LA-UR-19-25789
Approved for public release; distribution is unlimited.

Title: A Lightweight Agile Process for Project Management

Author(s): Herring, Angela M.

Intended for: Report

Issued: 2019-06-21

A Lightweight Agile Process for Project Management

[1]**Angela Herring**
[1]*Los Alamos National Laboratory, Los Alamos, NM*

June 21, 2016

Example Scrum Processes

This document provides examples of a complete "scrum" process for two different kinds of teams. No matter which team type you are, it is the scrum master's job to make sure the team is functioning well. The scrum master must be flexible and able to handle a range of situations, adapting the meeting process to the situation at hand. This handout is meant to give teams a starting place to develop their own process and is based off of two teams I am currently working with as scrum master. Names have been changed to protect the not so innocent.

Why Use This Process?

The goal of using a "scrum-like" process is to create a cross-functional, self-organizing, agile team.

A self-organizing team manages its own process and work. Therefore, the team is not reliant on being told by its leader exactly what to do and is able to adapt to a variety of situations such as missing team members, changing priorities,etc.

Cross-functional teams are composed of members who can work across the breadth of the project. This does not mean that every member is an expert in every area, but that everyone is familiar enough with each area to feel comfortable (with assistance from other team members) doing work in each area.

Agile teams can adapt to changing requirements and project goals. By setting short-term uninterruptable goals (sprints), you ensure that work progresses in manageable parts, and the team is not interrupted by short-term fire drills. However, the direction of the project is free to change radically between sprints.

Who Owns the Process?

The process is a tool adapted by and owned by the team. It is important that the team feels empowered to change the process and that the process is helpful for them. If the team feels that the process is burdensome, it will not work for you.

The scrum master **guides** the team through creating a process based off of a loose framework and following some basic principles.

The Scrum Master's Role

The scrum master is the team advocate, process guide, and meeting facilitator. The scrum master helps their team create an effective process.

By facilitating all the ceremonies, the scrum master is able to make sure meetings run on time and on topic–a crucial part of ensuring the team doesn't get bogged down by the process. The role as advocate is equally important. A scrum master needs to be willing to go to bat for their team, whether what they need is a better workspace or less pressure or different expectations from management.

The rest of this document will discuss how a scrum master can guide the process evolution.

What Kind of a Team are You?

Here we discuss two teams with different processes to show that the process can be adapted for different team types.

Team type 1

- **Scrum master/Technical lead**: manages process, acts as team advocate and is the technical lead, participates in technical contributions to the team
- **Product owner**: higher level manager with oversight but less daily involvement; chooses the gross direction of the effort with input from scrum master/technical lead
- **Customer representative**: scrum master/product owner interface with customer representative on regular basis–little formal daily involvement (customer is too busy, so we must make the collaboration work on their schedule....)/invited to participate in demos

- **Team**: Experts in field of interest, but with different emphases. Computer science experts, numerical analysis experts, etc. All can do work (often with guidance from the main technical expert) in each field.

Team type 2

- **Scrum master**: mainly manages process and acts as team advocate
- **Product Owner/Technical lead/Customer rep**: Technical expert, often provides guidance and mentoring to team, may participate as team member occasionally or all the time. Works directly with the customer to determine needed features.
- **Team**: Expert software developers, some subject matter experts, but not necessarily expert in the field of interest (expertise in related fields). With guidance from the technical lead, able to contribute to all aspects of project.

How did my team get this way?

Team type 1: The technical lead has experience with agile processes and would like to manage both the process and technical direction of the project.

Team type 2: the technical lead of the project asked for help managing the team, but needs to maintain the technical leadership role. An outside person with Agile experience is brought in to be the scrum master.

Must I have separate people for each role? No. This is another of the deviations from formulaic scrum. You must, however, be sure that all roles are covered or your process may founder.

Must my staff be full-time and co-located?
No. This is another deviation from strict scrum. An Agile process should fit your team–you should not have to fit your team to the process. Each step of this process accounts for the fact that your team may be part time (maybe very part time) and not co-located. The goal in evolving the process is precisely so that the teams can accept contributions from every team member, even those who are with your team as little as 10% or 20%.

Should my team members be experts or generalists?
The most effective teams have members that can cross-cut all of the technical areas of a project. This does not mean that one team member is a drop-in replacement for the others, but rather, that, as scientists, they are curious and willing to work on things outside of their areas of expertise.

Rather than looking for technical superstars to solve a problem within a silo, look for these curious people with a good grounding in an area of interest to

the project. As the team members work together, it becomes clear that "Jane" is a numerics person and "Jill" is a computer scientist. The method encourages "Jane" to jump in and do some computer science work (even though it might take longer) and "Jill" to code a few numerical algorithms. In each case, "Jane" and "Jill" will have to talk about each person's expertise. In this way, you ensure that you are not overly dependent on one person for a whole area of your project.

Sprint planning meeting (2 hours)

Set a sprint goal.

Guiding principles:

- Scrum master must take into account staff availability. Although ideally, the team is more than one deep on all functions, often, a key person is needed for technical consulting, even if work can be shared throughout the team. For example, we might need a mesh expert to give the team an idea of what kinds of operations are critical to a mesh data structure. The team can then work together to design a data structure in one sprint, then in the next sprint the work can be divided among the team pretty evenly.
- The goal should be concrete and achievable. i.e. Complete feature A and do performance profiling on feature B. Write it down and (if possible) display it where your team can see it. '

This can be accomplished several ways:

- **Team type 1 suggestion**
  The scrum master and product owner (who might be a higher level project/program manager) meet beforehand to discuss several possible goals. This can be a very informal, brief meeting. The scrum master (the technical lead as well as the process lead for this team) may come back with a list that looks something like:
  - Focus on performance of current implementation of software
  - Design distributed parallel version of current software
  - Design an interface reconstruction algorithm

  These ideas can then be presented to the team and the team can choose based on their availability/interest (i.e. the technical expert for interface reconstruction is unavailable during this sprint, so let's focus on performance...). The product owner agrees that the team can decide the goal from this list and may or may not attend the planning meeting, though the scrum master will meet with the PO afterwards. This method is best

for teams with the experience to design their product with minimal guidance. Often the team's expertise is needed to tease out the finer points of what needs to be done to achieve a given goal (i.e. a research team) **Attendees: team, scrum master, product owner (optional)**

- **Team type 2 suggestion**
  The scrum master (who is mainly the process lead for the team) and the product owner (who has technical expertise in the area, but will not be very involved in the actual creation process) meet beforehand to discuss possible goals. In some cases, the product owner has enough knowledge to both set an appropriate goal and produce a draft backlog. This can be the start of the sprint backlog, although the product owner can bring this as his/her own contribution to the planning meeting. This method is best for teams that have a concrete goal in mind or more time pressure and where the product owner is an expert in the area of the actual product. **Attendees: scrum master, product owner, team**

Additional Guiding principles team type 1:

- Scrum master works with team to break the larger product into sprint goals. These may be decided on a per-sprint basis or a couple of sprints at a time.
- Facilitate a discussion amongst the team members to choose a sprint goal (no more than 15 mins). Remind them of the overarching goal and help them to choose among a couple of options. i.e. would you rather work on performance this time or should we focus on a new feature?

How to use your backlog

A backlog is a tool for organizing a team's work. As such, every team uses it slightly differently. Some teams keep the backlog updated fully with all the tasks they hope to do for a year, while others keep only a few items outside of the current sprint.

Guiding principles:

- You as scrum master, should always make sure that there is enough work on the backlog for folks to pick up some extra work if they have the time. Also use the backlog it to catalog bugs that will need to be worked on eventually.
- Be sure you have some mechanism in place to clean out your backlog every sprint or it will soon become useless.
- Encourage your teams to take ownership of their own backlogs (yet another deviation from the process) so that everyone has a good idea of the project's priorities.

Maintain your backlog

Options:

- Product owner maintains and orders the backlog
- Team owns and maintains the backlog

Option 1 is discussed in the scrum literature, so I will focus on option 2.

In this method the team owns the backlog (although the product owner still gets to decide the priority). In both team examples, the teams keep tasks and bugs in their backlogs, but the backlogs do not extend much past the current sprint. The backlogs are formed in the sprint planning meeting, and the only things on the project backlog (as opposed to the sprint backlog) are tasks that are discovered during sprint planning, but not able to be taken in this sprint.

Each team must devise a way to clean out these old tasks, or the backlog will get cluttered with old work. Team 1 came up with an innovative solution to this problem:

At the beginning of each sprint planning meeting we spend about 15 mins going through the backlog items in JIRA and clearing them out. Other teams may wish to have a separate meeting for such a process, or the scrum master or product owner may have a pass at removing tasks, although this would detract from the team's ownership of the backlog.

Make a sprint backlog (up to .75 hour)

Supplies Needed:

- Super sticky post-its
- A timer
- Pens

Process:

1. Write the sprint goal on the board.
2. 10-15 mins: Ask your team members to silently write down all the tasks (one per sticky, things that can be done in no more than 4 work days, ideally in one work day) they believe are necessary to accomplish the sprint goal. Additionally, your team may add tasks that are not relevant to the current sprint goal, but are important to do.
3. Remove the duplicates. Ask the team to put all their tasks on the board, putting duplicates together in stacks. Look through the duplicated items and choose the one sticky which best describe the task to keep for the backlog, removing the other duplicate stickies.

Quickly Estimate a Backlog (15 mins) (based off Hi-Lo story showdown–see handout)

1. Make a big open space on the conference table and lay out the following header cards for columns: Already Done (0), Easy (.5), Medium (1), Hard (8), You must be kidding! (100), More information (??).

2. Have the team members (without discussion) each grab a few stickies and put them in the columns they feel are appropriate. Other team members may re-arrange stickies that are already placed.

3. Once the notes stop moving around, you may have the team order the tasks within a column.

4. Stickies in the 100 column are an indication that a task is too large. Take these and have the team break them up into smaller tasks which can fit in the other column.

5. Stickies in the more information column are an indication that the task is unclear. Have a discussion about the task and then place it or break it up as is necessary.

6. Remove the tasks that are irrelevant to the sprint goal and put them on the product backlog.

7. **Mark all the tasks that block other tasks.** You might want to set dates that these tasks are due. Otherwise someone might turn in a blocker on the last day of a sprint. Not good.

Take the Tasks! (5-10 mins)

You made it! The planning meeting is almost over, but the manner of the task taking is important.

1. Give everyone a few minutes to silently look over the tasks. Remember no one's name should already be on a task. Anyone can choose any task.

2. Have each team member write on the board how much time they have to devote to this team during the sprint (in days).

3. Have everyone take the tasks that they believe they can complete in this sprint.

4. Double-check that all the necessary tasks for the sprint goal are taken.

5. Have everyone read their tasks aloud and have the team comment on whether they have taken too much (don't worry if someone takes too little, they can always get something off the backlog).

Name the Sprint (5 mins)

This is a great morale builder, but not totally crucial to the process. After all you've been through, why miss out on the fun? Each team should have a unique naming scheme, which can follow a sequence (eg AA, BB, CC...) and with some pattern i.e. place name / action. Have your team write up their proposals on the board and vote. The sillier, the better. Remember, your management will have to say these names in meetings....

Scrum Meetings (< 15 mins)

The daily scrum is a time for your team to give a very quick status update. The only people free to talk at this meeting are the scrum master and the team members, though others are welcome to observe.
Each team member answers three questions:

- What have I done since we last met?
- What am I planning to do before we meet again?
- What are the obstacles to my productivity?

As the team gets good at this format, you will find that the scrums last considerably less than 15 minutes.

How is it possible to keep this meeting so short? Since everyone has been involved in the planning and goal-setting phases of the process, the team has a shared context and needs less explanation time.

Often the scrum meeting is a great place to catch your other team members, especially if you are not co-located. If a topic requires further discussion, request (during your status update) a side meeting for after the scrum to discuss the topic further. In this way, only the people interested in this topic stay around to listen.

How many scrum meetings should my team have? In traditional scrum, teams hold scrum meetings daily. For a part time team, this is usually overkill. Teams should meet at least twice a week for a daily scrum, preferably in person.

Must all the scrum meetings be in person? Teams should meet in person at least once a week (more if the team is closer to full-time). This enables the team to have the informal and very necessary (especially when the team is not co-located) side meetings. Having a teleconference number is helpful as well, so that team members can keep up with goings on while on travel, etc.

What is the real point of the status meetings? Can I (the scrum master) just get the updates from each person individually? The main

point of the status meetings is to give the team time together so that collaboration meetings happen. Without these meetings, your team is in danger of becoming isolated performers rather than a cross-functional team.

End of Sprint Demo (1 hour)

The end of sprint demo is where your team gets to show the work they have completed to everyone. This is an informal meeting, and the focus is on showing working code. Invite everyone who might be interested in seeing your team's progress.

Guidelines (adapted from Mike Cohn's scrum presentation):

1. Show working code.
2. No PowerPoint.
3. <2 hour preparation time.
4. Leave time for audience Q/A for each presenter.

Retrospective (30 mins)

The retrospective is a time for the team to adapt the process. There are whole books on this topic, so be aware that this is not the only way to do a retrospective.

A simple retrospective

Supplies:
Whiteboard for list making

1. Have the team make a list of all the things that went well in the sprint. The scrum master writes these on the board.
2. Have the team make a list of all the things that could have used improvement or were puzzling in the last sprint.
3. Have the team suggest and agree on solutions to the issues in 2.
4. Go through the list of improvements from the last retrospective and determine whether they worked. Suggest new solutions if not.
5. Have the team offer appreciations to one another. The appreciations are of the form "I appreciate you (Name) for (what you appreciate). It really helped me to (what it helped you to do)." This format is from Jerry Weinberg.

The scrum master should keep the data from the retrospectives, especially the "needs improvement" data.

Is the retrospective one of the parts of the process I can remove? It sounds awfully touchy-feely.
No!! Do not be tempted to remove the retrospective from your process. The retrospective is crucial to adapting the process for your team and discovering problems within your team.