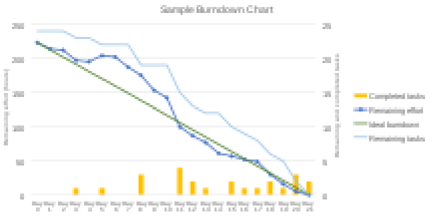
Burn down chart

A **burn down chart** is a graphical representation of work left to do versus time. The outstanding work (or backlog) is often on the vertical axis, with time along the horizontal. Burn down charts are a run chart of outstanding work. It is useful for predicting when all of the work will be completed. It is often used in agile software development methodologies such as Scrum. However, burn down charts can be applied to any project containing measurable progress over time.

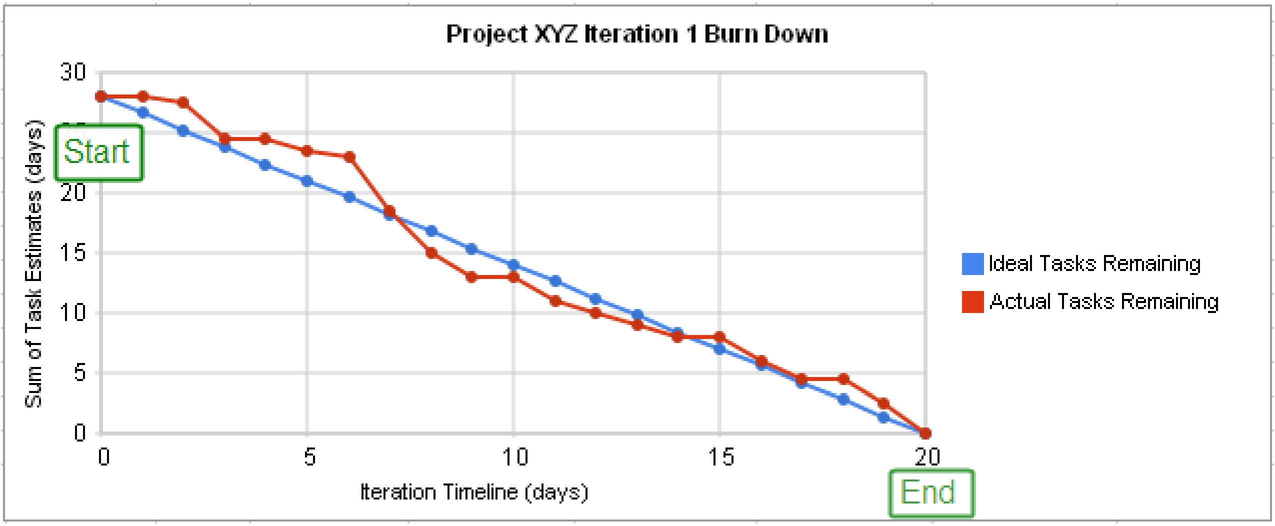


A sample burn down chart for a completed iteration, It will show the remaining effort and tasks for each of the 21 work days of the 1-month iteration.

Outstanding work can be represented in terms of either time or story points.[1]

Contents

Reading burn down charts



A project burn down chart

A burn down chart for a completed iteration is shown above and can be read by knowing the following:[2]

X-Axis	The project/iteration timeline
Y-Axis	The work that needs to be completed for the project. The time or story point estimates for the work remaining will be represented by this axis.
Project Start Point	This is the farthest point to the left of the chart and occurs at day 0 of the project/iteration.
Project End Point	This is the point that is farthest to the right of the chart and occurs on the predicted last day of the project/iteration
Number of Workers and Efficiency Factor	In the above example, there are an estimated 28 days of work to be done, and there are two developers working on the project, who work at an efficiency of 70%. Therefore, the work should be completed in $(28 \div 2) \div 0.7 = 20$ days.
Ideal Work Remaining Line	This is a straight line that connects the start point to the end point. At the start point, the ideal line shows the sum of the estimates for all the tasks (work) that needs to be completed. At the end point, the ideal line intercepts the x-axis showing that there is no work left to be completed. Some people take issue with calling this an "ideal" line, as it's not generally true that the goal is to follow this line. This line is a mathematical calculation based on estimates, and the estimates are more likely to be in error than the work. The goal of a burn down chart is to display the progress toward completion and give an estimate on the likelihood of timely completion.
Actual Work Remaining Line	This shows the actual work remaining. At the start point, the actual work remaining is the same as the ideal work remaining but as time progresses, the actual work line fluctuates above and below the ideal line depending on this disparity between estimates and how effective the team is. In general, a new point is added to this line each day of the project. Each day, the sum of the time or story point estimates for work that was recently completed is subtracted from the last point in the line to determine the next point.

Measuring performance

Actual Work Line is above the Ideal Work Line	If the actual work line is above the ideal work line, it means that there is more work left than originally predicted and the project is behind schedule.
Actual Work Line is below the Ideal Work Line	If the actual work line is below the ideal work line, it means that there is less work left than originally predicted and the project is ahead of schedule.

The above table is only one way of interpreting the shape of the burn down chart. There are others.[1]

Removing variability in time estimates

One issue that may be noticed in burn down charts is that whether or not the Actual Work line is above or below the Ideal Work line depends on how accurate the original time estimates are. This means that if a team constantly overestimates time requirements, the progress will always appear ahead of schedule. If they constantly underestimate time requirements, they will always appear behind schedule. This issue is corrected by incorporating an efficiency factor into the burn down chart. After the first iteration of a project, the efficiency factor can be recalculated to allow for more accurate estimates during the next iteration. Some templates automatically calculate the efficiency as a project progresses. This can be used to identify areas/phases where inaccurate estimates consistently occur.[2]

References

1. "Feel The Burn, Getting the Most out of Burn Charts - Better Software, Volume 11, Issue 5, pp. 26-31" (http://idiacomputing.com/pub/BetterSoftware-BurnCharts.pdf) (PDF). July–August 2009.
2. Wenzel, Joel (December 2012). "Burn Down Chart Tutorial: Simple Agile Project Tracking" (https://web.archive.org/web/20130707200105/http://joel.inpointform.net/software-development/bur

n-down-charts-tutorial-simple-agile-project-tracking/#). Archived from the original (http://joel.inpointform.net/software-development/burn-down-charts-tutorial-simple-agile-project-tracking/) on 2013-07-07. Retrieved 2011-06-14.

Further reading

- Release Burndown (https://web.archive.org/web/20071101053001/http://www.mountaingoatsoftware.com/release_burndown)
- Burndown Chart Example Patterns (https://zepel.io/agile/reports/burndown/)

Retrieved from "https://en.wikipedia.org/w/index.php?title=Burn_down_chart&oldid=965798348"

This page was last edited on 3 July 2020, at 13:42 (UTC).