BURNDOWN CHARTS

Outcome

A burndown chart is a graph that shows how the amount of work remaining to be completed changes over time.

Function

This technique helps a team track and makes visible the progress of work against an estimate. A regularly updated chart is a source of early feedback and allows the team to change and adapt accordingly.

A burndown chart is most useful when it's displayed on a wall or whiteboard so that it is visible to the whole team. At relevant intervals the team plots the estimate of the work remaining on the chart. The work remaining is thus 'burnt down' towards zero.

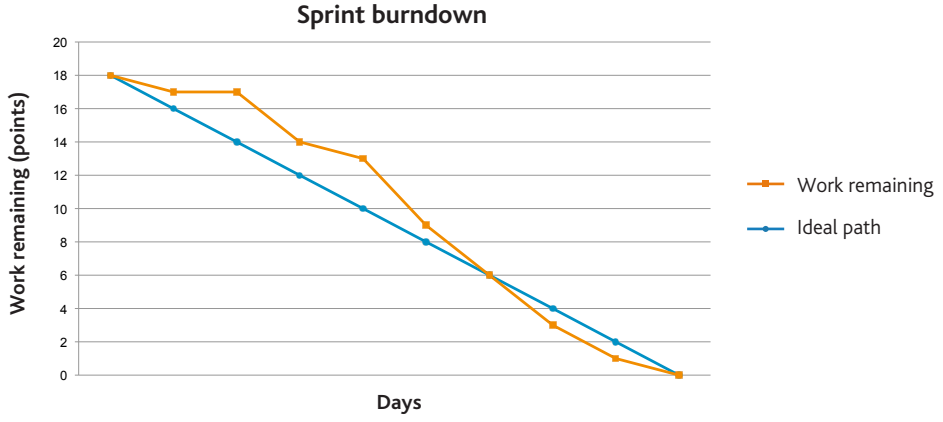
Burndown charts are most commonly used in Scrum, where two types of the chart track different periods of time:

- A **Sprint** burndown tracks progress over the course of a specific Sprint (iteration), so the time period covers the length of the Sprint. The chart is updated daily.
- A **Release** burndown tracks progress over multiple Sprints against a release plan. The time period is therefore longer and the chart is usually updated at the end of each Sprint.

Benefit

A useful addition to the burndown chart is an 'ideal path' - a line indicating a predicted ideal amount of work remaining over time. It is drawn as a straight line from start to finish. Although a team will not typically burn down their work in a linear fashion, this additional line can help to highlight whether the team is significantly ahead or behind of schedule. A sample Sprint burndown chart is shown below.

Who



Scaling Factors

Difficulty

Implementation

Prerequisites

There are a few prerequisites before a team can create a burndown chart:

- There must be an agreed period of time over which progress will be tracked.
- The scope of the work to be completed is outlined. For example, in the form of a Sprint commitment.
- All work planned to be delivered in the agreed time period has been estimated.

Agile teams often represent work (requirements) in the form of user stories that can be broken down further into individual tasks. The estimates for the user stories can be given in relative points. For the burndown chart to be meaningful the work remaining should be tracked in the number of relative points remaining and only updated when a particular story is considered complete. An explicit definition of done for each story lets the team know when work is completed.

To create a **Sprint burndown chart**:

1. Get a flipchart or a large piece of paper that you can put up on the wall.
2. Draw a horizontal line for the time from the beginning to the end of the Sprint, create a scale in days.
3. Draw a vertical line for the work remaining, in relative points (or whatever unit you are using for your estimates). It should go from zero to slightly more than the sum of the estimates for the individual user stories (or requirements).
4. Plot the team's starting point: a dot or cross showing how much work there is to be done at the beginning of the Sprint.
5. Every day, the team plots the work remaining on the chart. The team does that by adding estimates of all outstanding user stories.

A **Release burndown chart** has the following differences:

- At step 2: each interval on the horizontal line should be a completed Sprint.
- At steps 3 and 4: the work remaining should include the whole release.
- At step 5: the chart should be updated at the end of each Sprint.

Potential pitfalls

- Be careful not to just burn down time spent: '90% done' is not done. This can particularly be an issue if your estimates are in hours (rather than, for example, in relative points). Avoid this by creating an explicit definition of done and only burning down the estimate for each item when it meets the definition of done criteria.
- Don't hide the discovery of new work. Make this visible on the chart by re-plotting the work remaining. You can also add a note on the chart so that it is clear afterwards why the chart has spiked up.

If you want to learn more, consider reading:
Agile Estimating and Planning by Mike Cohn