



PATTERN 02

Hoarding the Code

2. Hoarding the Code

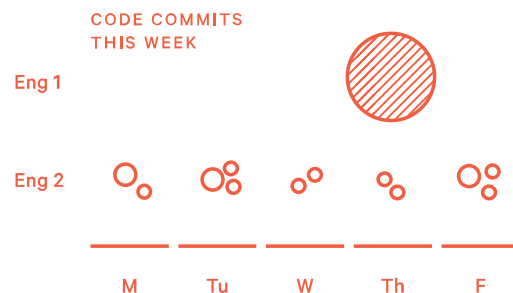
This pattern refers to the work behavior of repeatedly working privately and hoarding all work in progress to deliver one giant pull request at the end of the sprint.

It's not uncommon for programmers to wait until their work is done to share it. In creative and research-intensive fields, it can be a natural tendency to avoid sharing work when it's only just started. There are plenty of reasons why this might be: a fear of having others judge the work in progress, a fear of others taking ideas, or a desire to make the whole process seem effortless, to name a few.

Whatever the reason, the heart of the problem is this: the more an individual saves up their work, the less they collaborate with others. Working alone is inherently riskier than working with others. And software engineering is a team sport.

This tendency to work privately and then submit work all at once doesn't just limit and slow down the individual — it's damaging to the team's progress as a whole. Submitting work all at once means that there weren't any opportunities for collaboration along the way. Even more, once the work was submitted, someone else had to *review* all of that work. So naturally, this work behavior can also lead to lower quality code — both from the Submitter's standpoint (who didn't check in their work early to get feedback or notice potential missteps), and the Reviewer's perspective (who likely doesn't have enough time or energy to adequately review all of that code).

When you see large and infrequent commits, first consider the pattern's **duration** (have we seen this pattern for years, or has it only recently been visible?). This information can help determine whether this is the engineer's preferred way of working, or if this is caused by something outside the normal development process.



How to recognize it

Large and infrequent commits can be a sign that the engineer is working privately until their project is finished, and then submitting their work all at once.

This pattern is typically first seen in the Work Log report but is also identifiable in the team's Review Workflow. These PRs are larger and usually come later in a sprint or project. Because of this, they'll typically either take a longer time to review (relative to the team's average) or will get a lower level of review (see Review Coverage).

It's also common for these engineers to show lower than average Receptiveness in the Submit Fundamentals. When people work in isolation, only submitting it for review once they've decided it's the 'right' solution later in the sprint, it's generally much more difficult to take feedback on that work objectively.

What to do

Above all else, be compassionate. Odds are, you've recognized this pattern right before or just after the end of a sprint, so these engineers are likely tired, stressed, and worn out. Make sure they get the time and space they need to recover from delivering such a big payload.

This can be great timing for an impromptu and informal 1:1. Going on a walk or getting coffee, for example, can keep the conversation casual. Get them talking about their latest project, ask what went well and what didn't, and recognize their achievement.

Along the way, bring up the topic of team collaboration, and how saving work until it's completed leaves little room for learning from others throughout the process. When teams do work together throughout a project, they can learn from each other's perspectives, reduce uncertainty and move faster, and even find improved solutions to the problem. In practice, that might look like submitting work far before the engineer thinks it's ready for a review.