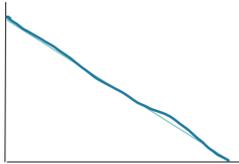


Burn-Down Patterns

matthiasorgler.com/burn-down-patterns

Patterns visible in sprint burn-down charts can tell you a lot about your team. Here are a few typical patterns and what you can learn from them. This is by no means an exhaustive list.

Burn-Down Chart

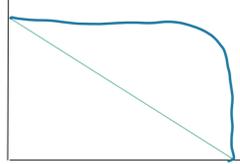


You're gaming the system!

It's unlikely that you really make exactly the same progress each day. You might make the numbers fit the expectations out of fear or pressure.

Fix: Be honest to yourself. This burn-down chart is supposed to help you, not to punish you.

Burn-Down Chart



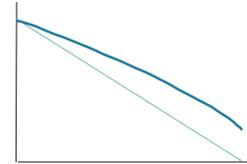
Accepting too late

The Product Owner accepts stories only at the end of the sprint. She is either not available during the sprint, doesn't accept stories on a daily basis or overrules decisions of a proxy-PO.

The problem: If stories are rejected, there is no time left to fix them in the sprint. The team lacks a timely feedback on how much is really done, thus depriving them of the chance to detect and fix problems early on.

Fix: Accept stories immediately. You could do this as "in-sprint inspection" right after your Daily Scrums.

Burn-Down Chart



Ambitious team

You might have bitten off more than you could chew.

Fix: Learn the lesson for your next planning and commit to fewer user stories.

Change of priorities

A critical issue interrupted your plans. You had to fix this first.

Fix: Try to avoid such interruptions. If priorities changed significantly, consider aborting the sprint.

Submissive team

You let yourself be pressured into an unrealistic schedule.

Fix: Speak up next time and take responsibility for your estimations.

External dependencies

You have dependencies outside of your sphere of influence.

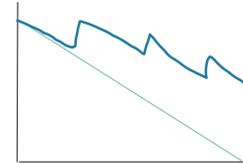
Fix: This has to be fixed on an organisational level. Make sure the organizational structure fits the product or service architecture. Sometimes a restructuring of teams or a refactoring of the architecture can help.

Capacity changes

Team members get sick or leave the team.

Fix: Not much you can do. Increase the resilience of the team by developing T-shaped characters.

Burn-Down Chart



Dynamic backlog

Urgent requirements are added to the sprint backlog mid-sprint. At the same time the original scope is not reduced.

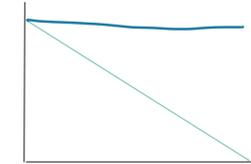
Fix: Consider a sprint backlog immutable by default. If an urgent task comes in, remove other stories from the sprint backlog. If your current sprint goal becomes obsolete by drastically changed priorities, consider cancelling the sprint.

Bad refinement

You failed to refine user stories accurately. Not well understood requirements lead to trouble during implementation.

Fix: Refine and estimate thoroughly. Ask the Product Owner enough questions to fully understand the requirements. If you lack knowledge in the business domain or technology used, consider training.

Burn-Down Chart

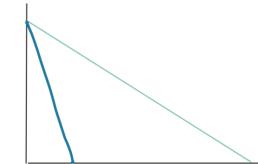


Burn-what?

You likely pay no attention to your burn-down. You probably do not split work into small chunks or deliver working functionality frequently.

Fix: Instead of dividing work as it comes at you, try to plan and split work in advance. Watch your burn-down – it is there to help you.

Burn-Down Chart



Excluding work

You might not have all work you do on your backlog. Are you missing QA or bug fixing?

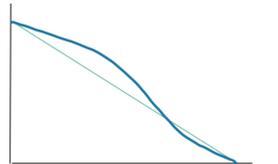
Fix: Include all work for the product in your backlog. Many things can be part of user stories, others might be separate backlog items.

Punished team

Your team might have been punished in the past for not reaching ambitious goals. This can lead to conservative planning with huge buffers.

Fix: Remember that you cannot predict the future – you merely forecast. Reality can be different and it need not be anybody's fault. Don't punish the team for not reaching the sprint goal.

Burn-Down Chart



Good

This is how a chart of a good sprint could look like. It usually starts slow but then speeds up as stories get done.