

Why Be Different? Shouldn't Everyone Be Using Agile to Deliver Their Clinical Trials?

Michael Rimler, GSK, Cincinnati, OH, United States

ABSTRACT

In GSK Clinical Programming, we have embarked on a mission to embed the agile mindset and ways-of-working into the delivery of clinical study data artefacts including, but not limited to, SDTM, ADaM, TFLs, and associated documentation. In virtually every area in the world where code is developed to produce something, modernized teams leverage agile concepts and tools to improve performance, oversight, customer satisfaction, and budget compliance. Yet, as a collective in clinical trial analysis, Programming functions have demonstrated resistance to embracing such a disruptive shift in their delivery pipelines.

This paper will discuss a model of agile implementation within GSK Clinical Programming teams - exploring the successes and challenges.

INTRODUCTION

If you run a quick search, you will find all sorts of definitions of Agile, such as being 'able to move quickly and easily'. However, for our purposes, the word agile has been applied in many ways from being agile in resourcing, agile in what work we do next, or being agile in what we deliver to our customer. For this discussion, we will anchor the notion of agile in the [Agile Manifesto](#), which details 4 pillars of value:

- Valuing individuals and interactions over processes and tools
- Valuing working software over comprehensive documentation
- Valuing customer collaboration over contract negotiation
- Valuing responding to change over following a plan.

These 4 pillars underpin the 12 core principles of Agile detailed in the [Agile Manifesto](#), reproduced here and emphasizing flexibility, collaboration, and customer satisfaction:

- Our highest priority is to satisfy the customer through early and continuous delivery of valuable software.
- Welcome changing requirements, even late in development. Agile processes harness change for the customer's competitive advantage.
- Deliver working software frequently, from a couple of weeks to a couple of months, with a preference to the shorter timescale.
- Business people and developers must work together daily throughout the project.
- Build projects around motivated individuals. Give them the environment and support they need, and trust them to get the job done.
- The most efficient and effective method of conveying information to and within a development team is face-to-face conversation.
- Working software is the primary measure of progress.
- Agile processes promote sustainable development. The sponsors, developers, and users should be able to maintain a constant pace indefinitely.
- Continuous attention to technical excellence and good design enhances agility.
- Simplicity--the art of maximizing the amount of work not done--is essential.
- The best architectures, requirements, and designs emerge from self-organizing teams.
- At regular intervals, the team reflects on how to become more effective, then tunes and adjusts its behavior accordingly.

This discussion is not intended to persuade you to jump headfirst into agile. However, I do hope to motivate you to think carefully and deliberately about how one might apply agile principles and ways of working to conventional reporting activities. Before we can begin, however, we must ask ourselves a few questions to establish a baseline of understanding for what follows.

QUESTION 1: WHY DO WE DO WHAT WE DO?

I've seen this question serve as an ice-breaker numerous times in group meetings and workshops related to innovation and change. By far, what I hear most often is "to help deliver life-saving medicines and vaccines to patients as quickly and confidently as possible." This truly is a noble endeavor - we want to help people, and we do that by performing complex data analyses on clinical trial data to help decision makers assess if a drug/vaccine is safe and if it works as proposed. But not everyone is in it primarily for patients. Some people love data. Some people

love to code. Some people love the science. Whatever your purpose is for being in your role in this industry, please keep this in mind as we continue to explore agile methodologies.

QUESTION 2: WHO ARE OUR CUSTOMERS?

One obvious answer might also be the patients and, while that's true, I posit that you are not delivering your analysis datasets to any patients – at least not *because* they are a patient. So, let's think a bit differently. Typically, during the reporting period of a trial, you will be delivering the product of your efforts to statisticians, clinicians, safety groups, and asset owners for review of the analyses. You will also deliver to medical writers for developing the Clinical Study Report and to your regulatory group for preparation for submission. If you work for a service provider (CRO), your sponsor is clearly also a customer. And, ultimately, we need to ensure that what we produce satisfies the Health Authorities' expectations, which will review your data for approval. All of these are our customers, and we need to know what features of what we produce are important to them. Perhaps you can think of other customers beyond what I've identified here.

QUESTION 3: WHAT DO WE PRODUCE?

I'll keep this one simple - but you may think of additional deliverables to augment the list. We produce SDTM databases, complete with database definition files (e.g., define.xml), annotated case report forms (aCRF), and the clinical study data reviewer's guide (cSDRG). Essentially the tabulated data and associated documentation to aid review, use, and traceability. Similarly, we produce analysis databases (ADaM) with their associated documentation (define.xml and ADRG). We also produce data analyses in the form of static summary tables, statistical analyses, graphical displays ('figures'), and patient level data listings. We may also produce interactive data visualization platforms to enable reviews of data and analysis results more efficiently than scrolling through hundreds (or thousands) of pages of electronic paper. We produce these artifacts to enable our customers to make decisions or pick up the next step in the clinical reporting and submission process.

QUESTION 4: HOW DO WE KNOW WHAT OUR CUSTOMERS WANT?

Conventionally, we know what our customers want because we have a study protocol, a statistical analysis plan (SAP), and a set of mock display (TFL) shells with programming specifications. We also have industry data standards (e.g., CDISC) and company data standards, as well as company reporting standards for the display shells. We may also have some idea of priority because of a safety review plan or the identification of top-line results.

In addition, we traditionally get feedback from our customers on whether we are delivering value through Dry Run deliveries. In my experience, these are lumpy deliveries of the full analysis package, usually once or twice before the final analyses are delivered for a reporting milestone. These deliveries enable statisticians, clinicians, and even sponsor-clients to review the work that has been done and ensure that it meets their needs.

Agile principles and ways of working put more priority on delivering value to the customer, pushing us away from dry run deliveries and toward more frequent, incremental partial deliveries and regular interaction with our customers.

APPLYING AGILE TO CONVENTIONAL REPORTING

How might we leverage agile principles to develop concepts, strategies, tools, or processes to create the artifacts we use to analyze clinical trial data for safety and efficacy, eventually submitting to a regulatory authority for review and approval to market a medicine or vaccine? In [Patel and Rimler \(2024\)](#), the authors outline a number of areas where these principles can be upheld by working differently than we have conventionally for decades.

When we think of conventional practices, we should ask ourselves if what we are doing is driving value to our customers in the best way. How might we drive value to statisticians, clinicians, asset leads, and even health authorities better and faster? Are we delivering value incrementally and seeking feedback, or are we relying on pre-planned specifications and delivering at the end? Do our teams communicate regularly? Are our teams empowered - or are they following centrally defined processes? Are our teams reflecting on how they can become better teams?

Agile is more than a project management toolbox for moving sticky notes around a calendar or shuffling deliverables tracked in rows of a spreadsheet. Agile principles and ways of working can help us to deliver value to our customers when they need it. By looking at the package of data and analyses as a product to deliver, through the lens of agile principles, we may be able to better serve our customers and do it with happier teams.

The implementation of agile ways of working at GSK is discussed at a high-level by Patel and Rimler (2024). In the initial phases, as with many change initiatives or innovation projects, we started with experiments and pilots. Some of our key characteristics introduced the Scrum Master role, defined the product and Product Owner, and explored the application of the backlog concept to the work our teams do. Introducing the Scrum Master role was a critical piece because our study leadership roles tended to manage everything: outputs, people, timelines, and prioritization. The

scrum master brought both experience with agile and the ability to take on some of the task management activities for the project. It was also important to define the Product Owner (our Study Lead Programmer (SLP)), in order to assign the overall accountability of what the study team would prioritize and deliver. And, finally, the active management of a backlog tends to be a foreign concept (in my experience) on many reporting teams. We tend to detail out the deliverables (tables, listings, figures), assigning a resource to produce each piece and another resource to perform quality control checks. The 'conventional' Programming backlog is essentially the rows of a spreadsheet of deliverables that have not passed quality control checks yet. With agile, we want to break those deliverables down into smaller chunks and manage those tasks more actively, prioritizing what will drive the most value to customer next.

After the experiments were completed, and with pilots well underway, it was time to transition the first wave of agile squads onto our programming book of work. We deployed multiple squads and recognized that for every person in every role, there was something uniquely new. Scrum Masters had not supported squads delivering the data and analytics packages that our teams produce. Product Owners (SLPs) were learning what it meant to be a product owner. They also were learning about agile ceremonies, daily standups, and sprinting. We also deployed multiple POs to a squad, which may not be a typical practice in agile, but we looked for a meaningful book of work for a squad to be responsible for delivering. This meant that a squad might need to work on multiple product backlogs at the same time, hence the ability to prioritize the needs of customers across the portfolio of products assigned to that squad was a critical aspect of success.

As we move forward to future squads migrating into this new way of working, we know that we must foster incremental learning. The teams themselves must reflect and evolve how they work, but we must also learn and adapt our general framework for supporting agile ways of working in our projects. How should our central guidance change, even with squad autonomy to self-organize? We need to ensure that squads are sharing knowledge of what is working and what isn't. We support our product owners to connect to each other and our scrum masters to do the same. This feedback loop is essential for evolution and growth – being agile about our agile.

Overall, our migration strategy focuses on migrating people and not projects. We want to form the squads and bring the work to the squads. In the beginning, business continuity is critical, hence most squads onboard with their current scope of work. We want to enable flexibility without volatility on these squads. That means stable squads with a stable work remit. As we move toward our target of full adoption, however, we may find we need to adjust our approach to accommodate the last few squads. Flexibility without volatility.

BENEFITS AND CHALLENGES

So far in the GSK journey, we are seeing a number of benefits and potential opportunities. These include:

- Combined backlog above study/project
- Strategic prioritization of work
- More efficient resource allocation – prioritizing needs of the group
- Potential for data driven estimation of change requests
- Potential for earlier identification of risks to delivery timelines

However, these benefits have not arisen without our share of challenges:

- SLPs (PO) sometimes struggle with separating their role from that of the Scrum Master
- Squads must learn how to breaking down the backlog into tasks – smaller than the dataset/display
- Leveraging backlog management software (e.g., Jira) is perceived as more work for the squad members
- Agile ceremonies are perceived to be more meetings
- Conventional sprint duration (2 weeks) and frequency of standups (daily) are often challenged by the squads

These are challenges that the squads will work through and are not unexpected. Fundamentally, all of these challenges stem from the difficulty of changing mindsets engrained in us for decades of our conventional ways of working. For example, changing the mindset from one of 'getting things done' to 'delivering value to the customer'. Empowered teams will reflect (retrospectives) and adjust, though they may need a nudge from their scrum master or project management.

THREE TAKEAWAYS

Let's now explore three additional discussion points included in the presentation slides.

HOW DO WE DIFFER FROM SOFTWARE DEVELOPMENT?

Although we often use software development teams as a comparison point to adopting agile, we do differ from those teams in a few fundamental ways. We must keep these in mind when applying agile principles to the work we do to deliver clinical trial analyses to our customers.

- We know what our product should look like. We have the analysis plan, the mock shells, and the programming specifications. We have data standards and reporting standards. And it all has to be delivered at study end.
- We generally have fixed timelines. Software delivered in agile may never be done. Those teams ask the question – what features will we include in our next release? In our world, our last release must have all the features that have been prespecified by analysis plan and standards.
- We must be 100% accurate at the end – no bugs are acceptable.

BREAKING DOWN THE BACKLOG

When working with our teams to granularize the backlog, most people, quite naturally, opted to break down the pieces of SDTM and work on them first, then the pieces of ADaM, then the displays. However, in the spirit of the Agile Principle to 'maximize the work not done', let us consider an alternative approach centered on delivering value to the customer.

Until submission of the clinical trial data, we primarily drive value to customers that only care about the data analyses (TFLs). For example, most medical writers or clinicians are not going to find value in a demo of the new SDTM datasets that were generated. Therefore, consider a **display-driven** approach where the squad focuses on delivering a particular set of displays and generates the minimum aspects of SDTM and ADaM which are required for those displays.

When all displays in the backlog are done, those customers will have received a draft version of everything that they value and have had the opportunity to provide feedback on the deliverables. Although, the SDTM and ADaMs are would not yet be completed, they are done 'enough' to generate the displays – hence deliver to the customers. This approach would put the displays in the hands of customers much sooner and the team can subsequently turn attention to making SDTM and ADaM submission ready for their last customer (regulatory). It's quite possible that the team could prioritize and demo top-line results (typically 10-20 displays) to the wider study team within 1-2 sprints, rather than many months after programming has started.

ESTIMATION OF EFFORT

How much time does it take to create a display? Answer: it depends. Is it a table, a figure, or a listing? Is it a summary table or a statistical analysis? Is it a standard display or study specific? How experienced is the person developing the display?

A benefit of capturing the work (set of tasks) to be completed by the team into a more granular backlog is the ability to estimate how much time it will take to complete that task. Here, the work 'task' is a piece of vocabulary within agile ways of working. A task generally takes less than a day to complete and a set of tasks collectively deliver a feature of the product. If we can estimate the amount of time to complete a task, then we can estimate the time it takes to deliver a feature. That means we can estimate the time it takes to deliver all remaining features to complete the product, which in our world is well-specified.

Moving to a granular backlog, hence, enables us to estimate the aggregate amount of time is needed to complete our product. And, if we know what resources are available based on the capacity of the squad, we can estimate if there is enough capacity to complete the backlog of work within the set timelines. Turning that around, we can also identify (and manage) risks to achieving our reporting timelines. In addition, if any changes to the requirements occur, e.g. a request to modify or add displays, we can more easily assess the impact on timelines and resources needed to accommodate the request. Perhaps changes are needed to the SDTMs, the ADaMs, or the associated documentation, in addition to the displays themselves.

Although this is a tremendous benefit of estimating effort, it does have a tradeoff. Relative to conventional ways of working, our teams may see this as micromanaging or increased monitoring of their performance. If a task was estimated to take 2-4 hours, and it takes 6 hours, what would we do with that information? Will the person's manager set up a call to discuss the issue or ask for an explanation? I hope not, but the information is there and that can be uncomfortable for the programmer until, and unless, leaders only leverage the information in the aggregate to help the team as a whole.

CONCLUSION

Agile is not just about sprinting and daily standups. Agile is not just about project management and completion of tasks. Agile Principles, built upon the 4 core pillars of value in the Agile Manifesto, underpin many of the elements of agile ways of working that you might see in a training program. If we look at the work we do to deliver clinical trial

data and analyses to our customers through the lens of the agile principles, I believe we can do more to do our part to deliver life-saving medicines and vaccines to patients as quickly and confidently as possible.

REFERENCES

Patel, Sheetal and Michael Rimler (2024). "Agile? WoW! How Agile Ways of Working can Drive Value for Conventional Clinical Trial Reporting," 2024 PHUSE EU Connect, LI07.

URL: https://phuse.s3.eu-central-1.amazonaws.com/Archive/2024/Connect/EU/Strasbourg/PAP_LI07.pdf

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

CONTACT INFORMATION

Author Name: **Michael Rimler**

Company: GSK

Address: 1250 S. Collegeville Road, Collegeville, Pennsylvania, 19426-0989, United States

Email: michael.s.rimler@gsk.com

Website: <https://www.linkedin.com/in/michaelrimler/>

Brand and product names are trademarks of their respective companies.