

Barely Sufficient Project Management: A few techniques for improving your scientific software development efforts

Date: September 13, 2017

Presented by: Mike Heroux, Sandia National Laboratories

The presentation material along with the video will be made available in two places:

<https://exascaleproject.org/event/agile/>

<https://ideas-productivity.org/events/>

Preface to my responses: My best overall advice regarding Kanban and similar tools is:

- 1. Get started now and try simple ideas first.*
- 2. Going too formal too quickly can significantly reduce your productivity.*
- 3. Regularly evaluate whether you are getting value from your new strategies. Adapt as needed.*
- 4. If you find a new effective practice, make it a habit. On your way to making it a habit, you will stumble. Just start again, recognizing that building habits takes time.*

Context: a smallish (academic) research group. One Kanban board for the team, or one for each team member, or both?

I think both team and personal Kanban boards are useful. I use a personal Kanban board to manage both my work and personal activities. I have many team Kanban boards, one for each project I lead, and even have two boards for a particular project where some information must be kept internal and the rest can be open.

When I started using Kanban, I thought cross-posting issues (having my tasks posted on both my personal and team Kanban boards) would be very important. But I have found that how I describe a given task on my personal board is different than how I describe the same task on a team board. I don't mind the apparent redundancy.

I have also found that one comprehensive Kanban board for a team (as the team increases in size) can become unwieldy, so that having separate boards for distinct subproject can reduce complexity.

If you're working with a large team comprised of smaller teams, how do you ensure the small teams adhere to the large team policies?

One way is to have a small team policy that states that large team policies will be observed. If there is a lack of conforming to policies (large or small) this can indicate that the policies need to be further discussed and refined. The best way to assure conformity is to get team members to see the value and make a personal commitment. If that doesn't work, you can consider revising the team incentive system (how people are recognized, compensated, rewarded, etc) so that conforming is desirable.

It is of course possible to enforce policies, but I think this should be a last resort. People in our community are in high demand and can find work in another group, if a team culture becomes too difficult. Steve McConnell makes a point that leaders should treat their team members as though they are volunteers. While this is not our actual arrangement (we are getting paid), the volunteer philosophy recognizes that we get the most from team efforts by creating this spirit on the team. Highly skilled people who are treated as volunteers tend to produce the most and become part of a synergistic environment.

Do group members add bits to backlog with person(s) on the team that "should" work on it?

While the Kanban advice I have seen does not recommend pre-assignment of backlog task, I think it can be a reasonable idea in scientific software. Our teams are composed of people with highly-specialized scientific backgrounds. There is often one or maybe two people on the team with the right background to take on a given task. This situation is not typical for most agile teams, where a given task can often be handled by one of several team members. In fact, the dynamic nature of task assignment is part of being agile. Scientific software teams often don't have the same agility in this particular area.

Which of these tools (trello, etc.) are affordable / open source?

Trello is free for small scale use. I use it for my personal Kanban and am well within the limits of free usage. JIRA has a free trial, but is otherwise not free. The cost may be acceptable, depending on your tastes. Confluence, used for collaborative content management, is similar to JIRA. All three of these products (and more) are available from Atlassian (<https://www.atlassian.com/try>).

Gitlab and GitHub provide free publicly-visible issue/task tracking, and support constructing a Kanban board. Private repositories and boards require an academic license or a subscription.

There are many tools for managing tasks and constructing Kanban boards, so it is worth spending some time exploring available tools.

If your workload comprises a mix of long term development and quick, urgent issues (say user support). How do you manage those urgent issues if they conflict with the number of tasks you have in your ongoing column?

I like to separate urgent task from important but longer term activities. The reason for separation is to be able to monitor and decide what fraction of your efforts are going to be committed to urgent activities. In my experience, if a team wants to optimize its productivity, urgent activities cannot be allowed to consume an unconstrained portion of team resources. If urgent issues dominate team activities, the long term activities will eventually become urgent as well, leading to an unhealthy team environment in the long run.

Which [framework] works well with GitHub and other software systems?

If by framework you mean Scrum, Kanban and similar process frameworks, they can all work in combination with the GitHub software platform. Having said that, I find GitHub's tools and policies to be particularly focused on small team projects. Larger projects can certainly use GitHub effectively (the Trilinos project is large but we use GitHub and it is the right choice for us overall), but at some size, I think Atlassian platforms are superior. The DOE Exascale Computing Project (ECP) and the ACME climate project use the Atlassian Confluence (structured documents platform) product and JIRA (issue management). These two product are built to serve very large, distributed projects well.

Is this [slide 26] a Kanban board or Trello?

Slide 26 is a Kanban board constructed using GitHub's Project tab. This tab is part of any GitHub repository. You can construct any number of Project boards and define the columns as you like. My example uses GitHub Projects to manually construct a Kanban board.

Trello provides a similar capability with similar flexibility to define column headings as you like, including creating Kanban boards.

Is this [slide 26] something that Github provides? Do you know if there is something like that on Bitbucket?

Slide 26 is a GitHub Project board, very easy to set up. The instructions on the last few slides of my talk go the the process of setting up a board like the one on slide 26.

Doing a quick search on Kanban for Bitbucket, I see that Trello (just recently purchased by Atlassian, the makers of Bitbucket) is integrated into Bitbucket, so it appears that Bitbucket has a very Kanban capability. Here is a link to the discussion:

<https://bitbucket.org/product/features/trello-boards>

What do you do if you're on multiple teams, each with their own kanban board? How can you give all teams the ability to see your total tasking?

I have not found a tool that supports cross-team viewing, other than the basic approach of making Kanban boards visible to other project team members. I don't know of a meta tools to manage multiple boards across distinct projects. However, I think it might be easier to discuss with your project leaders to pre-determine a portion of your time for each project. This way, the leader of one project knows they have a fixed portion of your time. Also, since you are the person who decides when you are ready to pull a new task into the In Progress column, you will have knowledge of your tasks for other projects and can make an informed choice about when to bring a new task into the In Progress column.

Mike mentioned that he uses Trello. This is a Kanban board. Does he transfer info from one to the other?

Yes, I manually manage tasks across my Kanban boards and across columns. One nice feature of Trello is that I can send an email message to a Trello email address and the contents of the message will be added to my Kanban board automatically. I use this to help build my backlog of tasks.

This [slide 26] looks like a Team kanban board. Is it helpful for each team member to have their own board? Both?

See first question.

How do you enforce the “maximum number of cards in progress” when you have multiple projects with each their own kanban board.

I do not try to enforce maximums at this point. As teams mature in the planning process, they can become good at uniformly scoping tasks, knowing maximums and measuring their team velocity (the rate at which tasks are complete). I am not at the level of maturity. Instead, I look at my current In Progress task and try to be honest if I can take on another task or not.

When working across several projects, you can allocate a target portion of your effort for each project and calibrate your In Progress task list for each project based on the target fraction of time allotted.

Do you have separate Kanban boards for different projects?

Yes, see discussions above.

I'd like more info about managing a team across multiple repos/Issues sets. Does managing multiple Kanban boards become clumsy - e.g. needing to sum the “In-progress” columns to keep to the predetermined limit?

I have not found managing multiple Kanban boards to be cumbersome. I have a target portion of effort for each project and calibrate the number of In Progress task to the portion of effort I have allotted.

How do you determine the right size for one task in Kanban?

Right-sizing comes with experience. There is a lot written about this topic. Much of the guidance is drawn from commercial or enterprise software experience, so our community should take this advice with some caution. It is best if basic tasks have a similar level of effort. The level of effort is sometimes measured by "story points", which are a kind of currency to describe relative cost.

Generally, according to some of the Agile methodology literature, humans are very poor at estimating actual effort levels for a given task, but we are fairly good at relative sizing of tasks. So I don't try to right-size in an absolute sense, but try to define my tasks so that they represent about the same amount of effort.

Warning: This epic-story-task discussion is probably too specialized if you are just getting started: *To give you some sense of the advanced concepts in defining and managing tasks, there is a hierarchical task framework called Epic-Story-Task, which enables aggregation of simple tasks (sometimes called sub-tasks) into larger "stories", which in turn can be aggregated into an "epic". with the following definitions:*

- **Epic:** *A collection of related stories of a common theme. Attributes include:*
 - *An aggregation of individual stories whose value is more than the sum of the parts.*
 - *A large effort that takes a long time to complete (i.e., implement new major feature set that takes 1 year) or may be unreachable target (i.e., optimized runtime performance).*
- **Story:** *A well defined task or other increment of value with a bounded execution time (i.e., a well defined start time and completion time, which defines the "cycle time"/"lead time" metric in Kanban). A story should have a clear "Definition of Done" and should be reviewed before being considered "Done". Other attributes of stories include:*
 - *The story description is written in the vocabulary of the user or client, not the developer.*
 - *The span of time and effort of a story can vary greatly from a few hours to several weeks.*
 - *If a story is longer than a few months, shorter stories should be identified if at all possible and, when appropriate, the original story promoted to an epic.*
 - *If a story has a simple subtask translation, subtasks can be listed in the story description. In this case you would not need to have explicit subtasks, described next.*
- **Subtask:** *The optional breakdown of a Story into more targeted tasks assigned to specific individuals to help complete the story. Other attributes of subtasks include:*

- *A subtask should be written using developer vocabulary.*
- *A collection of subtasks serves to describe the work being done to complete the story and serves as the translation from requirements to specifications.*
- *The number of subtasks can be very few (even just one) or quite a few, if the story is long.*

How to set up the GitHub Kanban board? Just use issue labels?

You can follow the steps mentioned on the last few slides. Basically, from the main page of a given GitHub repository, select the Project tab (it should be in about the middle near the top of the page). Then create a new project. Then create columns in the project and type in the headings you want for each column, starting with Backlog, then Ready, then In Progress, In Review and Done.

Can backlog items in the Kanban system (for example on Github, or JIRA) be given different priorities/order?

In my experience, backlog items are not sorted within the backlog column. Instead, they are reviewed for readiness on a periodic basis, and moved into the Ready column if deemed ready. From the Ready column, any task can be promoted to the In Progress column, using a variety of criteria.

Is Trello integrated with Bitbucket?

Yes, this seems to be true. See discussion above.

Maybe I missed it, but can you use both GitHub projects and trello together (i.e. sync)?

I am unaware of this kind of syncing. But I do not find that I miss it.

Can you describe more how you run a meeting from a Kanban board?

Sure. Here are the steps:

- 1. Prior to starting the meeting, I ask team members to review their Kanban tasks, making sure that their status is up-to-date and each is in the right column.*
- 2. During the meeting we go through the In Progress tasks one-by-one, asking if there is any discussion. Often, there is little discussion because the status of the task is already up to date and can be read without discussion. However, sometimes the status needs clarification.*
- 3. Next we discussion tasks in the Ready column, to see if there is discussion or can confirm that the task is actually "done done" and can be moved to the Done column.*

4. *Next we discuss the Backlog column. We review for tasks that can be deleted (no longer relevant, called “trimming the backlog”), tasks that can be promoted to Ready, because we know how to complete the task once a slot opens up, or we can directly move a task to In Progress if it is both well-defined and important to start.*
5. *Next we review the Ready column if there is an open slot in the In Progress column.*

We do this for each Kanban board, if there is more than one related to the project.

Meetings can include other discussion, but most of the time is spent referencing the Kanban boards.