

Visualize sprint progress – an alternative to burndown charts

 ontheagilepath.net/2012/11/visualize-sprint-progress-an-alternative-to-burndown-charts.html

By Sebastian Radics

From traditional burn-down-charts to the sprint flow

When I started with the Scrum implementation I followed descriptions from Scrum books and the Scrum Master training and started with a “traditional” burndown chart.

In the beginning we used the hour-based burndown chart, followed by the story based burndown chart. Today we use an artefact – we call **Sprint Flow** – a kind of a Gantt chart, but with more sprint related information.

BBQFTX43DQ8T

To set the stage – what do I need and use this artefact for

- visualize the sprint progress and see on a daily base where we are in the sprint (and adjust sprint strategy if necessary)
 - for the team including the product owner – used during the daily scrum
 - for stakeholders and other interestest people passing by during the sprint
- input for the sprint retrospective to have a progress representation as input for inspect & adapt (what went well? what to improve?)

Requirements to make it useful are

- **easy to maintain on the daily base**
- **highly visible and usable during the daily standup**
- **progress information easy to understand with less room for interpretation**

Let’s consider the different ways for displaying the sprint progress based on an example

Example overview

Team X had a 2 week sprint and committed to implement 5 stories:

story	estimation
-------	------------

I	8
II	5
III	8
IV	5
V	3

Starting with the hour-based burndown chart

In illustration 1 you can see an example for an hour based burndown chart. The team has in total a capacity of 300h available for the sprint. And after sprint planning II they had a count of 260h to implement the committed stories.

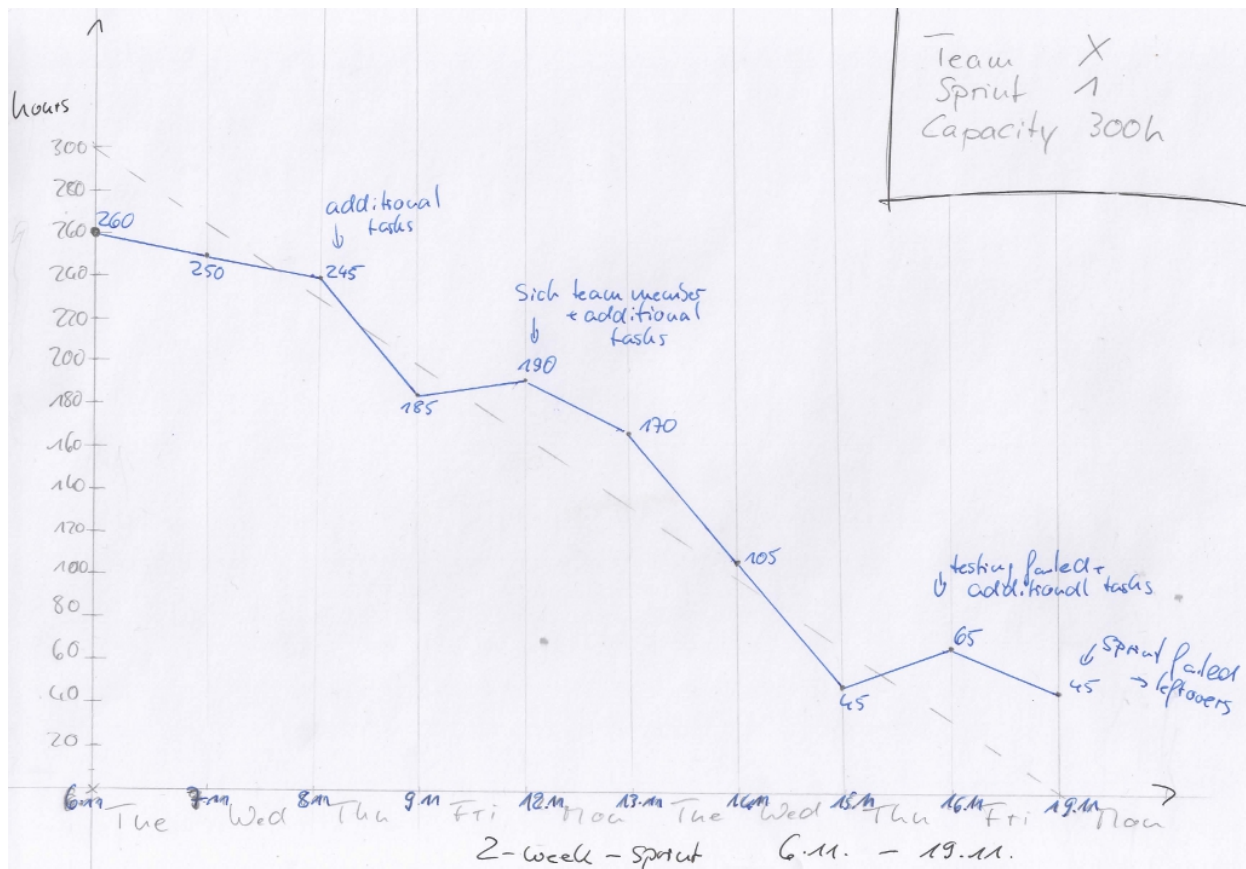


illustration 1 – hour based burndown chart for a 2-week sprint

During the sprint various new tasks (not stories) were added (what I think is usual) and there was a sick team member.

Problems with this chart

- How many stories are finished? You can't answer this main question from the chart. It looks not that bad – but it could be that no story at all was finished...
- To draw the chart line daily – you need to count the hours burned and added daily. At least you need some way of doing this calculation on your scrum board (what I still favour over tools) – This counting is time consuming
- On the 15.11. all looked fine – sprint committment looked like achievable – but exploded at the end. At least from this chart it was not visible at all

Mapped to the given requirements:

- it delivers input for the retrospective (and can be enhanced with event bubbles – see below)
-
- maintainance efforts are higher compared to the other variants described below
- it's visible if you put it on a flipchart paper in your team area. But it's not that easy to embedd it in you daily scrum flow because you need to count the hours and you won't wait with the whole team for the counting. A meeting after the daily scrum to discuss the progress is from my opinion not really an option.
- it leaves a lot room for interpretation. You don't know really where you are in the sprint.

I don't use this kind of chart any longer. It got replace by the story burndown chart.

Followed by the story-based burndown chart

In illustration 2 you see an example of an story based burndown chart. You draw the line down as soon as a story is finished. In this chart we also used bubbles to note down noteworthy things that happend during the sprint (as input for the next retrospective).

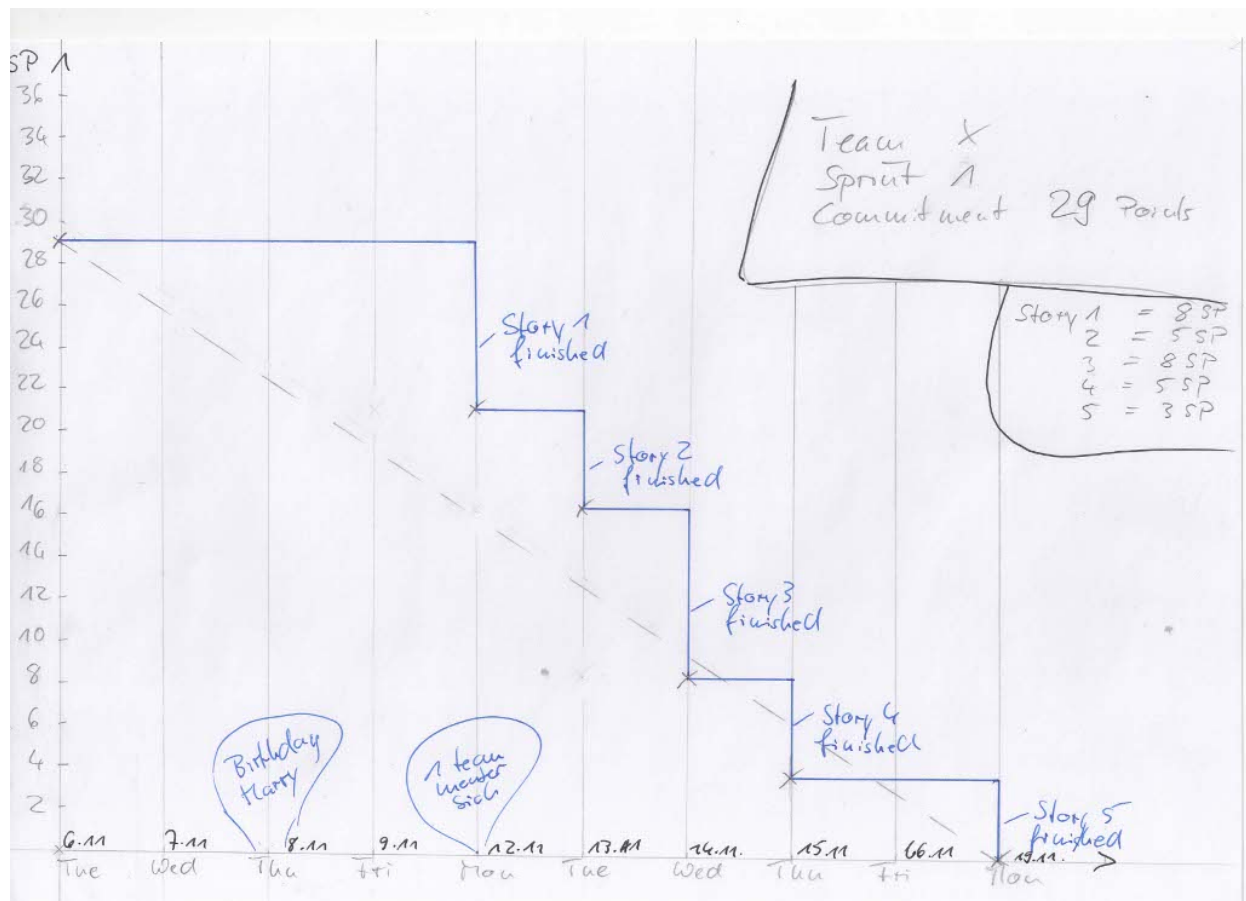


illustration 2 – example of a story based burndown chart + event bubbles for a 2 week sprint

The team took a bit longer to finish the first story (sick team member, story a bit underestimated, stories started in parallel). Working in parallel on stories explains why story 2+3 got finished that fast (compared to story 1).

Advantages compared to the hour based chart

- focus on the main deliverables – the stories. You know how many stories are finished and are left to be finished
- much faster to update – as finished stories lead to a burndown, otherwise the line remains straight.
- can be better integrated in the daily scrum – whole team can update it and can discuss actions necessary
- lesser room for interpretation

There are still some problems with this chart

- it's only useful if you more than 4 stories in your sprint and stories are not too big (what's anyway the recommendation, but we encountered sometimes problems with it ... then this chart is not that helpful)

- if you work on stories in parallel the line remains straight for some days and you can't see the progress. During these days the progress is not easy to understand and it opens room for interpretation.

Mapped to the requirements:

- it delivers input for the retrospective and visualizes the sprint progress on the daily base (story based)
- it is easy to maintain
- you can embed it in your daily standup (better)
- progress information for stories is available but leaves room for interpretation.

What would happen if on the last sprint week 2 people planned holiday?

What if this time there are too many testing tasks and your tester is overloaded?

To answer these questions we used something to discuss in which order we start working in the stories, when we start testing it. It was a kind of a prototype of our currently used sprint flow. But at this time we did it in parallel to the burndown chart.

To remove this duplicated work – we introduced what we call the **sprint flow** that combines the information.

And now we use the sprint flow

We discuss and set it up at the end of the sprint planning I. It allows us to validate our commitment by including the availability of everyone and considering public holidays or necessary release dates. The frame (flipchart paper) stays the same, so you only have to put new stories (printed from JIRA and fixed with sticky tape) and sticky notes on it.

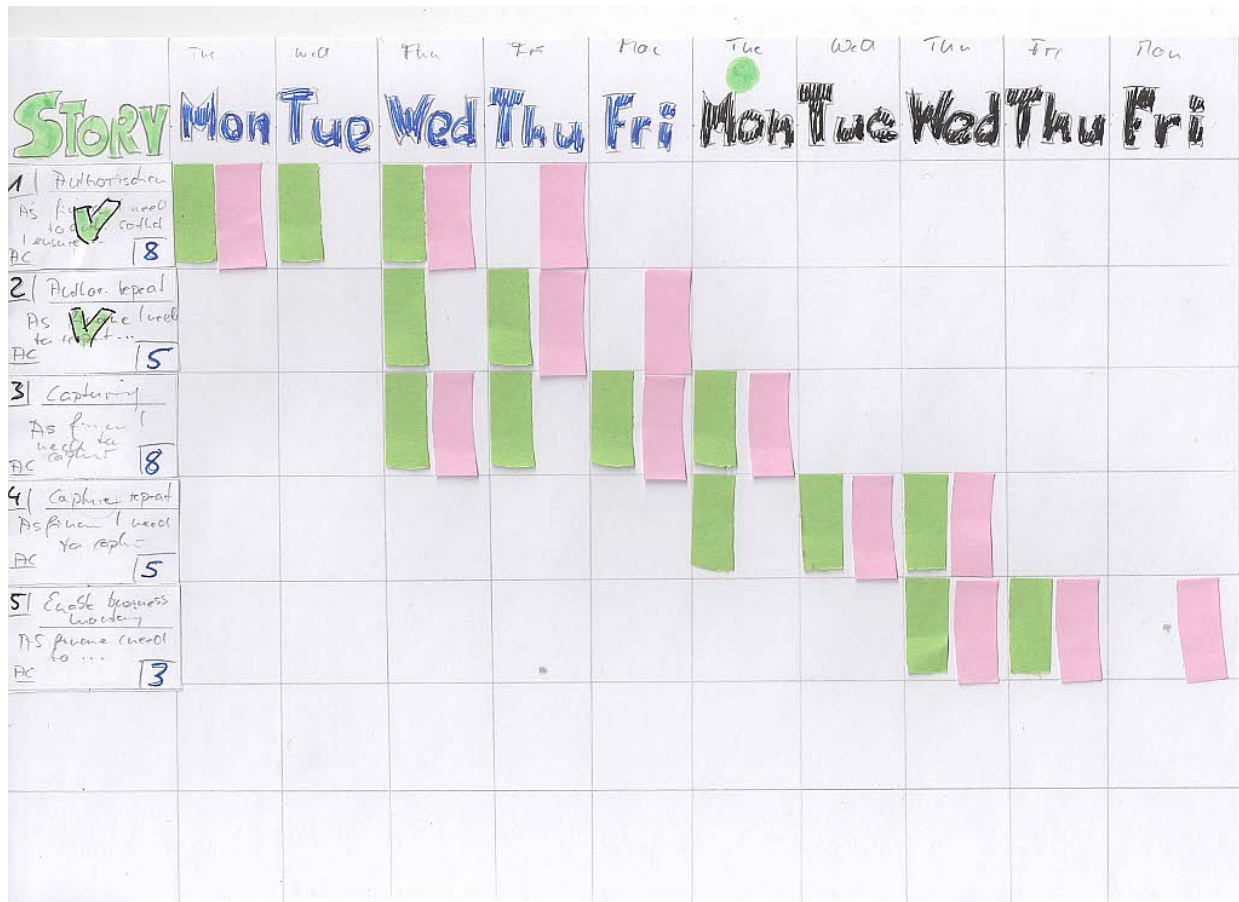


illustration 3 – example of a sprint flow for a 2 week sprint

The color coding is to distinguish between testing efforts and implementation efforts (green = implementation, pink = testing, done by our tester in the team).

We use it during the daily standup and check if we are still on track with stories like planned. As soon as we discover that we need more time on a story, we adjust the sticky notes (sometimes you need to arrange some more sticky notes, but it implies that you need to think properly how the rest of your sprint will work out).

When a story is finished we mark it as finished in the first column (in the picture I assumed we are on the 2nd monday – green dot – and 2 stories are finished). On this day it shows that we are still on track (as there is still some room on friday and thursday).

Advantages compared to the story burndown chart

- even easier to maintain
- includes public holidays and roles better
- shows work in parallel

Mapped to the requirements:

- it's highly visible and provides input for the retrospective (as it shows the last state how you worked on it)
- it's easy to maintain and nice to integrate in the daily scrum
- progress is easy to see, there is still room for interpretation but less than in the previews ways
- it removes the "hidden" thinking about how to tackle the sprint and embeds it in the sprint planning meeting.

Can you help me?

- How is your experience using sprint progress visualizers?
- What do you use?
- Do you agree with the problems outlined above?
- How do you embedd using it in your daily standup?