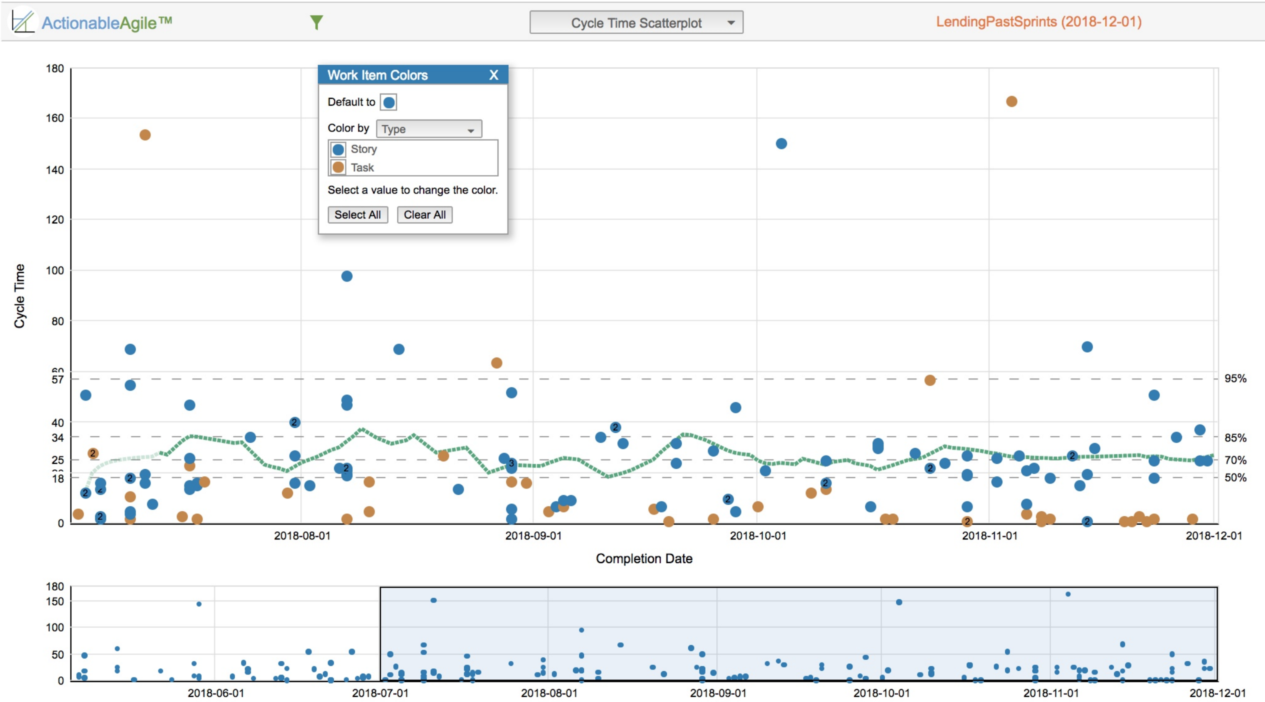
Now What?

Better predictability improves forecasting delivery dates. I'll explain more on that in one of the next blog post.

We can conclude that the team is able to finish items faster. Whether this is solely a result of the team split or overlapping with other improvements like story quality, providing more context, enhancement in communication is from systemic perspective tough to separate. And we did not choose an approach to apply just one change at the time and look at the results for a longer period. But the team delivery change correlates with the team split timeframe and that is a strong indicator.

As a result we have a really improved predictability. In addition the cycle time decreased and at the same time not yet for the 50% percentile. We need to further investigate how we can slice items even further to improve on the cycle time.

Timespan 3 – Beginning to End – global perspective



Default to ●

Color by Type

Select a value to change the color.

What?

- 85% percentile = 34 days
- 95% percentile = 57 days
- 70% percentile = 25 days
- 50% percentile = 18 days
- spread 50-85% = 16 days
- spread 85-95% = 23 days

So What?

Longer term we can see a reduced spread for 50-85%. Reduced by 7 days, 30 percent. Also the cycle time went down to up to 34 days with 85% probability. An improvement of 6 days, 15%.

Now what

Also looking on longer term we can confirm the improvements.

Conclusion

For analyzing the cycle time development over a timeline the cycle time scatterplot diagram provides insights. We were able to derive an improved cycle time and an improved predictability. By working with the visualization of percentiles we do not wrongly applying averages. Considering different time spans (with the plugin in an easy way) enables a fast comparison of the cycle times and it's changes.

In further articles we will explore how the teams throughput developed. In addition we'll have a look on delivery date forecasting. Now that we know that our predictability improved, we can build on top of that with the forecasts.

Further steps from here

- Scan https://actionableagile.com/ for exploring the Actionable Agile Metrics plugin. Please also check the great books by Daniel S. Vacanti.
- Check out all my blog posts about measurements
- And share your comments, thoughts and experiences

1. https://actionableagile.com/about-us ↩
2. https://actionableagile.com/ ↩
3. It is important to consider because often in the beginning of process improvements rather old items can get closed and potentially provide a picture that can be interpreted the wrong way! ↩
4. Please check for a more detailed explanation on the flaw of average when considering cycle time Daniel S Vacanties book Actionable Agile Metrics for predictability ↩
5. Please lets not connect that with an outside monitoring but as a way to individually sense ones impact on the product. ↩
6. http://www.liberatingstructures.com/9-what-so-what-now-what-w/ ↩