

Agile team organization in pharmaceutical data science

Why we should be moving lots of small rocks rather than the one big mountain

Joe Zhu, PD Data Sciences, Roche Shanghai
Daniel Sabanes Bove, PD Data Sciences, Roche Basel

Abstract

During our work on data science software projects in Roche we have been practicing agile team organization. We have also integrated some of the agile practices and mindset into our daily work on clinical studies. This paper will share our personal perspectives on the basic ideas, processes and tools for using agile team organization in pharmaceutical data science, and how it can lead to more satisfactory and efficient team work.

Introduction

“The foolish old man removes the mountains” is a well-known story from Chinese mythology about perseverance and willpower. In this story, the old man and his family planned to remove the mountain by moving rocks, and he planned to achieve this beyond his lifetime with help from his descendants. Changing the context of this story by replacing it with complex task management, “moving rocks rather than moving mountains” is exactly the essence of the “agile” mindset. In this short paper, we will discuss how our data science team practices “agile” at Roche.

Methods

The biggest mis-conception of “agile” is that people think it is a tool or a method to be applied. People often phrase themselves “agile” when they use a kanban board, or running a daily stand-up. The word “agile” is a mindset, rather than a tool or a ceremony. However, “scrum”, “kanban board”, “retros” are significant components when being agile. In this section, we will give short introductions to these tools/concepts, and elaborate their application in pharmaceutical product development with practical examples in the next section.

The Agile Manifesto

One important reference for the word “agile” is the Agile Manifesto (<https://agilemanifesto.org/>, 2001), which is emphasizing the components of the agile mindset:

We are uncovering better ways of developing
software by doing it and helping others do it.
Through this work we have come to value:

Individuals and interactions over processes and tools
Working software over comprehensive documentation
Customer collaboration over contract negotiation
Responding to change over following a plan

That is, while there is value in the items
on the right, we value the items on the left more.

Notably, the Agile Manifesto also clarifies that processes and tools alone are not the most important. Instead, individuals and interactions, i.e. the practices and mindset being applied, are valued more.

The origin of the agile movement was in software development, in particular to carve out an alternative way of working compared to the traditional documentation driven, heavyweight software development processes. Nevertheless, we will show in this paper that this agile mindset can likewise be applied in other pharmaceutical data science projects.

Scrum is Iterative and Incremental

Scrum (Sutherland and Schwaber 1995) is a framework to help teams and organizations generate value through adaptive solutions for complex problems. Projects are carried out incrementally with scrums, which is usually referred to as Project Increment (PI). It includes several key components:

1. Planning stage.
2. Sprints.
3. Daily stand-up meetings and Scrum of scrums
4. Kanban Board
5. Retrospective

PI planning

PI planning is the most important part of a PI. At the planning stage, teams and members need to:

1. Set objectives and deliverables for the PI.
2. Refine objectives and deliverables to actionable tasks in detail.
3. Measure team's capacity and bandwidth in order to assess if all objectives are achievable during the PI.
4. Identify key blockers and dependencies of the tasks, and plan their execution.

6. Identify key risks and potential problems that the team may encounter during the increment, and assign risk owners to remediate these risks and issues.

Sprints

A PI is carried out over a period of time, and usually divided into 3 to 4 segments. Each segment is referred to as a sprint, and typically lasts for two weeks. Within each sprint, the team members align the objective within the two weeks, and carry out the execution. The key to success is to refine tasks into actionable tasks. These tasks can be incrementally implemented if the sequence of execution is necessary; otherwise it is recommended to take an iterative approach, adapt and adjust to changes (see Fig. 1). In agile development, we thus conjoin the two tactics of incremental and iterative progress, depending on the nature of the tasks.



Fig. 1: Top shows an incremental development of the Mona Lisa picture: Starting from a final idea, each increment adds another square of the image onto the canvas, until the picture is finished. Bottom shows an alternative, iterative development of the same image: Starting from a vague idea, each iteration builds a complete picture, evaluates it, and then refines the same picture in the next iteration. (Source: Jeff Patton, https://jpattonassociates.com/dont_know_what_i_want/, CC-BY license)

Stand-up meetings and Scrum of scrums

Daily stand-up meetings are meant to be efficient. The origin of the “stand-up” terminology is that research found that meetings usually take longer when participants are sitting down and feeling comfortable. One way to conduct a short meeting is to have the meeting while standing up. At the stand-up, the team quickly updates everyone's progress and identifies blockers. The team will have break-out follow-up meetings for more discussion if needed. A scrum of scrum can be seen as a higher level of daily stand-up: team representatives will gather once (or twice) each sprint, and share with other teams the progress, blockers and risks.

Nowadays most teams will have members based in different locations and stand-up meetings will be performed via video calls, hence other means, e.g. scheduling short time slots (e.g. 15-30 mins only) help to maintain the efficiency of the stand-up meetings.

Kanban Board

A kanban board is a tool to help teams to communicate and keep track of work progress. Minimal kanban boards comprise the following columns:

1. Todo
2. In progress
3. Done

Most software development work requires collaboration. Therefore, kanban boards often contain a “Review” column to keep track of the task status before approval and moving them to the “Done” column.

In order to make the prioritization of tasks clear, we also often use a “Backlog” column to place all tasks initially. Afterwards, we move currently prioritized tasks into the “Todo” column to start their implementation and execution. The key to ensure the “agility” of moving tasks across the board is to ensure good refinement of each task, and making sure the task’s instructions are clear and concise. In order to ensure the flow is smooth, it is important to avoid blockage from downstream. Therefore, one should always prioritize reviewing a task waiting for a review over starting a new task. See Fig. 2 for an example kanban board.

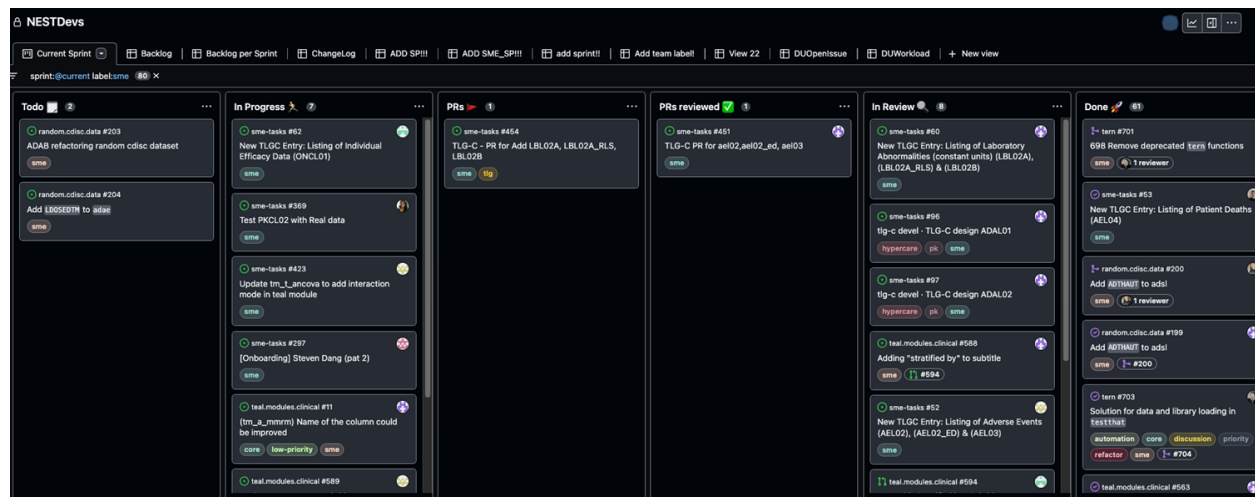


Fig. 2: Example kanban board (Source: NEST Project)

Retrospective

A retrospective is a scrum ceremony where team members gather to review and discuss at the end of a phase, which could be:

1. a PI planning

2. a sprint
3. or an increment.

The goal of a retrospective is to revise and adjust the team's agile development approach. At the retrospective, team members:

1. Acknowledge each other's work, identify the positives and successes.
2. Identify things that didn't go well or didn't go as planned.
3. Identify actions that should be encouraged and strategize improvement for the future.

Retrospectives are important as they ensure the team's mode of operation evolves and becomes more efficient over time.

User Stories

Next, we are going to describe three user stories from our practical experience, to illustrate that the agile approach can be tailored to the project at hand.

Case 1: NEST project development

NEST is an open-source software development project to support clinical reporting for both regulatory and exploratory work in R. The project was initiated in 2018 at Roche, and has open-sourced more than 20 software packages since.

The NEST project team is made up from 4 core product teams:

1. Libraries and packages to support interactive tool development
2. Libraries and packages to support static output creation
3. Table, listings and figure template creations
4. Packages integrations, deployments and releases

Each of the teams are made up with their own product owners, engineering leads, programmers, and operate with their own daily stand-up, kanban board, and retro. However, the entire NEST team shares the same development cycle, and leverages the shared PI plan to identify inter-team dependencies and risks.

Thirty-three team members are scattered across 6 different time zones. Managing a global agile team is challenging – while making sure effective and efficient communication within and between teams, we also need to be mindful of team members' working hours and work-life balance.

During the PI planning week, we first video-record PI objectives and review them before the planning session to ensure the objectives are aligned before the planning. We then divide the planning sessions into two parts, where European and Chinese colleagues will attend the same

session, and the rest of the team members will attend the other session, where some European colleagues will act as the bridge between two sessions and ensure effective communication.

If the team is small and time zone differences between team members are within 8 hours, the team will conduct normal daily stand-up meetings as usual. Otherwise, we split the team into small groups and join other team's daily stand-up meetings to make sure teams are updated and communicate with each other.

Case 2: crmPack

crmPack is an R-package with a JAGS backend to support model-based dose escalation study design and execution, originally developed and published as open source by Sabanes Bove et al. (2019). Its extensible modular design led to adoption across several other companies. For example, individual extensions to crmPack included additional tests for the validation and additional escalation rules, simulation setups etc. These created effectively new modified R-packages on top of crmPack, which led to higher maintenance burden, as for each new version of crmPack, also the modified package had to be updated.

In 2022 a global, cross-industry collaboration of around 10 crmPack developers assembled thanks to an initiative by the University of Cambridge. This interest group believes in the open source development, collaboration, knowledge sharing and contributing to a common code base that have the potential to define industry standards. The members have common background in clinical statistics and an interest in enhancing dose escalation study designs with modern technologies.

The development team meets every two weeks and discusses ongoing issues and pull requests, which are organized on GitHub and with a Kanban board. This virtual in-person interaction via video chat is very important to create the trust between members of different companies. Quick communication via chat channels is another important ingredient that allows us to address questions and resolve blockages quickly. Given that all developers have competing tasks in their daily jobs, and being spread across several companies, keeping everyone motivated is key. We let developers assign tasks themselves, because they know how much time they have and the difficulty level and topic they would like to tackle. We focus on encouraging and building up skills, rather than micro-managing developers. We also witnessed that the diversity of our backgrounds and skill sets really improves the results of the implementation.

Case 3: Clinical study sky04

Sky04 is a Phase 2 clinical trial in cervical cancer. The study aims to deliver 50 tables, listings and graph (TLG) outputs based on 18 programs in R. Unlike a typical software development team, a statistical programmer team is part of a clinical study team, therefore, the timelines and deliverables are planned ahead. In addition, different outputs are required at different stages, and the time differences between deliverables are not exact even. Therefore, we only divided

the output deliverables into batches, and implemented them incrementally over a time period of time without development sprints.

Summary and Conclusion

To sum-up, an “agile” way of working is about:

1. Moving stones rather than moving mountains.
2. Making plans, reflect and adjust at the same time.
3. There are many tools and techniques to being agile, one should choose the methodology that best fits the situation. We have seen in the case studies that different components can be used and tailored to the projects, see Tab. 1.

	PI/project planning	sprints	stand-ups/regular catch-ups	kanban board	retro
NEST SME	yes	yes	yes	yes	yes
crmPack			infrequent	yes	
Sky04	yes		yes	yes	yes

Tab. 1: Overview of the components used in the three user stories.

In our experience, the following mindset changes (compared to classical data science approaches in pharmaceutical industry) are critical to ensure moving towards an agile practice in the team:

1. Do everything together as a team (“Now I take task A, but you could also have taken it”) - instead of creating task silos in the team (“I can only do A, you can only do B”).
2. Have short, frequent (e.g. daily) stand-up meetings to unblock problems quickly - instead of long, infrequent (e.g. weekly) meetings with status updates from everyone.
3. Ensure all tasks and their progress is visible on a single kanban board - instead of every team member tracking those on separate documents and sheets.
4. Let team members self-assign tasks to match their current time availability and motivation - instead of telling everyone what to do next and micro-managing them.
5. Scale up the team's speed naturally by adding more team members when needed - instead of increasing the pressure on existing team members.

References

- Agile Manifesto <https://agilemanifesto.org/>

- <https://www.knowledgehut.com/blog/agile/safe-agile-vs-agile>
- <https://www.scrumpoker-online.org/en/room/36546439/scrumpoker>
- Sky04 clinical trial <https://classic.clinicaltrials.gov/ct2/show/NCT04300647>
- Sabanés Bové, D., Yeung, W. Y., Palermo, G., & Jaki, T. (2019). Model-Based Dose Escalation Designs in R with crmPack. Journal of Statistical Software, 89(10), 1–22. <https://doi.org/10.18637/jss.v089.i10>
- Sutherland, Jeffrey Victor; Schwaber, Ken (1995). Business object design and implementation: OOPSLA '95 workshop proceedings. The University of Michigan. p. 118. ISBN 978-3-540-76096-2.

Contact information

Joe Zhu
Roche
joe.zhu@roche.com

Daniel Sabanes Bove
Roche
daniel.sabanes_bove@roche.com