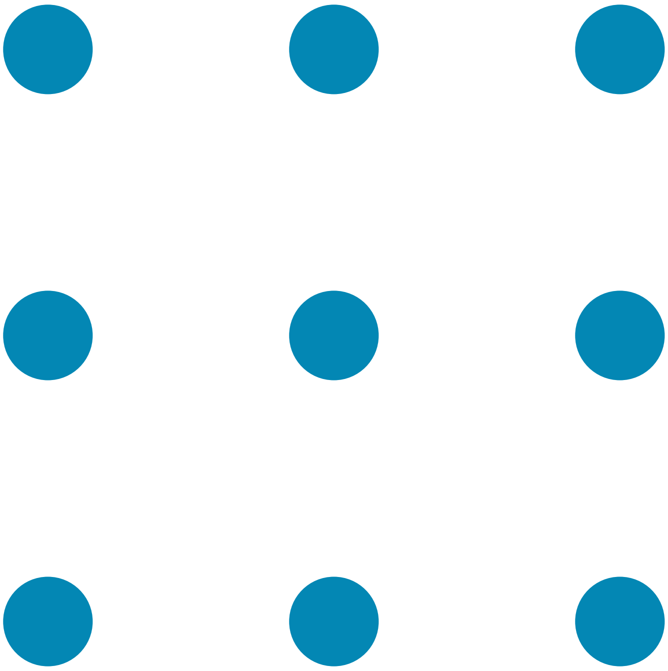
nave

The Ultimate Guide To

READING THE CUMULATIVE FLOW DIAGRAM

How to Recognize the Most Common CFD Patterns and What They Mean for Your Workflow

How Many Dots Do You See in This Picture?



How many dots do you see in this picture? This isn't a trick question.

There are nine dots in this picture. But did you count them one by one or did you know the answer immediately?

You knew straight away. How did this happen? Your brain recognized a pattern. Patterns are all around us. And detecting them is an important part of how humans learn and make decisions. Our brains are very good at doing this. Everything we do is shaped by pattern recognition.

Pattern Recognition & Decision Making

Our ability to recognize faces, understand spoken words, count or read is shaped by pattern recognition. We use patterns to predict. To connect a cause with the most likely effect. By definition, this is a process of matching the incoming stimuli with the information already stored in the brain.

Ever since our birth, we have been treating the information in this manner so that we can take the best course of action. If you see dark and heavy clouds in the sky, chances are it will rain. You'll grab an umbrella on your way out so you don't get soaked. You've made a decision based on the most likely outcome. Recognizing patterns plays an essential role in decision making. That's why this skill has been crucial for our survival.

Of course, predicting the weather is only a basic example. Today, pattern recognition is used alongside data analytics to optimize stock market forecasting (Yard Charts), audience research (Google Analytics), Natural Language Processing (Copyscape, Google Translate, Grammarly), and the practice we're interested in the most - product management.

The Cumulative Flow Diagram: Pattern Recognition and Workflow Optimization

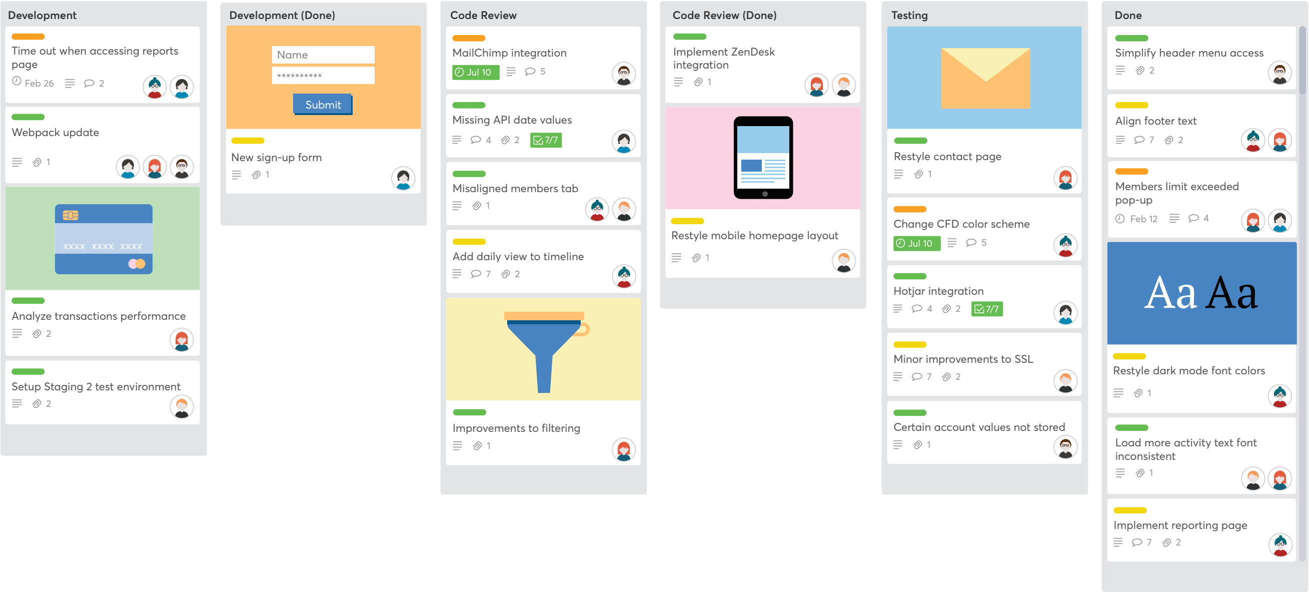
With the visualizing power of the Cumulative Flow Diagram (CFD), our data comes to life and we're able to spot certain patterns to avoid productivity losses, delays, poor predictability and other process inefficiencies.

Where there is data, there are patterns. And these patterns tell the stories through ebbs and flows, flat lines and sudden spikes. You just need to learn how to read them to be able to evaluate your process performance and enable early discovery of problem areas such as:

- **Increasing Work in Progress (WIP)**
- **Process blockages**
- **Productivity losses**
- **Frequent delays**
- **Poor predictability and other flow inefficiencies**

An ideal CFD represents a steady growth of completed work and a continuous delivery of value. But we don't live in an ideal world and deviations are bound to happen. Spotting and reacting to these deviations quickly can help us stay on the right track and better perceive the direction we're moving in. By recognizing some of the most common CFD patterns and understanding what they mean for our process, we can optimize our workflows and focus on resolving bottlenecks as they occur.

Visualizing the Flow of Work on a Kanban Board



Development

- Time out when accessing reports page — Feb 26 — 2
- Webpack update — 1
- XXXX XXXX XXXX
- Analyze transactions performance — 2
- Setup Staging 2 test environment — 2

Development (Done)

- Name — •••••••••• — Submit
- New sign-up form — 1

Code Review

- MailChimp integration — Jul 10 — 5
- Missing API date values — 4 — 2 — 7/7
- Misaligned members tab — 1
- Add daily view to timeline — 7 — 2
- Improvements to filtering — 1

Code Review (Done)

- Implement ZenDesk integration — 1
- Restyle mobile homepage layout — 1

Testing

- Restyle contact page — 1
- Change CFD color scheme — Jul 10 — 5
- Hotjar integration — 4 — 2 — 7/7
- Minor improvements to SSL — 7 — 2
- Certain account values not stored — 1

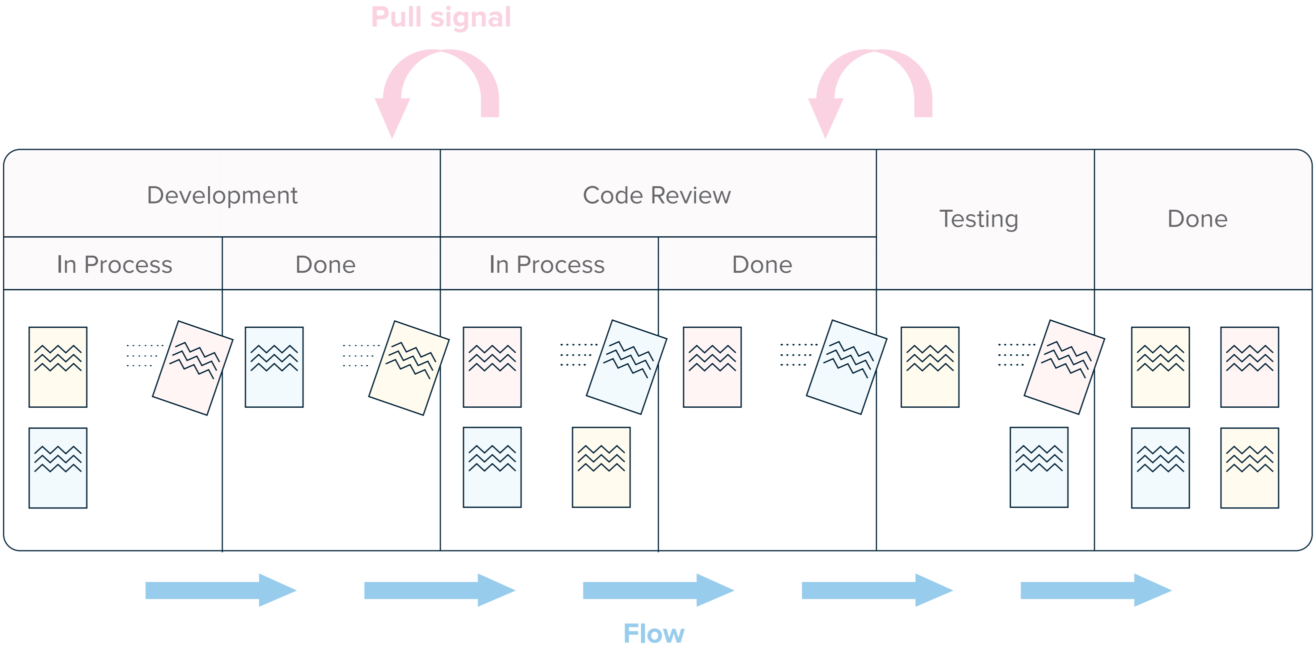
Done

- Simplify header menu access — 2
- Align footer text — 7 — 2
- Members limit exceeded pop-up — Feb 12 — 4
- Aa Aa
- Restyle dark mode font colors — 1
- Load more activity text font inconsistent — 1
- Implement reporting page — 7 — 2

Before we can analyze our data with a CFD, first we need to visualize our process in a Kanban board. Our Software Development process has 4 process states - Development, Code Review, Testing, and Done. There are also two queue states - Development (Done) and Code Review (Done). These are states with tasks that are waiting to be worked on.

A Kanban Pull System

The tasks move across process states when a pull signal has occurred and the team has the capacity to handle new work. A pull signal occurs when a card is placed into a queue state - that's why it's so important to split all in progress steps into active and passive states, that's essential in kanban pull systems. This approach will also enable you to track how much time your work is waiting in your process.



Working with the Kanban board helps you visualize the flow of work but relying on the board alone isn't enough to optimize your workflow. This is where the CFD comes into play. Let's see how we can build one of our own from an existing data set.

Building a CFD from Your Board Data

Method 1 - Listing Your Stories and Their Arrival Dates in Each Process State

Story ID	Development	Development (Done)	Code Review	Code Review (Done)	Testing
1	1 Jul 2019	1 Jul 2019	4 Jul 2019	4 Jul 2019	6 Jul 2019
2	3 Jul 2019	6 Jul 2019	8 Jul 2019	9 Jul 2019	12 Jul 2019
3	4 Jul 2019	10 Jul 2019	12 Jul 2019	13 Jul 2019	15 Jul 2019
4	7 Jul 2019	11 Jul 2019	11 Jul 2019	13 Jul 2019	13 Jul 2019
5	9 Jul 2019	15 Jul 2019	16 Jul 2019	18 Jul 2019	22 Jul 2019

To structure a data set to build your CFD you need to track when each task has moved through a certain process state.

There are two ways to do this. We can list all of the tasks with their arrival dates for each process state.

This is an approach with a downside you should keep in mind. You can't track when work is going back and forth through process states. This approach is suitable if you only move tasks forward in our board. Keep in mind that if you move tasks backwards, this won't be taken into account - which could lead to some major problems going unnoticed.

Building a CFD from Your Board Data

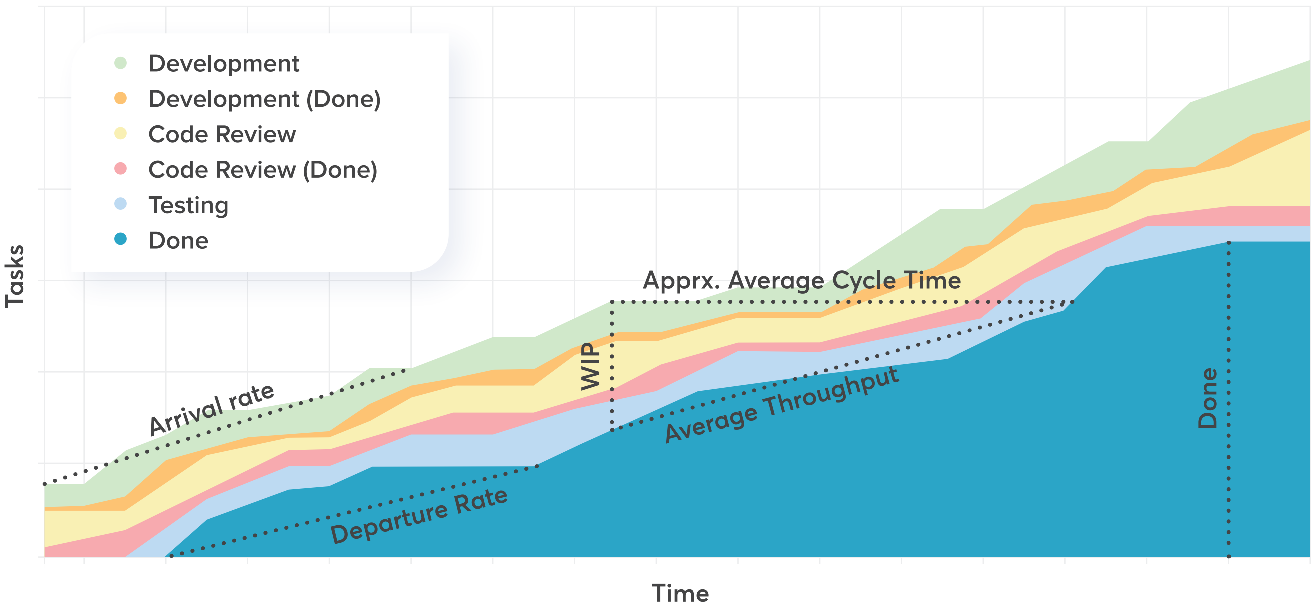
Method 2 - Listing Story Activities in Each Process State

Story ID	Process State	Arrival Date	Departure Date
1	Development	3 Jul 2019	4 Jul 2019
1	Development (Done)	4 Jul 2019	5 Jul 2019
1	Code Review	5 Jul 2019	7 Jul 2019
1	Development	7 Jul 2019	11 Jul 2019
1	Development (Done)	11 Jul 2019	14 Jul 2019

So to improve upon these flaws, there is a different way of structuring your data when building a CFD.

Instead of capturing just the arrival dates for each state, we will collect a list of task activities for each process state. Here, the end date of a previous activity will mark the start date of the following one. This way, we can record multiple activities per state, thus those where a task went through the same state multiple times. As a result, if we move cards backwards on our board, we'll have a more realistic overview of our workflow.

Introducing the Cumulative Flow Diagram



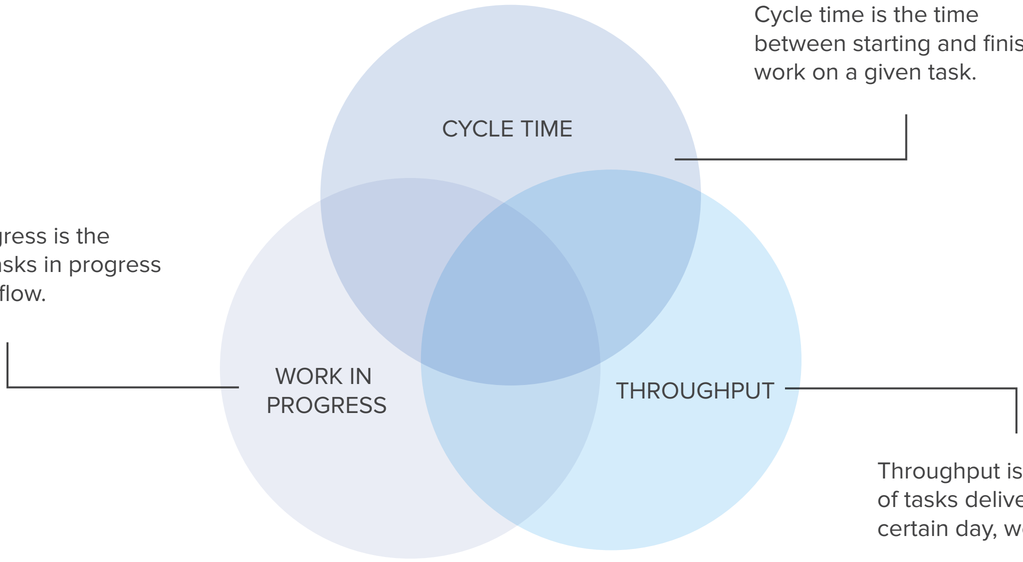
Let’s spend some time going over the anatomy of the Cumulative Flow Diagram.

The horizontal axis of the chart shows us a timeline. This is the time for which you're examining your workflow. The vertical axis is a cumulative number of the tasks in your process - giving the tool its original name. The graph is built from different colored bands corresponding to a state in your workflow. Each band tells you how many tasks are in each state at any given time with the bottom band showing how many tasks have been completed.

When your team pulls a new task into the workflow, the top band expands as the task enters the first process state. Once the task is moved onto the next step, that band shrinks and the one below widens, and so on until the task reaches the Done state.

Understanding the Three Key Flow Metrics

There are three main flow metrics we can track on the Cumulative Flow Diagram - work in progress, or the amount of tasks in progress in your system; cycle time, which is the time between starting and finishing a task; and throughput, which is the number of tasks delivered in a certain day, week or month.



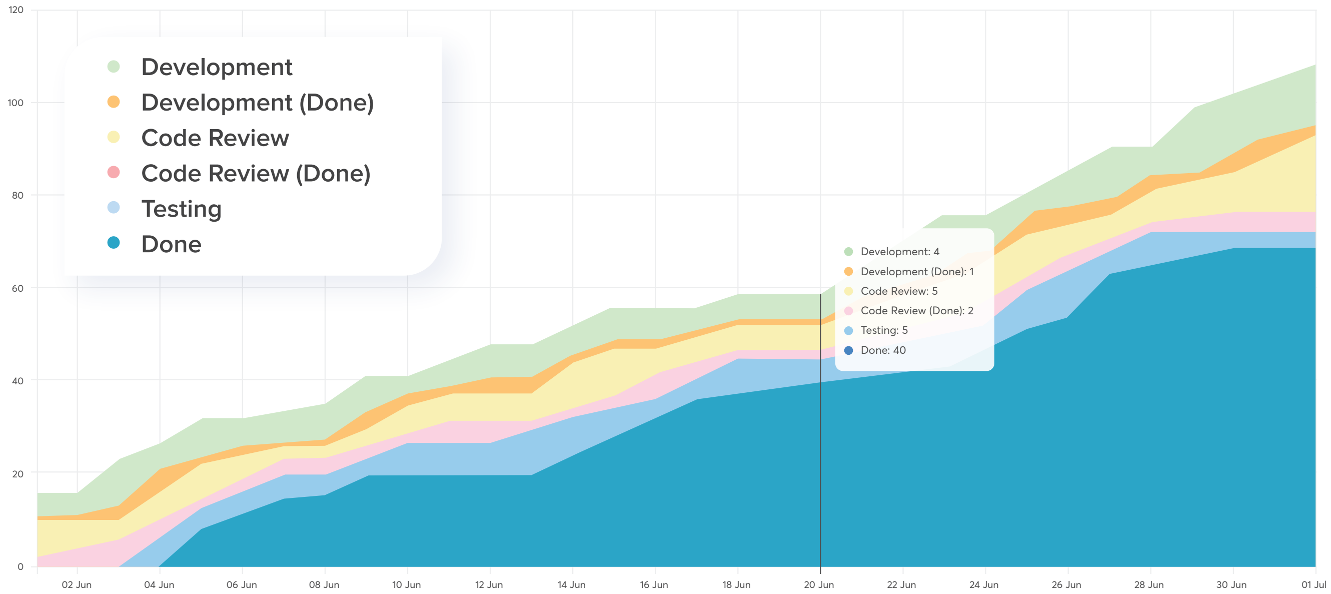
Cycle time is the time between starting and finishing work on a given task.

Work in progress is the number of tasks in progress in your workflow.

Throughput is the number of tasks delivered on a certain day, week or month.

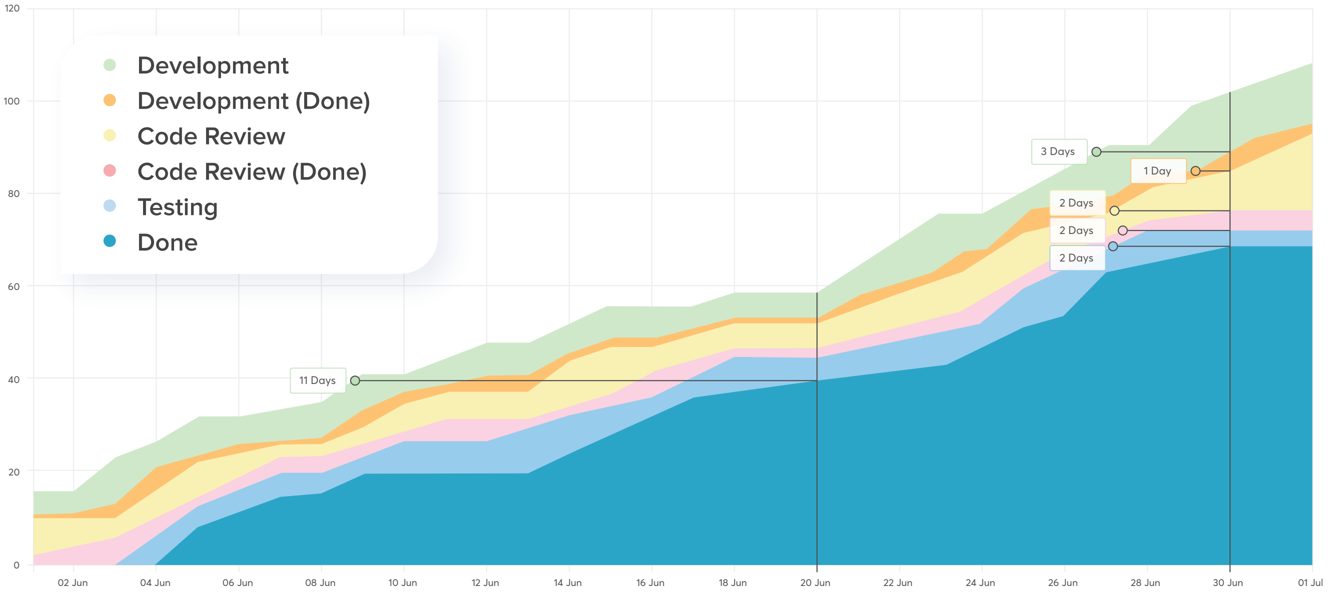
The goal is to limit work in progress, reduce cycle times and increase throughput. In other words, work on fewer things at the same time while delivering results faster and more often.

Work In Progress



Each band on your CFD represents a state in your workflow. The height of each band shows you the amount of work in progress in that state for a given time. For example, we can see that on the 20th of June we had 4 tasks in Development and 2 tasks in Code review (Done). We also know we have completed 40 tasks between the 4th and 20th of June.

Approximate Average Cycle Time

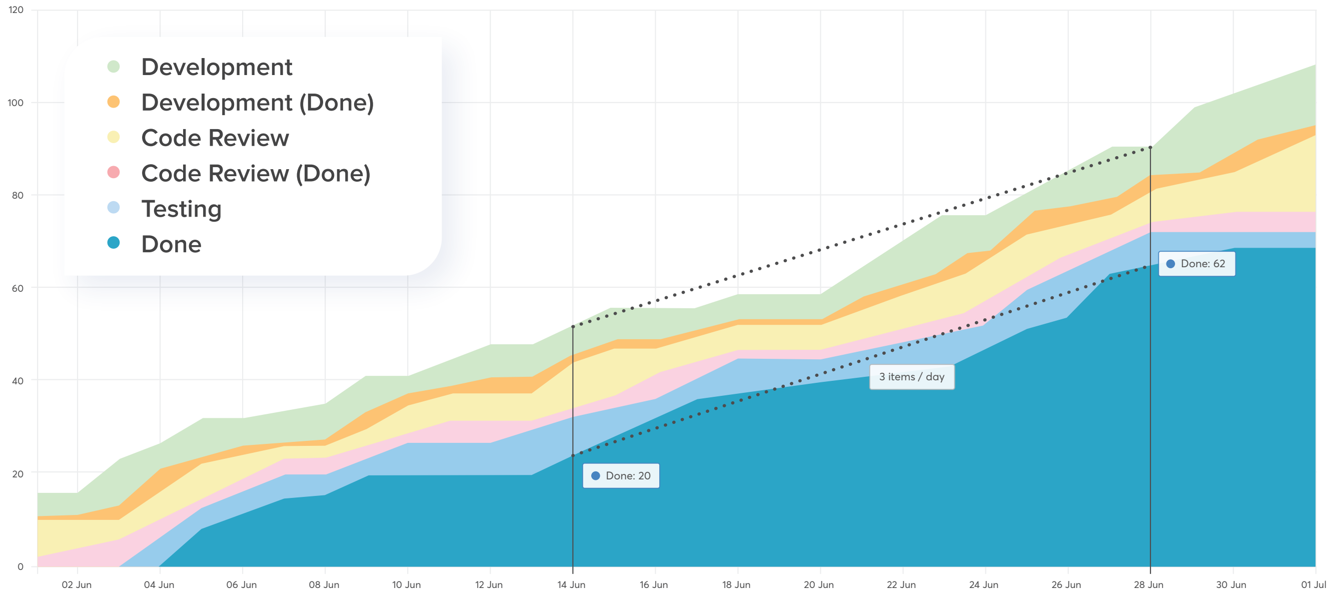


The approximate average cycle time tells you how long approximately, on average, you took to complete all the tasks in a certain date. To calculate it on your CFD, choose a point on the Done state and draw a horizontal line backwards until it meets the top line of the graph.

The Approximate Average Cycle Time then i the subtraction of the two dates.

So to calculate the Approximate Average Cycle Time for tasks that finished on the 20th of June, draw a horizontal line backward until the line intersects the top line - 9th of June. Our approximate average cycle time for the tasks completed on 20th of June is 11 days; Approximate average cycle time can also be calculated for single process states.

Average Throughput



To figure out the rate at which we're delivering work, we use Average Throughput. You can find your Average Throughput between any two points on the bottom line of your chart. On the other hand, if you draw a line between any two points on the top of your chart, you'll see your average arrivals. To maintain a stable workflow, aim to keep an equal distance between these two lines. Just like before, we can calculate our Average Throughput for each process state, too.

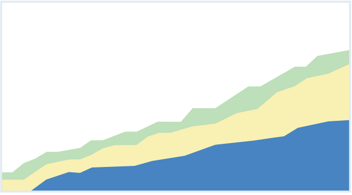
To calculate our average throughput in the period between 14st of June and 28th of June, we need the total amount of tasks for that timeframe. Given we've completed 42 tasks in those 14 days, our average throughput is 3 tasks per day.

The Most Common CFD Patterns

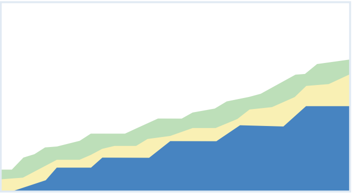
Now that we know how to track the main flow metrics, we can move onto the most common patterns on your CFD. A word of notice before we start.

Remember that, like any other analytical tool, the CFD won't tell you **how** to fix your problems. But it **will** help you detect them early. When there's a flaw within your system, this is where you'll see it first. Still you need to look into it within your context and take preventive action accordingly.

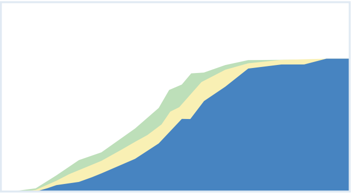
So let's have a look at how to spot the most common CFD patterns and what they mean for your process.



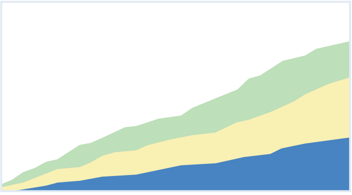
Differences in gradient



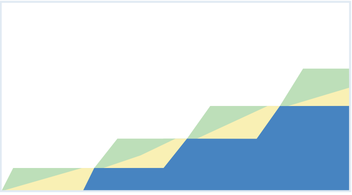
Flat lines



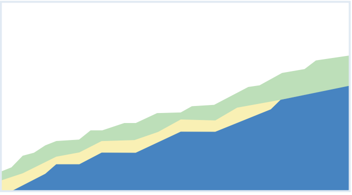
The S-curve



Bulging bands

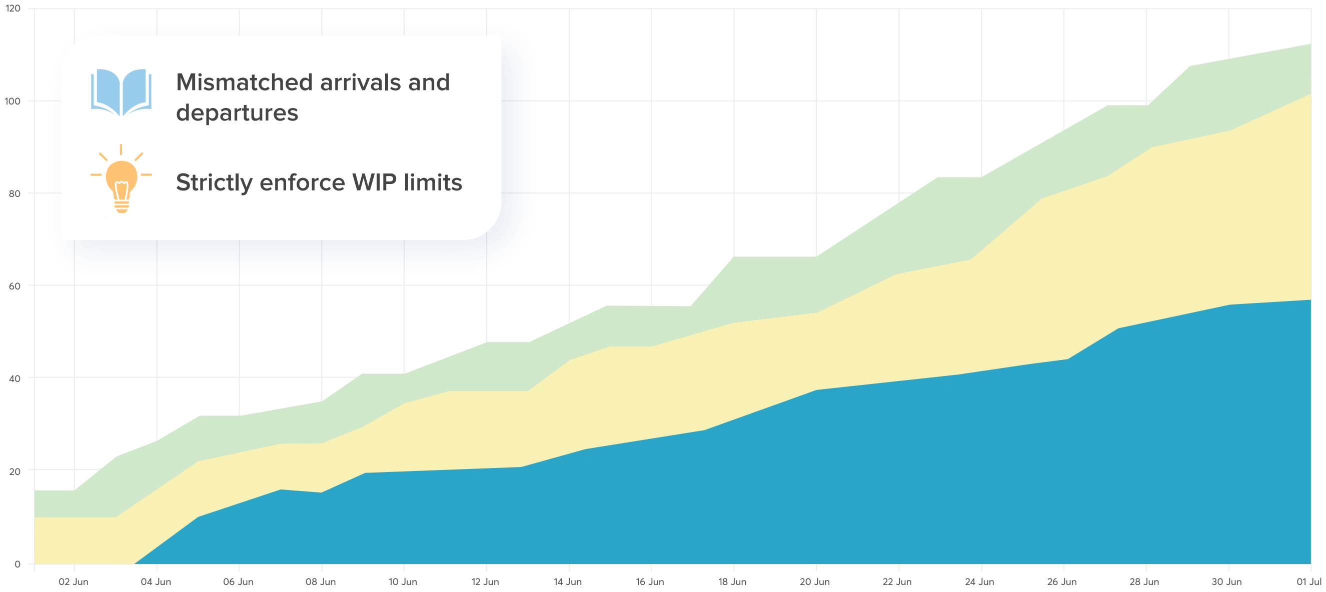


Stair steps



Disappearing bands

Differences in Gradient



Meaning:

Mismatched arrivals and departures. This is probably the most common CFD pattern you will come across. Usually, the upper line of a band will be steeper than the lower one. This means that tasks are entering the stream faster than they are being completed. When new work is arriving faster than old work is completed, your work in progress will increase over time. An increase in WIP will most certainly lead to an increase in cycle times, too. Also, as there is more and more work in progress, your team will start to multitask and switch contexts to keep up with the demand, ultimately becoming less efficient.

When work is taking longer to complete, deliveries become delayed and more difficult to forecast. This pattern is a good indicator the predictability and stability of your process are at risk.

Of course, we are assuming the team setup hasn't changed and the same amount of people are spending a similar amount of time on the project. If you see this pattern and you've recently expanded your team, an increase in WIP across process states is completely fine as there is now more people to handle the workload.

Suggestion for improvement:

There are certain Kanban practices that can help you tackle this pattern and stabilize your process. Strictly enforce and adhere to WIP limits.

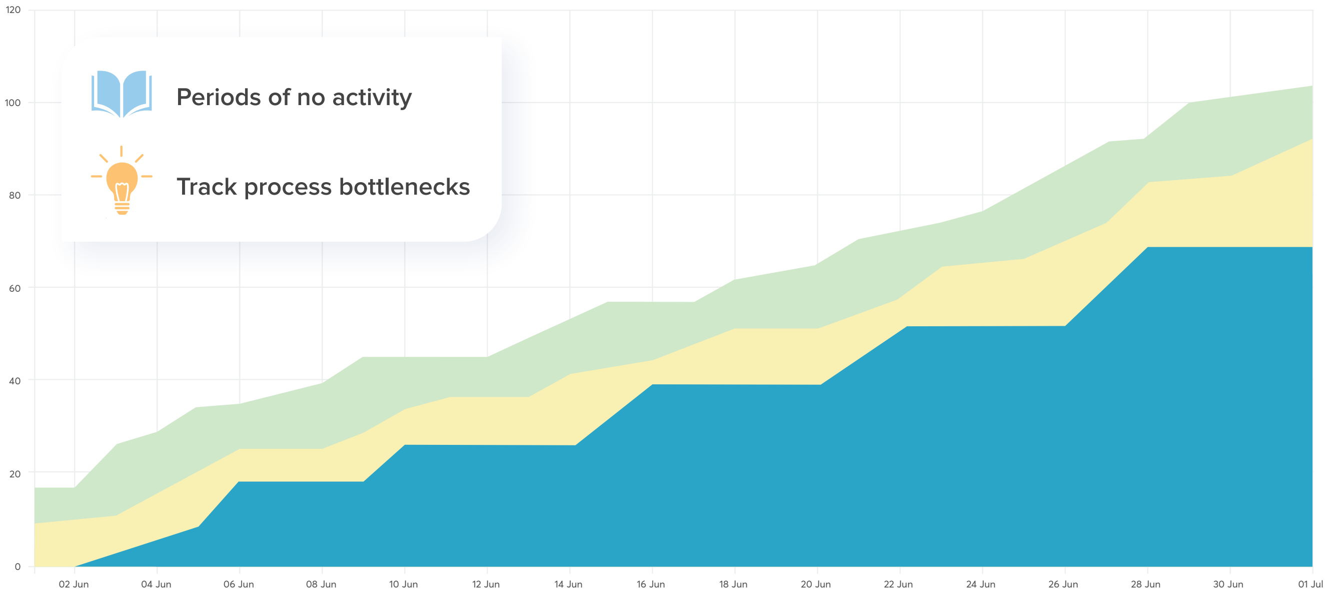
As stated in the Kanban Maturity Model, at maturity level 0, teams realize a limit to human capacity, a desire for relief from overburdening. They recognize that multitasking causes work to take longer and its completion to be unpredictable. It is better to “stop starting, and start finishing.” At this stage, it is crucial to understand that limiting work in progress helps you avoid multitasking and reduce the times for delivering work.

At level 1, teams start to apply WIP limits on a personal level, and at maturity level 2, organizations need to establish both personal WIP limits and team WIP limits.

Each new practice will cause the bands in your CFD to decrease, the arrival and departure lines will go in parallel meaning entering work is matching the rate of completed work. Once you ensure no work is started until the due tasks are delivered your team will keep their focus on one thing at a time. This prevents constant context switching and team overburden which boosts your overall efficiency and productivity.

Get more things done rather than do more things. Stop starting and start finishing.

Flat Lines



📖 Periods of no activity

💡 Track process bottlenecks

Meaning:

Almost every team experiences the CFD pattern in which process state lines flatten out. Flat lines on your CFD bands mean periods of no activity. The longer the lines, the longer your tasks aren't being moved through the workflow. Something is blocking your process and nothing is getting done.

Suggestion for improvement:

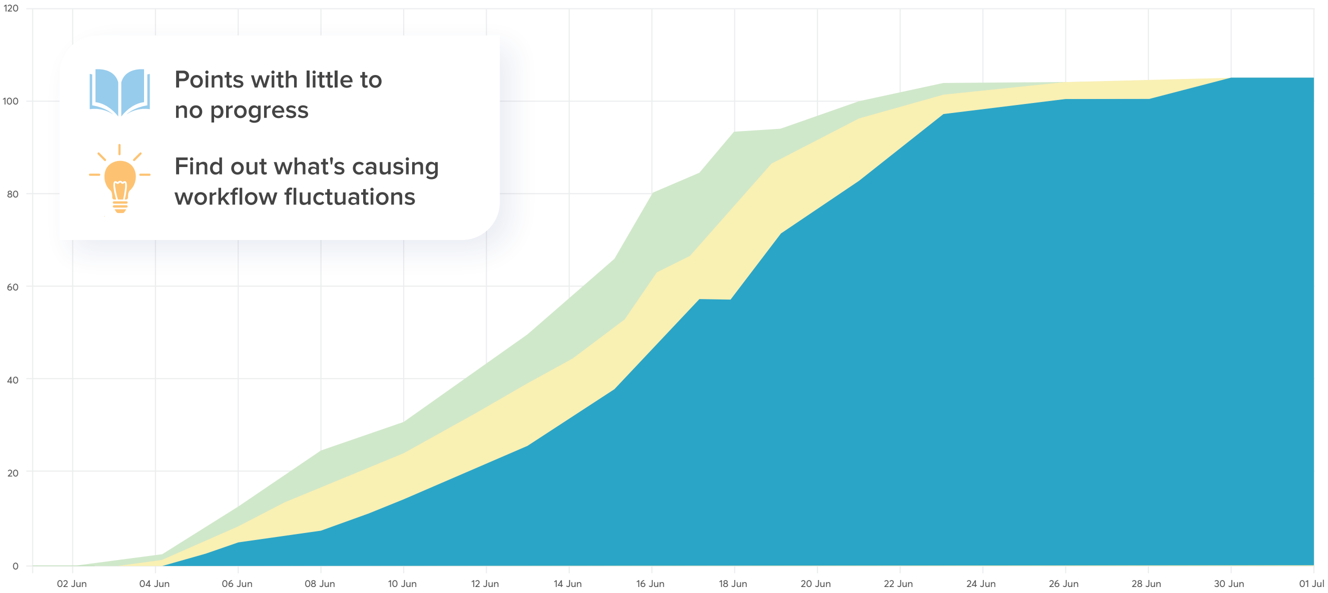
Let’s say all lines on your CFD flatten out at the same time. Before you raise alarms here, check for public holidays or company-wide events your team took part in. There might be a perfect explanation why your team hasn't been working during this time. Or maybe everyone is stuck due to server maintenance or any external blocker. Either way, talk to your team, identify and resolve the blockage. Brainstorm the ways of getting things moving forward and prevent the issue from happening again.

If the lines flatten out in the Testing state for example while other ones keep a steady progress, chances are you have a more serious problem at hand. Maybe your testing environment is causing the blockage, maybe your testers are struggling with demand and have started multitasking and switching context, unable to move work through. Either way, work is not leaving the testing phase, and consequently, you're not delivering anything - the lines become flat in the Done state too.

There is no point to continue starting items if they aren't getting tested and delivered. This indicates a cooperation issue, meaning your team is likely still at the Organizational Maturity Level 1. The team has developed some initial definition of process workflows, management policies, and decision frameworks but are struggling to follow them. WIP limits for the entire team have to be set. Teams at maturity level 2 start visualizing aging work and blocked items, which helps them understand why the process is getting stuck.

> **There is no point to continue starting new work items if they aren’t being delivered.**

The S-Curve



Points with little to no progress

Find out what's causing workflow fluctuations

Meaning:

The S-curve CFD pattern emerges when your progress is steepest in the middle. The preceding and following bands are either very thin or flat, indicating times with little to no work in progress. There are two classic examples of when this might happen. First, a project usually begins with zero WIP and ideally ends with zero WIP. An ideal sprint starts with zero WIP and ends with zero WIP. The S-curve CFD pattern causes your process to become less predictable and forecasting - less accurate.

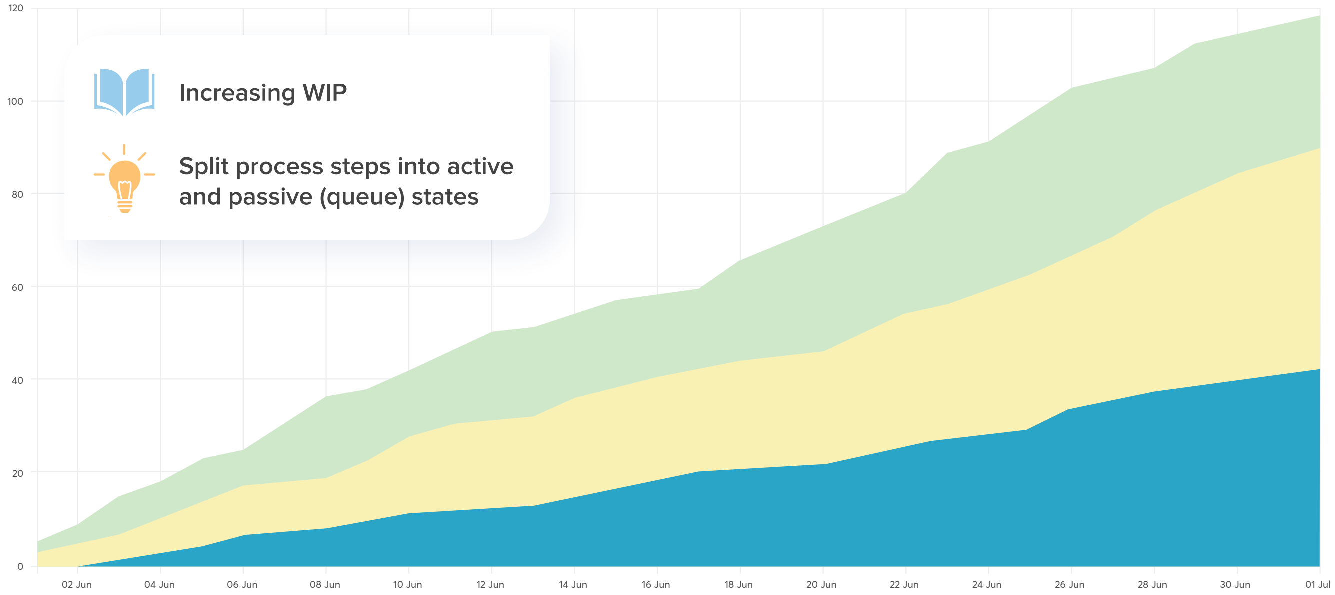
Suggestion for improvement:

WIP should be as consistent as possible in order to maintain a stable and predictable system. Aim to remove zero WIP flat spots to turn the S-curve back to a consistent gradient.

Oftentimes, there will be a blockage in one of the upstream processes preventing your team from continuing to work on tasks, thus causing zero WIP. It is important to visualize the blocked work items - a practice at ML2, understand the reasons for the blockages, and resolve them quickly. This way, you can raise awareness of the impact blocked work can have on your workflow.

As long as WIP is consistent and the average age of WIP is consistent, you are maintaining a predictable system.

Bulging Bands



Increasing WIP

Split process steps into active and passive (queue) states

Meaning:

This is a pattern where one or more bands increase in thickness. It means our work in progress is increasing. Our delivery times increase as well. More work in progress usually causes context switching and multitasking. This can take a serious toll on the efficiency of your team.

Suggestion for improvement:

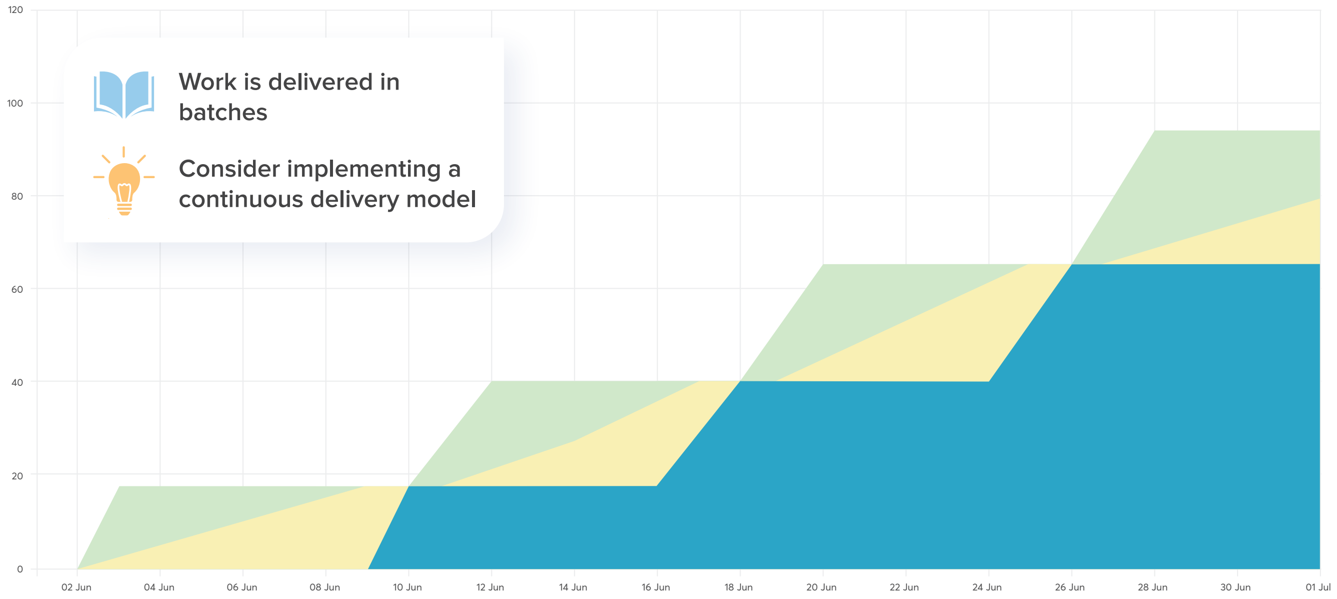
As with every other pattern, we are looking for the root cause of this trend. Why is our WIP increasing?

Assuming the size of our team is the same we need to investigate for other clues. Are tasks being pushed by the management instead of pulled by the team? Is there a blockage in a downstream process state? Try splitting your process states into Ongoing and Done – adding Done states or queue states lets you see where tasks are spending too much time waiting to be pulled. You'll be able to spot those problematic states right away.

This is a visualization practice for teams looking to reach maturity level 2. By separating your workflow stages in active and inactive columns, you can see exactly where your team is struggling with the demand.

> **Split your process states into Ongoing and Done to track where tasks are spending too much time waiting to be pulled.**

Stair Steps



Work is delivered in batches

Consider implementing a continuous delivery model

Meaning:

This is a really obvious CFD pattern. One or more of your bands will have a shape of stair steps. Stair steps are really common in the last state of our workflow, the release state. This means you're delivering tasks in batches or cadences, but it can also happen if you are moving a number of tasks to the next process state at once.

Suggestion for improvement:

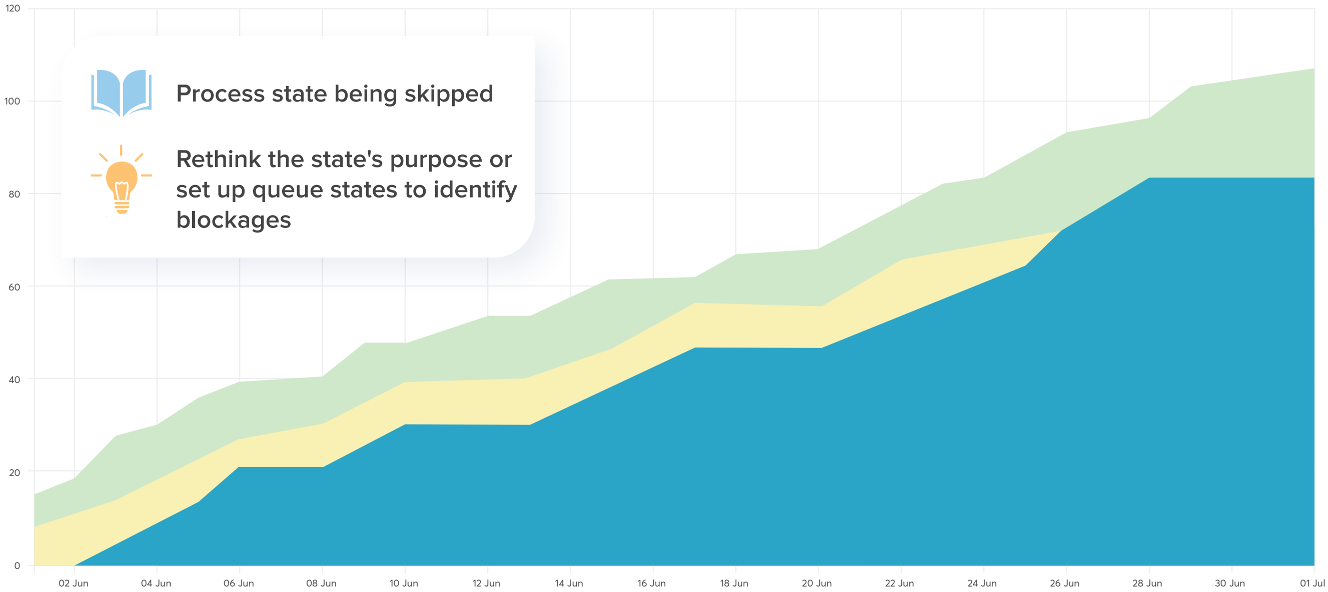
Stair steps are an indicator of regular cadences if the steps are equal in size. If the steps are uneven, chances are you are experiencing non-regular batch transfers.

Large batch deliveries put your system at risk, especially if we're talking about non-regular deliveries. Releasing work as soon as it is ready makes your future predictions far more accurate.

When you see stairs steps appear on your CFD, ask yourself how batches are affecting your system. Could they be reduced? How it will affect your cycle time? If your business model supports continuous delivery, transferring work in batches calls for unnecessary delays, risk of rework and late discovery of quality issues. Start delivering tasks as you complete them to maintain more predictable system.

> **Large batch deliveries put your system at risk. Releasing work as soon as it is ready makes your future predictions far more accurate.**

Disappearing Bands



Process state being skipped

Rethink the state's purpose or set up queue states to identify blockages

Meaning:

This is a pattern where a process state isn't visible on your cumulative flow diagram. Tasks could be skipping a process state in the workflow, or an upstream blockage is stopping them from getting to this state.

Suggestions for improvement:

Check for process states being skipped – maybe this state has no purpose in your workflow. Again, it might be useful to set up queue states (Done states) in order to identify hidden blockages. Splitting process states into Ongoing and Done will let you see where work is lingering and not being pulled forward.

Recap of the Most Common CFD Patterns

Pattern	Meaning	Suggestion for improvement
Differences in gradient	Mismatched arrivals and departures	Strictly enforce WIP limits
Flat lines	Periods of no activity	Track process bottlenecks
The S-curve	Points with little to no progress	Find out what's causing workflow fluctuations
Bulging bands	Increasing WIP	Split process steps into active and passive (queue) states
Stair steps	Work is delivered in batches	Consider implementing a continuous delivery model
Disappearing bands	Process state being skipped	Rethink the state's purpose or set up queue states to identify blockages

Keep in mind that you won’t see patterns as clean as our examples. You're more likely to see an overlapping combination of two or more patterns. Once you understand what the graph is telling you, it will be clear where to investigate for improvements and getting down to the root of your bottlenecks to deliver value to your customers more quickly and predictably.