ESCAPE VELOCITY

Better Metrics for AGILE TEAMS

DOC NORTON

Escape Velocity

Better Metrics for Agile Teams

Doc Norton

Escape Velocity

Better Metrics for Agile Teams

Doc Norton

ISBN 978-0-578-64483-7

Leanpub

This is a Leanpub book. Leanpub empowers authors and publishers with the Lean Publishing process. Lean Publishing is the act of publishing an in-progress ebook using lightweight tools and many iterations to get reader feedback, pivot until you have the right book and build traction once you do.

Cumulative Flow Diagrams

A Cumulative flow diagram (CFD) is a simple but very powerful tool used in queuing theory that helps to show the flow of work through a system. A CFD shows the quantity of work in a given state over time, which allows us to determine lead times, cycles times, Work in Progress, and changes in scope. I've often heard cumulative flow diagrams referred to as an alternative to the burn up chart, which is an alternative to the burn down chart. Just as email is an alternative to a printed letter which is an alternative to a hand written letter when I want to send the same urgent message to 50 people. All of them get the job done. One of them is so demonstrably better, the comparison suggests a strong penchant for less efficient tools, or suggests more than a bit of naivety.

Making a CFD

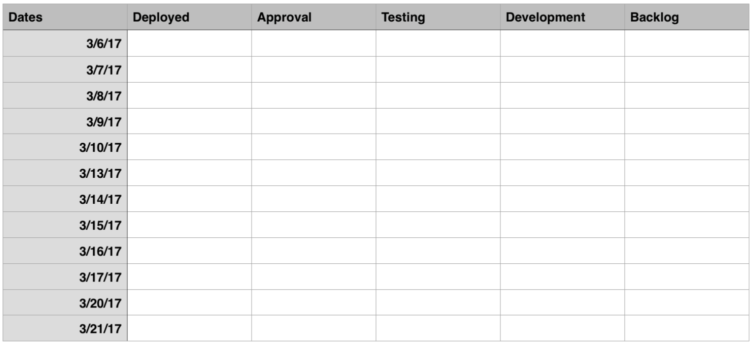
To make a cumulative flow diagram is relatively easy. More and more agile tracking tools offer a CFD report so you might not need to generate your own. But, in case you ever do, let's walk through the basic steps.

Just about any spreadsheet software can be used to create a CFD. On a sheet, create a column for each step in your software process. The least you'd need would be To Do, Doing, and Done. I recommend a more detailed set of steps, especially for newer teams. Be honest about the steps most stories go through. If stories go through grooming before development, and there is a formal testing phase after development with a formal approval process to promote to production, I'd consider making columns for all of the action items as well as the wait states between the action items. We'll demonstrate this in a bit.

Each day, or at some other regular interval, total the work in each of the states you're tracking. I recommend once per day. If you do a daily stand-up meeting, the end of the meeting is a great time to run your totals as the board is going to be as accurate as it is going to be for that given day. Record the totals in the spreadsheet and create a stacked area graph against them.

For our example team, we're going to say they have a backlog from which they pull work into development. After development, they have a formal testing phase followed by formal business approval and deployment into production.

For this team, I'd create a sheet that looks like the following:

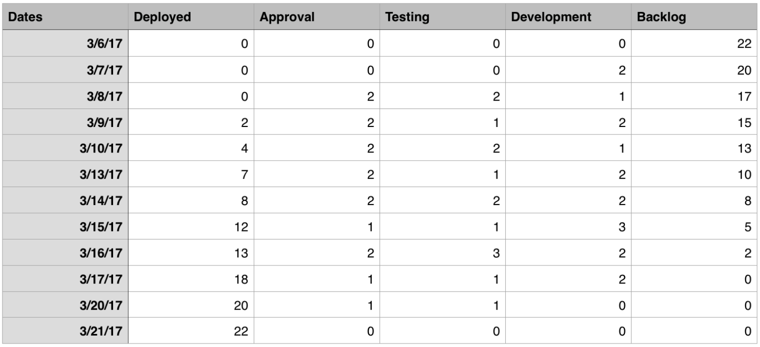


Dates	Deployed	Approval	Testing	Development	Backlog
3/6/17					
3/7/17					
3/8/17					
3/9/17					
3/10/17					
3/13/17					
3/14/17					
3/15/17					
3/16/17					
3/17/17					
3/20/17					
3/21/17					

Cumulative Flow Diagram Data - Empty

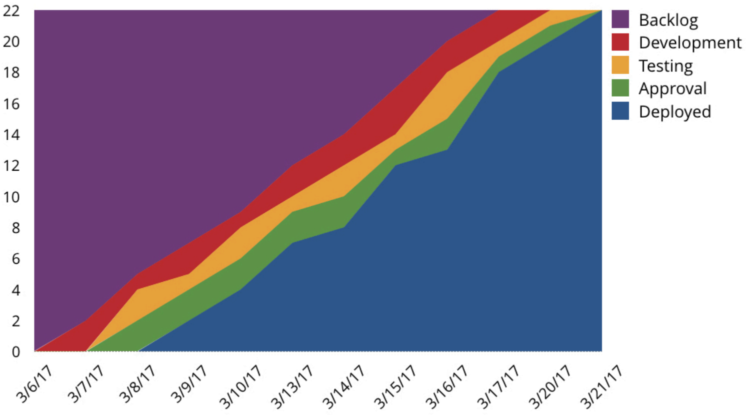
You might notice that the columns are in the opposite order of the actual flow of work. This is due to the way a stacked area graph works in most of the software I use. Your mileage may vary - it is a quick experiment to figure it out.

Then, as mentioned, fill in the values for each day (or whatever interval you select). Your stacked area graph will show the flow of work in a simple but useful way.



Dates	Deployed	Approval	Testing	Development	Backlog
3/6/17	0	0	0	0	22
3/7/17	0	0	0	2	20
3/8/17	0	2	2	1	17
3/9/17	2	2	1	2	15
3/10/17	4	2	2	1	13
3/13/17	7	2	1	2	10
3/14/17	8	2	2	2	8
3/15/17	12	1	1	3	5
3/16/17	13	2	3	2	2
3/17/17	18	1	1	2	0
3/20/17	20	1	1	0	0
3/21/17	22	0	0	0	0

Cumulative Flow Diagram Data - Populated



A Simple Cumulative Flow Diagram

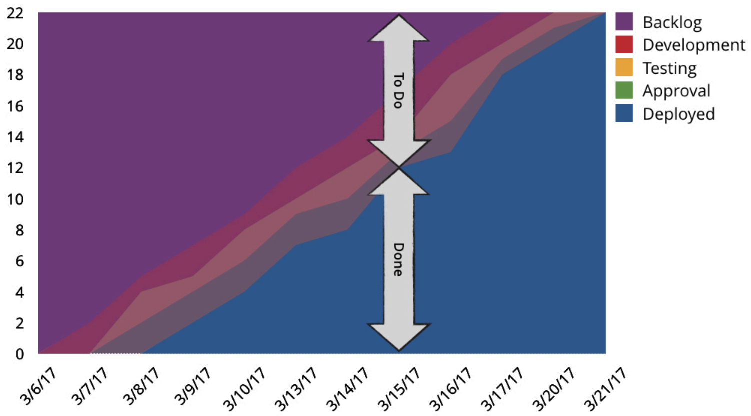
Reading a CFD

The above is a relatively simple cumulative flow diagram. The Y axis represents the amount of work in queue and the X axis represents time. Each color area on the chart represents a stage in the team's flow of work. Work starts in the form of stories in the 🟪 backlog. As the team works on a story, the major stages are 🟥

development, 🟧 testing, and 🟩 business approval. Once approved, they are deployed into 🟦 production.

This first example is perhaps a diagram from a team in Lake Wobegon [31] where the scope never increases, there is no re-work, and everything flows through at a smooth even rate.

Thin relatively consistent lines moving steadily up and right with the backlog decreasing in size while the deployed to production increases in size are all signs of a team operating in healthy fashion. Healthy in this case, means delivering accepted work at a consistent (predictable) pace. A cumulative flow won't tell us everything. For example, it is possible that the code is of poor quality or that team morale is down. We wouldn't be able to determine either of these things by looking at the diagram. A cumulative flow diagram may not be able to help us see all possible issues, but it is good at helping us determine if our process itself is effective.



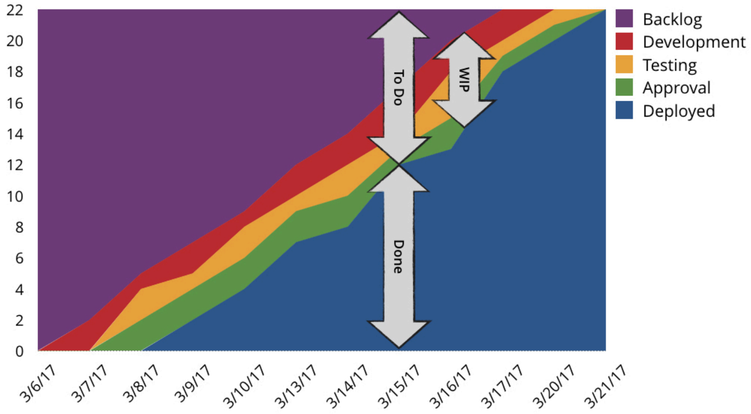
To Do and Done

First exposure to a CFD often leads to a sense of underwhelm. It looks like a multi-colored burn-up chart. And if we ignore the

[31] Lake Wobegon. (2017, April 15). Retrieved April 23, 2017, from https://en.wikipedia.org/wiki/Lake_Wobegon

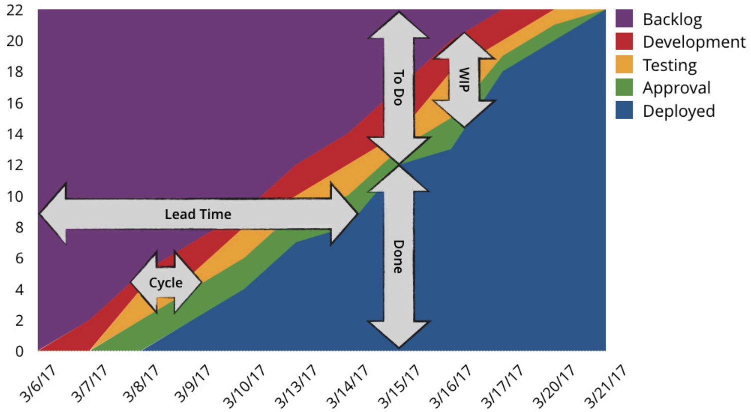
stages that represent “doing”, it is a burn-up chart. We can see the work to do and the work done. Add a simple trend line on the work completed and we’ve an actual burn-up, replete with an estimated completion date for the work remaining.

If this were all the value you got out of a CFD, it might not be worth the effort of tracking the data and generating the chart. But this isn’t the only value one can get out of a CFD. Let’s take a closer look and learn a bit more about how to read one.



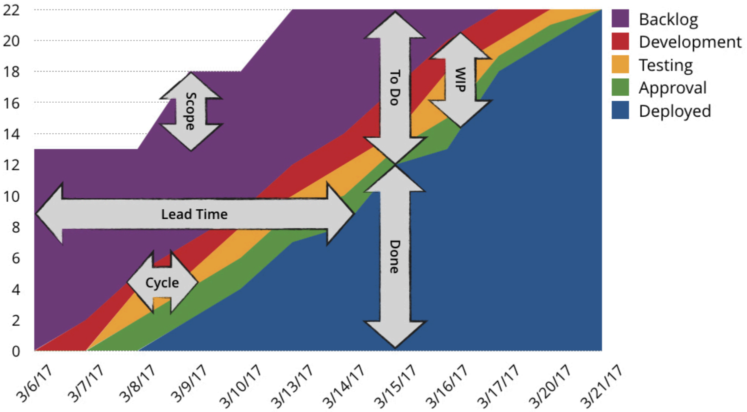
Work In Process

With a CFD, we can also see how much work is in process at any time including how much work is being attended to at each stage of the active process.



Lead and Cycle Time

We can also see the lead time and cycle time for stories, which gives us a clue about how long the overall process takes compared to the actual development effort. Adding more steps to the CFD, including times where items are waiting between activities can tell us a great deal about where the delays are in our process. This can help to focus our efforts for improvement. If it turns out that the most significant delay is when stories are waiting for testing, we can put more effort into looking at root cause and reducing the delays. Without this information, we might spend more time optimizing phases of work that increase the throughput prior to the bottleneck, which would produce no discernible improvement in delivery. Local optimizations done without understanding the system are a perfect path to increased waste without decreased lead time. A well thought out CFD can give us insight into the overall system.



Scope Change

A CFD can also show us changes in scope. The above CFD shows two changes in scope for the project. In both cases, more work was added to the team. Changes in scope are represented through a raise (addition) or dip (removal) in the top line of the graph. Here we see that new work was added on March 7, pushing the delivery date from March 15 to March 17. And additional work was added on March 10, pushing the delivery date out to March 21.

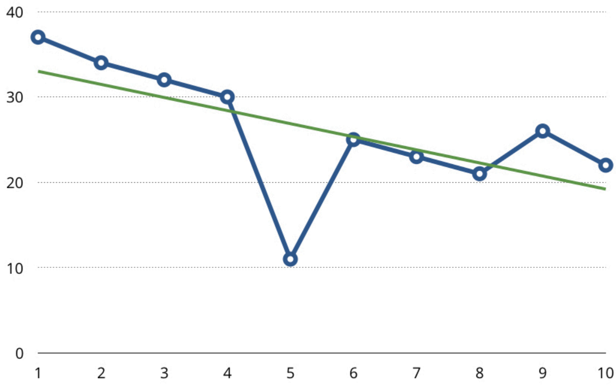
Getting Value out of a CFD

Alright. You can make a CFD and you can read a CFD. This is pretty good, but I want to help emphasize the benefit of a CFD.

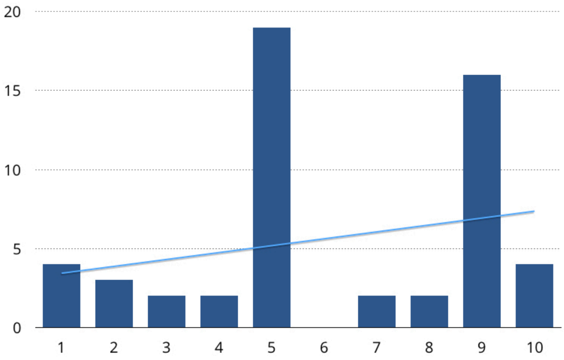
Let's take a look at an imaginary team. This team has been together for multiple iterations, and their velocity is all over the place. The stakeholders are frustrated. New feature requests, even when expedited, take months to get into the customer's hands. This team is slow and unpredictable.

In a retrospective where they are discussing how they can improve their process, the team looks at a product burn down and a velocity chart. But can’t determine a great deal from either.

What about you? Take a look at the Product Burn Down and the Velocity Chart below and see if you can make an educated guess about what challenges this team is facing.



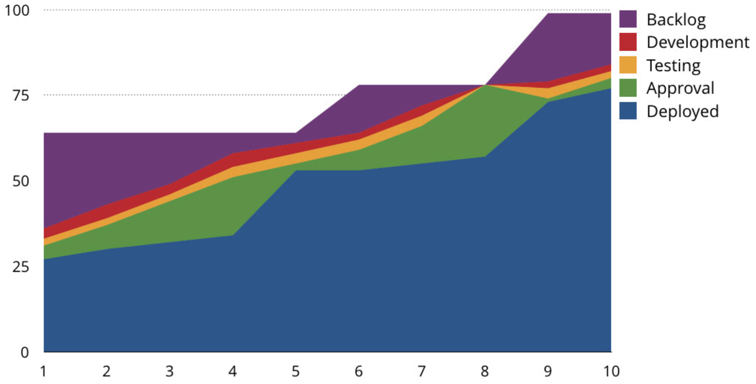
Product Burn Down



Velocity Chart

The team then takes a look at a Cumulative Flow Diagram for the same period of work and a key issue jumps out at them right away.

What about you? Take a look at the following CFD and see if you can make an educated guess about what challenges this team is facing.



CFD

As you may have deduced, the team's primary challenge is in the approval process and delivery of new work. The product owner is a traveling sales person and does not have time for approvals when they are doing a lengthy tour of a foreign market. Upon their return, they review a bunch of features and then proceed to add new work to the backlog based on customer feedback they received while on the road.

A CFD tells us more about our process than most other charts and, as we've seen, it can often be fairly easy to intuit where the problems lie. If we discover that the development process is where the bottleneck seems to be, we have the options of breaking development down into smaller stages if appropriate, or perhaps looking at different types of work - such as stories versus fixes, or items that require database or infrastructure changes compared to items that don't to gain further insights.