

# 4.6: Monitoring Sprint Progress

*In this chapter we show two ways to measure the progress of a Sprint, the Task Hour BurnDown and the Checklist Item BurnUp. We discuss the pros and cons of each, although we have a preference for the Checklist Item BurnUp.*

## Table of Contents

<i>The Sprint Task Hour BurnDown</i> .....	1
<i>The Sprint Checklist Item BurnUp</i> .....	4
<i>Discussion</i> .....	5

### Notes:

- Need to address when story won't fit in here
- No partial credit for stories for review or Sprint metrics (release metrics)

Once the Team commits to stories for a Sprint, it starts work on them. During the Sprint the Team needs to know if it is on track – if it will finish what it has committed to. There are many ways to do this, from simply observing the movement of Stories and Tasks across the Task Board to graphing metrics.

In this chapter we consider some metrics to measure and graph, if that's the way the Team wants to track its progress within a Sprint. Historically, the most common way to do this is the Sprint (Task Hour) BurnDown chart that is presented in most writings on scrum. In this chapter we show this BurnDown as well as introduce what we think is a better chart to use, the Sprint (Checklist) BurnUp.

These are fairly simple graphs to understand, so we'll just describe them quickly and give examples. After the examples we'll discuss these graphs, with particular emphasis on why we think the Checklist Item BurnUp is better.

## The Sprint Task Hour BurnDown

The most common way teams measure a Sprint's progress is the Task Hour BurnDown chart. I assume you are familiar with it, and know that it displays *the estimated remaining hours of work for the Sprint*. It is calculated by re-estimating the hours remaining on the Sprint's tasks every day, adding them up, and charting the result on a

day-by-day basis. It is called a BurnDown because the total estimated remaining hours should "burn down" to 0 as the Sprint is completed. Here is an example of a typical BurnDown chart for a 2-week Sprint.

### ***Example***

The CatAir Team doesn't do a single Planning Day as described in Chapter 4.1. It does its review and retrospective on every other Wednesday afternoon, and its planning on the following Thursday morning. It then immediately starts work. Its daily meetings are just before lunch.

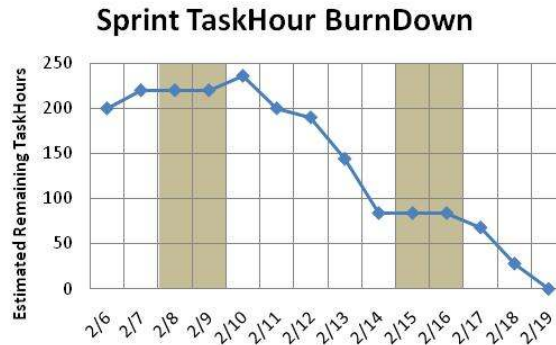
The following table shows the progress of the first Sprint, held from 2/6/08 to 2/19/08 (planning on 2/6, Review and Retro on 2/19). This table shows the Team's estimates on the number of hours remaining on the tasks, as collected at the daily meeting just before lunch.

The way to read this is:

- The first column contains the initial estimates for the tasks, as derived at the Sprint Planning meeting. For a new task, its initial estimate is on the day it was added – their names are also prefaced by an asterisk (\*).
- A new estimate (of time remaining) is added at the daily meeting if the task was worked on since the previous daily meeting – so a "blank" means the task was not worked on.
- The "total" row adds up all the current estimates, either as changed or carried along from previously.
- Note that the estimates can go down, remain the same, or even increase. The estimates of time remaining do not depend on the hours actually worked.
- Note, also, that the stories in this table are prefaced by their storytypes in square brackets, like "[backbone]" – these storytypes will be discussed in Section 8.

	2/6	2/7	2/8	2/9	2/10	2/11	2/12	2/13	2/14	2/15	2/16	2/17	2/18	2/19
[enviro] Install Copy of CUTLASS in Lab														
Set up clean machine in lab	8	12			12	0								
Install CUTLASS on new machine	8	12			18	0								
*Test CUTLASS on new machine		4			6	0								
[analysis] Get Smart on CUTLASS Capabilities:														
Analyze CUTLASS Documentation	12	16			24	24	0							
Prepare briefing for Team	4	8					8	0						
[backbone] Get List of Flights from CUTLASS														
Architecture and Design	32						36	4	2			0		
Write Functional Tests	12						14	12	0					
Code and Unit Test	80						88	84	48			42	12	0
[enviro] Set Up CatAir Website in Lab														
Set up website on test machine	24											10	4	0
*Test CatAir on test machine												6	4	0
[analysis] Determine Size of "Check Status of Flights"														
Investigate "Check Status of Flights"	16								8			8	6	0
Prepare briefing for Team	4								2					0
<i>Total Estimated Hours Remaining</i>	200	220			236	200	190	144	84			68	28	0

And, here is the Sprint TaskHour BurnDown as calculated from this data...



There are two things to notice about this graph:

1. It goes up at the beginning. This often, if not usually, happens with a task BurnDown. The reasons for this are twofold: some of the tasks are harder than estimated, and there are often tasks that are discovered once the work begins. There are examples of both of these cases in the data presented.
2. Using a TaskHour BurnDown puts the focus on doing tasks, which is not what the Team committed to. The Team committed to completing stories, as defined by their "doneness" Agreements.

For these two reasons, I like to use a different graph for measuring progress within a Sprint, which we now present.

## The Sprint Checklist Item BurnUp

Just like the Sprint BurnDown, the Checklist Item BurnUp's purpose is to show the Team that it is "on track" to finish the Sprint. Unlike the BurnDown, however, the focus is on the "doneness" Agreements that were agreed to during planning. The reason for this is that the "doneness" Agreements are the what the Team has actually committed to.

In order to turn this into a graph, we convert the "doneness" Agreement for each Story to a checklist, which must all must be "checked off" in order for the Story to be completed. So, by combining all the Story's checklists we can get the Sprint's "doneness" checklist. Thus, progress can be measured by simply counting up the number of "checked off" Items, and comparing it to the total number of items in the Agreements.

There are two basic ways to show this information: as a BurnUp graph that just shows the count of "checked off" Items, or as a "percent done" graph, showing what percentage of the total Items are checked off. We will display the BurnUp graph here. In other words, if the Team has committed to 35 Items in the Agreement, the BurnUp just has to climb to 35 in order for the Team to be done with the Sprint.

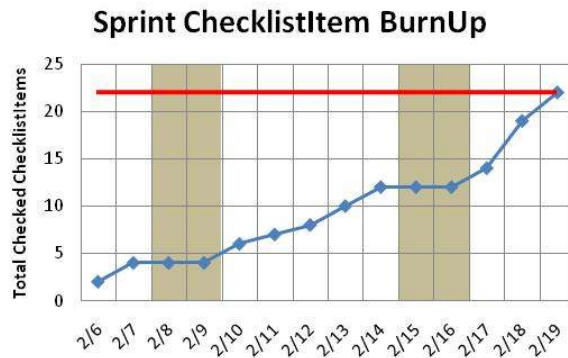
### An Example

The following table shows the progress of CatAir's first Sprint, held from 2/6/08 to 2/19/08 (planning on 2/6, Review and Retro on 2/19) – the same Sprint as for the Task Hour BurnDown example. In this table the "doneness" criteria are listed beneath the respective stories, which are given including their storytypes (see chapters...). The date that each checklist item was completed is given in the right column.

	Date Completed
Story: [enviro] Install Copy of CUTLASS in Lab	
✓ Get CUTLASS Install from SirJeff -----	2/7
✓ Set up clean machine-----	2/6
✓ Install CUTLASS-----	2/10
✓ Do Smoke Test to see if it works-----	2/10
Story: [analysis] Get Smart on CUTLASS Capabilities	
✓ Get Documentation and other Sources-----	2/6
✓ Study, Analyze, etc. (includes exploration)-----	2/11
✓ Report Results to Team -----	2/12
Story: [backbone] Get List of Flights from CUTLASS	
✓ Review Architectural Decisions with Team -----	2/13
✓ Design Review-----	2/14
✓ Review Functional Test Strategy-----	2/13
✓ Review Unit Tests -----	2/14

✓ Verify working on Development Machine-----	2/17
✓ Code Review-----	2/17
✓ Functional Tests Written -----	2/18
✓ Verify working on Integration Box -----	2/18
✓ Add Functional Tests to Regression Test Suite -----	2/19
Story: [enviro] Set Up CatAir Website in Lab	
✓ Set up website on test machine -----	2/18
✓ Set up Default Home Page -----	2/18
✓ Verify website can be reached from Development Machines-----	2/19
Story: [analysis] Determine Size of "Check Status of Flights"	
✓ Get Documentation and other Sources -----	2/7
✓ Study, Analyze, etc. (includes exploration)-----	2/18
✓ Report Results to Team -----	2/19

The following chart shows the completion chart of the Sprint. The dates are given along the bottom, with the cumulative number of completed checklist items shows on the vertical axis. The total number of checklist items is given by the line across the top. This shows what the graph should be climbing towards.



## Discussion

The Task Hour BurnDown has always bothered me, for a number of reasons, as follows:

1. The Task Hour BurnDown shows information that is about effort, and not the product. That is, the focus is on tasks, and not the “doneness” Agreement. This re-enforces the (wrong) notion that scrum is about people doing work, rather than a team producing product.
2. The BurnDown chart goes *down* when showing progress – and this is counter-intuitive. Charts showing progress should go *up*. I know that this is a silly reason to object to it, but many people (and teams) have expressed this concern to me. Many of them try to make their BurndDowns go up – which is hard to do,

actually, because you don't know where the "top" is. In order to make it go up, in fact, many of them insist that the estimates for tasks can't increase from one day to the next – and this is *so wrong*, as it intentionally denies reality.

3. The BurnDown chart often has a strange shape, with the chart going up in the beginning, and then going down, as we see in our example. This is irritating and hard to explain to Stakeholders, and often leads to an expressed management need for "better estimating" from developers, which isn't the point. Developers need to be better developers, not better estimators. We need better product and better commitments, not better estimates.

Since I am presenting a new chart, you would expect that it addresses the objections I just listed. It does. Just like the Task Hour BurnDown, the Checklist Item BurnUp's purpose is to show the Team that it is "on track" to finish the Sprint. Unlike the BurnDown, however, the focus is on the "doneness" Agreement, and this is what the Team has actually committed to. Therefore, it more accurately represents something of value to the Stakeholders.

To summarize, then... The Task Hour BurnDown and Checklist BurnUp charts have the same purpose: to show how the Team is doing throughout the Sprint – to answer the questions "Will we finish on time?" and "How are we doing?" However, they do it by focusing on two completely different things:

- The Task Hour BurnDown focuses on the work the Team is doing. It uses the Team's estimates of how much time is remaining on their tasks to indicate if the total remaining effort is trending to zero.
- The Checklist BurnUp focuses on the "doneness" criteria for the stories the Team has committed to. It counts how many items have been "checked off" and indicates if the Team will check them all off by the end of the Sprint.

Either of these charts does the job of determining if the Team will finish. However, only one of them actually focuses on what the Team has committed to and what provides value. We believe that the Checklist BurnUp has several good qualities:

- It measures what is important – the product
- It goes up, not down
- It is easy to calculate and explain

Give it a try...