

## Applying Agile Methodology to Statistical Programming

Gina Boccuzzi, PROMETRIKA, LLC.

### ABSTRACT

When working in a CRO environment, Waterfall project management methodologies are often the default and seem like the only way to guide a project. While a full transition to Agile-based project management may not always be practical or the right fit for every workplace, many Agile methodologies feature practices that can be easily implemented to optimize daily workflow. These tools include: Daily Standup, a daily team meeting to discuss progress and upcoming work; Sprint Planning/Retrospective, bi-weekly meetings to discuss the work needed in the coming weeks; Post-Mortem, a meeting at the end of a project to discuss how it went and what could have been done differently; and Cyclical Timelines, working on different components of a project at the same time rather than waiting to start one until the last is finished. Regardless of the overarching project management style, these tools can be implemented individually or in tandem, depending on the user's needs. These tools are easy to learn and understand and can optimize programmers' time. Agile does not have to be a practice reserved only for the software engineering world; its tools are broad and can streamline communication and workflow.

### INTRODUCTION

Agile is often presented as an alternative project management style that requires a huge commitment and overhaul of current practices. The purpose of my paper is to break down Agile methodology into a series of smaller ideas that can be implemented at any time by any team member. When used in conjunction, these ideas have maximum efficacy but alone their effect can still be great. As a former Computer Science student, my professors spent ample amounts of time teaching us about the benefits of an Agile approach to work. A software company I formerly worked at had converted all their departments to a fully Agile environment. They were very pleased with the results and found their productivity and time management to be greatly improved. They spent extensive amounts of time teaching employees about the potential Agile held when used properly. My career led me away from software engineering and IT teams but the knowledge and the practices always stayed with me.

### PROBLEM STATEMENT

In this paper, I will be using Waterfall as the juxtaposition to Agile, as Waterfall is the most common default project management style. When working in a Waterfall environment, things can get hectic. You are constantly chasing due dates – once you meet one, you are running to catch the next. This creates periods of lull and hyperactivity, as your projects aren't being paced properly. In a project, changes inevitably arise and the team needs to be able to adapt and integrate these changes. During Waterfall, these changes are found during the final review at the end of the project and leave much to do with little time left. Agile places more focus on reviewing work on an ongoing basis, allowing these changes to be addressed earlier and mitigating the burden placed on the team before a deliverable. Making a full conversion from Waterfall to Agile may not make sense everywhere, but it can help mitigate these challenges and increase productivity.

### WHAT IS AGILE?

At its core, Agile is a set of twelve principles and four core values. These values are:

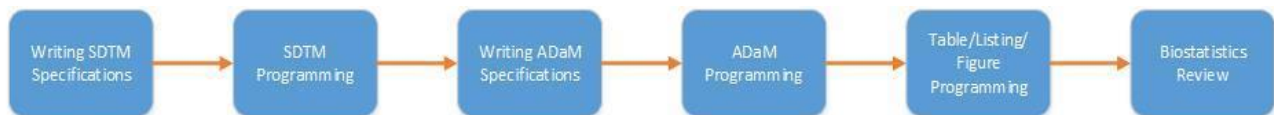
1. "Individuals and interactions over processes and tools" [1]
2. "Working software over comprehensive documentation" [1]
3. "Customer collaboration over contract negotiation" [1]
4. "Responding to change over following a plan" [1]

By leading a project with these core values in mind, the team focuses on collaboration, communication, deliverables, and adapting to change. Any project management system that follows these core values is inherently Agile. There are many project management styles that have come out of these values; the practices I describe will most closely align with the Scrum project management style.

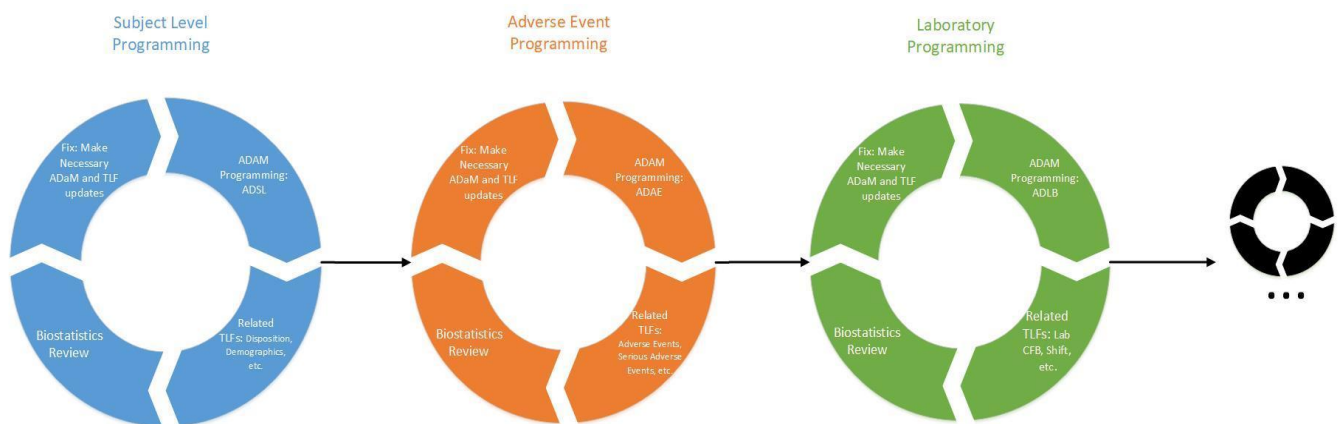
When put into practice, Agile is a cyclical style of project management. A project is broken down into small tasks that are completed and reviewed as the project progresses. Typically, there is a less firm project end date but work progresses on a rolling basis in two-week cycles, with certain tasks due at the end of each cycle. Agile can seem especially daunting when deadlines are due to external clients (CRO) but this does not mean that Agile can't be applied, the tools it implements can still be used with great results.

In a linear style of project management, such as Waterfall, the project moves from one task to the next upon completion until the end of the project is reached. For example, this could look like: Writing SDTM Specification → SDTM Programming → Writing ADaM Specifications → ADaM Programming → Table/Listing/Figure Programming → Biostatistics Review; the next task begins after the completion of the first. Figure 1 provides a visualization of this. There is a definitive start and end date, where the entirety of the work needs to be completed by this end date. These dates are inflexible and thus, the projects have a harder time adapting to unexpected changes. When a large change is needed in a project, this can be very time and work-intensive but the end date remains the same regardless.

The same study from Figure 1 can be run with a more Agile approach. After completing the SDTM programming, the study can work in cycles. First beginning with Subject Level programming – creating the ADSL dataset and related tables. These tables can then be reviewed and any applicable changes can be implemented in ADSL and the tables, before moving on to Adverse Event programming (ADAE and related tables) and so on. Figure 2 provides a visualization of this. Working in cyclical steps like this, periodically reviewing as the project progresses allows for issues to be addressed on a rolling basis and for the initial requirements to be updated. When working in a CRO environment, clients are often unsure of what they do or don't want until they see it. By uncovering these changes earlier, the team is able to adapt sooner to better satisfy the clients' needs. In the Waterfall approach, all issues are typically left to be found during the final reviews; this leads to last-minute changes to ADaM specs and programs.



**Figure 1. Waterfall Project Progression Example**



**Figure 2. Agile Project Progression Example**

## HOW TO LEVERAGE AGILE

Looking at these breakdowns in the context of my own CRO work environment, Waterfall still remains the overarching project management style due to our external clients and deadlines; however, it has room for improvement. Just because being 100% Agile may not be practical, doesn't mean that it can't still be used to your advantage. Agile can be broken up into many different smaller practices. I will be discussing the tools that are the easiest to implement but there are many more available.

### DAILY STANDUP

A frequent feature of Agile is the Daily Standup. This is a brief meeting, less than fifteen minutes, that takes place every day with the entire team working on a specific project. During this meeting, you can discuss what you did yesterday, what you plan to do today, and questions you may be stuck on. This takes the guessing out of teamwork; there is a constant line of communication so team members are able to provide adequate support to each other. For example, you are falling behind on your work for Project A because Project B is requiring immediate attention; you may not reach out to any of your Project A team members because you assume they are all busy with their own work for Project A. If you had a Daily Standup, you could communicate these challenges and the team can decide how to handle it – a team member may have finished their work and be able to take this work from you, the timeline may be long enough that this delay is okay, or you may split up your Project A work across the other team members. When the communication is occurring daily, resources are able to be allocated more efficiently and you may not have to work overtime to make sure you complete your work for Project A and Project B. This tool is very easy to implement and can be done on a study level by a lead programmer or project manager. If the idea of doing it daily seems daunting, try starting with a Monday/Wednesday/Friday or Tuesday/Thursday approach to get used to the additional communication.

### SPRINT PLANNING/RETROSPECTIVE

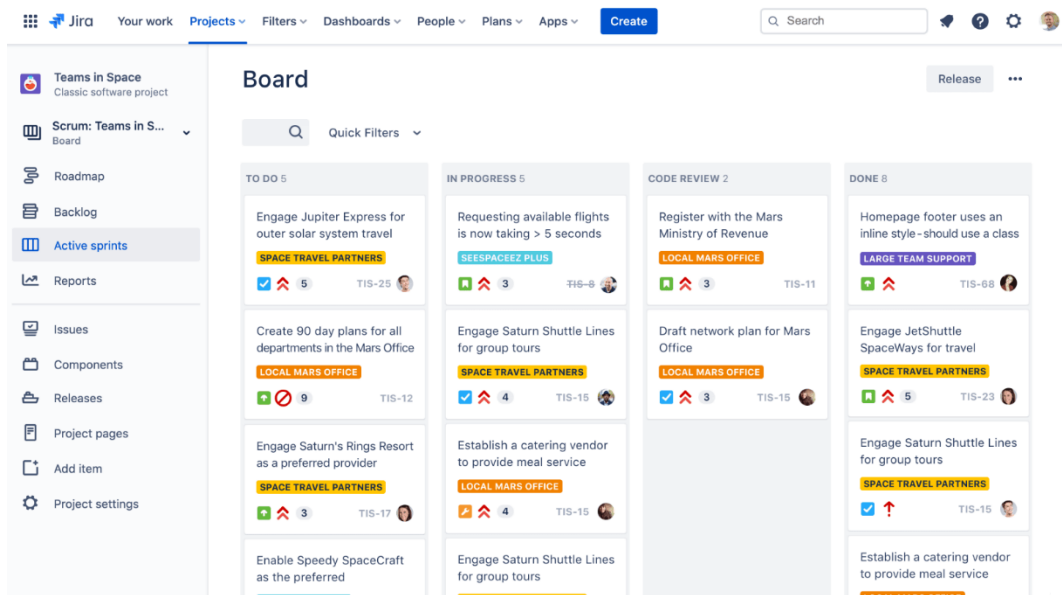
A Sprint is a two-week period of work (give or take a week). At the end of the two weeks, you have a Retrospective to review the work that you were able to complete during the Sprint. You look forward at the rest of the work left for the project and use your output from the last Sprint to gauge how much you will be able to complete during the next Sprint. This can be implemented by creating a recurring Retrospective meeting for the team every two weeks; if you are implementing Daily Standups, then Standup could be extended once every two weeks to accommodate this additional discussion.

Reviewing the pace at which work is completed is critical when trying to meet deadlines. Sprint planning and review can take the guesswork out of this. For example, in the prior Sprint planning, we thought we could complete 3 ADaM domains and 10 tables in two weeks, but at the end of the two weeks, we were able to finish 2 ADaM domains and 6 tables. Looking ahead, in the next two weeks, we had hoped to finish 2 ADaM domains and 15 tables. The team can now discuss if they need more programmers and how to allocate time to ensure that they do not fall behind. By setting these small timelines, the team is able to stay on top of their schedule during the entirety of the project. Alternatively, it is harder to notice these trends and to keep the team from falling behind before reaching the end of the project when there is no time to correct their trajectory.

Sprint planning also allows an individual to learn more about themselves; their skillset and the workload that they can manage. You are forced to evaluate how much work you can take on and complete each Sprint. At the end of the Sprint, you ask yourself: did I over commit or under commit myself this period?

There are many resources available to enable Sprint planning, Jira™ being one of the most popular. Jira™ is a project tracking software that allows tasks to be assigned point values, giving more challenging and time-intensive tasks a higher point value. Programmers are then able to set the points that they can manage within a week and take on tasks until they hit that value. Tools like Jira™ are very powerful and allow the team to visualize all their work and create metrics for the work being completed. Display 1

shows a Jira™ board; the columns display the work to do, currently in progress, being reviewed, and completed. The numbers in the bottom left corner are the points assigned to the task, indicating the difficulty. Having this visualization of work is very powerful when planning Sprints.



**Display 1. Jira™ Project Board Example [2]**

## POST-MORTEM

A Post-Mortem is conducted at the end of a project after all work has been completed. Post-Mortems occur less frequently in software development, but in a CRO environment where projects are regularly being concluded, they can be very helpful. The Post-Mortem meeting looks over the entire span of the project – the successes and failures and how to learn from them moving forward. For example, the team thought two programmers would be able to complete 35 SDTM domains in the period of one month (two Sprints) but they needed an additional programmer and two more weeks. During the Post-Mortem, you would discuss with the team to learn about why this happened: Did we gauge the complexity of the domains incorrectly? Were there unexpected challenges with the raw data? Were the programmers devoting more of their time to other projects than initially anticipated? Working together to find the reasons behind the challenges that arose helps the team to account for these in future project planning and handle them more appropriately.

A Post-Mortem is typically a lengthier meeting, including the team members from your Daily Standup, as well as management that may not have been working on the project daily. Post-Mortems typically result in a set of action items to use during the next project. After reviewing the failures of a project, the team works out definite steps that can be taken to mitigate these problems during the next project. Leaving this meeting with concrete steps to take is a way to hold the team accountable to improve both their workload and the quality of their output.

## CYCLICAL

All the aforementioned tools are specific practices that can be easily implemented as soon as you decide. Changing your workflow to a more cyclical approach can require a bit more work but ultimately, comes down to a change in mindset. Working cyclically means doing tasks in phases, rather than step by step. If every step happens sequentially with specific start and end dates, there is little room for adjustment.

These adjustments can come from changing client requirements, changes in the raw data, or a difference in the programmer and Biostatistician interpretation. For example, you have one month for SDTM programming, two weeks for ADAM programming, one month for TLF programming, and one week for Biostatistics review. At the end of TLF programming, a problem with ADSL arises during the Biostatistics review. The spec needs to be updated and changes need to be made to ADSL, ADAE, and ADLB; all TLFs need to be rerun, and some need to be updated. This leaves a lot of changes to be implemented in the last week of the project. Referring back to Figure 2, if we had been working in cycles, ADSL and the related tables could have been programmed first and already reviewed by Biostatistics at the beginning of the project. These problems could have been caught and addressed at this early point and all the subsequent ADSL-dependent ADaM domains and tables wouldn't need to be retroactively updated. Working in cycles can help distribute the unexpected challenges over the course of the project, rather than saving them all until the end of the project.

## CONCLUSIONS

Agile is not a perfect fit for every company and every field. In a CRO environment where the due dates are to external clients, Waterfall may remain a more natural fit. This does not, however, mean that you must miss out on the increased efficiencies that Agile has to offer. Agile's tools can be introduced in bite-sized increments with great results. These small steps require little to no commitment and can greatly optimize your team's workflow and communication. Agile is an industry buzzword, it is thrown around often and can seem like an intimidating methodology meant for software companies with different project structures but this is not true. If you are leading your projects with the core principles of collaboration, communication, deliverables, and adaptability, you will always be Agile. While some of the tools require some time investment, they will be paid back as projects run smoother. Any of these tools will yield these benefits, whether alone, in conjunction with one another, or in a complete Agile system – so, why not give them a try?

## REFERENCES

[1] Beck, K., et al. "The Agile Manifesto". Agile Alliance. 2001. <http://agilemanifesto.org/>.

[2] Jira™ Software. "Sprint Board". Atlassian. 2022. <https://www.atlassian.com/software/jira>.

## CONTACT INFORMATION

Your comments and questions are valued and encouraged. Contact the author at:

Gina Boccuzzi

PROMETRIKA, LLC., Cambridge, MA, USA

Email: [gboccuzzi@prometrika.com](mailto:gboccuzzi@prometrika.com)