

Capacity Planning: Respecting the Buffer

Tamara Sulaiman, PMP, CST

March 9, 2009

When I go in to help teams suffering from incomplete or missed iteration goals, one area that I naturally look at is the iteration planning. How is the iteration planned? Does the team use velocity as a measure of “yesterday’s weather” with capacity as a double-check? Is the team empowered to say no to more work? Are they using a buffer in their capacity planning? If so, what is it? Has it been adjusted lately? How?

I also look at what happens during the iteration. Are the team and management respecting the buffer? How are critical fixes and production support being handled during the iteration? While there can be many root causes for a team consistently not meeting iteration goals, I do find that a misunderstanding of what the buffer is for--or simply not respecting that buffer--is a leading cause of missed iteration goals.

What is the buffer?

The buffer is calculated as part of capacity planning for iterations. It is, in a nutshell, time during the sprint that is “protected” in the iteration plan. If a team is planning out their iteration and fills it absolutely to capacity, then the sprint plan is extremely brittle. This is very risky, with teams usually discovering that for one or many reasons they can’t complete all of the work planned for the iteration.

The buffer is time that is used for various things that aren’t detailed in the task breakdown portion of the iteration plan. Every organization has a certain amount of “noise” that occurs. By noise, I mean the things that are non-iteration related that take team members attention away from meeting those goals. Noise includes the e-mails that have to be read and answered, other meetings, colleagues that stop by to ask questions, and a ringing phone.

Also included in the buffer is time for the “overhead” associated with using Scrum or other agile project management techniques. This includes time for daily standup meetings, sprint planning, sprint demo and the sprint retrospective. Experienced teams include time in the buffer to “groom” the backlog. The grooming activity is time used by team members for things like reviewing and sizing stories, working with the product owner to split large stories into smaller stories with appropriate acceptance criteria, and all of the discussion that is around those types of activities.

An additional benefit having a buffer in the capacity plan is that it also acts as a way of mitigating the risk of the unexpected, including brief absences and tasks that are more complex than originally estimated. If a team member’s iteration is filled to capacity, then an unexpected absence of one day can put it all at risk. The same thing happens if a task is initially estimated to take eight hours--and turns out to be more complicated and ends up taking 16 hours. Having time in the iteration that is not allocated to a specific task provides the necessary flexibility that that the team needs to self-organize around the work on a daily basis.

It’s right around this point in my explanation to clients and classes that a product owner or project manager starts to get agitated about “wasted” time and “underutilization” of staff. It is important to recognize that planning for capacity and allocating for time actually spent on a project during one iteration are not the same thing. When team members are attending the daily stand up meeting, they are working on the project and should record their time appropriately. The same holds for unplanned (but necessary) design meetings or any of the myriad other types of meetings and discussions that come up each day as the team sprints toward meeting their goal.

Timesheet tracking records actual time spent working on a project, without too much regard to what that time was spent doing. Capacity planning for a sprint, on the other hand, is planning for how much time is available to complete the “hands-on” tasks to deliver the work.

What is the relationship between velocity and capacity planning?

At this point, many agilists will be asking, “Isn’t that what velocity is for, to help determine the amount of work a team can bring into an iteration based on the amount completed in the previous iteration?” Well...yes, it is. But real life is, at times, more complicated than that.

When a team reviews the top priority backlog items during iteration planning, they will often use their velocity as a gross measure of how much work they can expect to bring into an iteration. That’s fine as far as it goes. Velocity, to me, is also a gross level metric that helps teams, product managers and other managers predict the amount of scope that can fit into a given time frame (like a release.) Or, said another way, velocity helps determine how much scope can be accomplished for a given time frame. That’s a great metric, as far as it goes.

As anyone who has helped to plan and track iterations knows, the capacity for work often changes from iteration to iteration. Even with the team members remaining the same, the availability of those team members to accomplish work during the iteration may not stay the same. For example, let’s say your iterations are 14 days long. You are using the calendar method to calculate the iteration start and end dates. The iteration starts every alternate Wednesday and ends every alternate Tuesday (stay with me here...). This provides the stability to set dates, arrange rooms, send out invitations and other coordination stuff in advance. This stability is very useful for planning and coordinating your iterations.

The downside to this is that there may not be the same number of work hours available (i.e. capacity) in every sprint. Holidays are a great example of this; vacation time also impacts available capacity. Team members may have scheduled time off for training and other personal development type commitments.

Another impact to actual iteration capacity is the allocation of team members to different projects. In my real world experience, I rarely have a team where everyone can be 100 percent committed to one project or scrum team. We often have team members with critical skill sets that are shared across teams. When these things are what you are dealing with, capacity planning for the sprint provides a finer grained tool than velocity to help with these circumstances.

How do you calculate the buffer?

As an agile project manager or ScrumMaster, I usually work with the team to calculate our iteration capacity near the beginning of our iteration planning meeting. That way, when we move to the task breakdown part of the meeting, we have an important input to the iteration plan already available. I'll lead them through calculating the capacity plan; including an appropriate buffer.

I've seen some team members attempt to define their needed buffer time in hours. They'll say, "Well, I need about five hours a week for production support questions, usually, and then there is..." A problem with this approach is that it is extremely rare to be able to think of everything that might fall into that buffer number. A better approach is to start with a percentage of the total available capacity in the sprint, and then adjust each iteration until the right buffer level is discovered. This usually takes less than three cycles.

For teams that are new to agile, I recommend that they start with a 50 percent buffer. Over time I have learned that this is a good starting point for most teams and most organizations. At the iteration retrospective, when the team is discussing the iteration and how it went, one topic that should be discussed is whether 50 percent is a good number for them to use in planning.

Buffer Calculation

*To calculate the buffer for a 4-week sprint, begin by calculating the Starting Capacity for each team member. E.g., a 4-week Sprint = 20 days at 8 hours per day = 160 Starting Capacity in hours. Then subtract vacation and holiday hours. Doing this for Java Joe, say, might reduce his Starting Capacity to 128 hours. Then find out from Java Joe what percent of those 128 hours he can devote to your project given his other commitments. Let's say he can allocate 75 percent to you which is 96 hours (128 * .75). Using my 50 percent rule of thumb for a buffer, that would leave 48 hours for the buffer and 48 hours left for what I call Net Capacity. You can whip all this up in an Excel spread sheet for the entire team and arrive at a Total Team Buffer and Net Capacity in hours. By the way, I only count those team members who are taking on work in the iteration plan; working hands on keyboard, delivering functionality in one way or another. I don't include myself unless I am taking work off the iteration plan. We then have net capacity hours that allows for buffer time for each team member and the team total.*

What happens if we don't respect the buffer?

One of the principles supporting the agile manifesto states that agile processes promote sustainable development: "Agile processes promote sustainable pace. The sponsors, developers, and users should be able to maintain a constant pace indefinitely."

If teams and product owners don't respect the buffer, they run the risk of having more work than they can complete and still maintain a sustainable pace--or even meet the commitment at all. This leads to frantic overtime in an effort to meet that iteration commitment. The discipline needed for the agile practices to work well begins to break down. With these mini "death marches" each iteration, team members are unable to keep the quality of work at acceptable levels, so technical debt starts to mount up. As we all know, increasing technical debt makes the work more difficult, slowing the pace of work significantly as we move through iteration cycles.

With the frantic pace to even try to meet the "over-full" iteration scope, reviewing and sizing upcoming backlog items takes a lower priority and often doesn't get done, which in turns impacts the quality of the next sprint planning; it takes away from the necessary time to hold effective team retrospectives, which is necessary to developing and implementing plans for incremental improvement. Everything begins to break down and can start to spiral out of control.

Over time, the level of commitment to meet iteration goals starts to fall apart. It becomes acceptable to "commit" to a scope of work and not meet it. "It's not important," I've heard team members comment, "we'll just move it to the next sprint." That's just not true. Meeting iteration goals is important. It is critical on many levels, including holding the team accountable for their work, providing predictability of delivery and keeping the discipline necessary for agile practices to be effective.

I've seen product owners and managers encourage this type of over-commitment in the belief that stretch goals are good. In my opinion, stretch goals stink! When I see that teams have them, it smells to me that the team may not be fully empowered and/or are unable to accurately plan their iteration. That's a subject for a different discussion, however.

Another scenario where the buffer may not be respected is related to taking on new work into the sprint. Emergency and production support items happen. For teams that are developing new features and providing production support, there must be an effective mechanism to account for the production support work before it overwhelms the team and makes it impossible for them meet their iteration goals. I have seen teams decide that their production support work will be built into the buffer. Typically, once the sprint starts, everyone forgets to track the number of hours going toward production support. Often they don't know how many hours of the buffer they allocated toward production support items in the sprint to begin with.

The hours the team is putting toward getting to the goal start exceeding 8 hours in a day and 40 hours in a week, and they aren't seeing their burndown match. They'll get into that frantic mode of the death march to try to get everything done. When a root cause analysis is done, often the fact that production support work was more than anyone planned for--and no one could see that happening--is a critical piece of information.

My preferred practice is to have the team include a "bucket" of hours in the sprint plan for critical production support issues. This brings the hours to be spent on critical fixes front and center, visible in the iteration plan and easier to track. The size of the bucket is based on their experience of the level of production support they have typically provided. It is discussed and agreed on. The product owner or product manager is responsible for prioritizing issues and deciding if it meets the criteria of "critical" enough to go into this iteration's bucket. It is important that the level of hours remaining be tracked closely. If the bucket becomes empty, the next time a critical production support item comes in, then something needs to be dropped off the iteration plan.

Having a capacity plan with a buffer is an important technique to help teams be successful in meeting iteration goals. This in turns plays into providing the predictability that organizations need in meeting large strategic goals. It's a small thing, planning for and respecting a buffer in an iteration, that helps lead to great results.

Tamara Sulaiman, PMP, CST is Managing Consultant at Applied Scrum, where she is focused on coaching teams and organizations transitioning to Agile software development. Tamara brings over 20 years of experience in management across a spectrum of industries including: information technology, construction, international development and education to her consulting expertise. She is a Certified Scrum Trainer (CST) and Project Management Professional (PMP). As a Managing Consultant, Trainer, Coach, ScrumMaster, and Project Manager Tamara is focused on coaching teams to effectively provide value to key

stakeholders and customers through the frequent delivery of software. Her passion lies in helping organizations plan and implement Agile transitions at the program level. She is currently serving on the Board of Directors for the Agile Alliance, and as stage producer for the Agile 2009 conference committee.

Copyright © 2019 ProjectManagement.com All rights reserved.

The URL for this article is:

<https://www.projectmanagement.com/articles/247726/Capacity-Planning--Respecting-the-Buffer>