

## Thermo Fisher Scientific: Conclusion – Did Scrum Work?

The Thermo Fisher Scientific engagement was my first opportunity to apply the ideas from my Agile Hardware research in a real hardware-development environment.

Prior to the engagement, I had spent roughly eighteen months researching Agile methods for hardware development and had documented my conclusions in my white paper, *Agile Processes for Hardware Development*. The Thermo Fisher project would become the first real test of those ideas.

The circumstances were challenging.

The product was a new mass spectrometer.

The team had never used Scrum.

More significantly, the team members had never worked together before.

We were asking a newly formed team to adopt a development process that was unfamiliar to everyone involved.

Fortunately, the initial training went well. Although some team members were understandably skeptical, they understood the process and were willing to give it a fair trial.

Release Planning also proceeded smoothly. Aside from the team's initial difficulty in writing Stories and Epics, the planning process worked largely as expected. By the end of the session, the team had defined its first Release cycle and established a target completion date.

Sprint Planning went equally well. We did need to account for the highly specialized nature of the team members' skills, but this proved to be a manageable adjustment rather than a significant obstacle.

Then came the first Sprint.

For me, it was an unusually tense experience.

This was new territory for everyone involved, including me. I had spent considerable time thinking about how Scrum should work in a hardware environment, but this was the first opportunity to see whether those ideas would survive contact with reality.

I was prepared for problems.

They never appeared.

The Sprint proceeded with remarkable smoothness.

By the end of the Sprint, the Burn-Up chart maintained by Scrum Master Ray showed a nearly perfect progression toward completion. In more than a decade of software-development work, I had rarely seen such a clean Sprint execution, and I certainly had not expected to encounter it during my first hardware transformation.

The Sprint Review was equally encouraging.

The team members were enthusiastic about demonstrating their accomplishments. The opportunity to showcase completed work was new to many of them, and the atmosphere was celebratory.

Subsequent Sprints followed a similar pattern.

As the team gained experience with Scrum, the ceremonies became more efficient and the process felt increasingly natural. There were occasional adjustments and refinements, but no major surprises.

Most importantly, the experience validated the predictions from my research.

Every major conclusion from the Agile Hardware work proved accurate.

Scrum worked well in a hardware environment—not because the team copied software-development practices unchanged, but because they adapted those practices to the realities of hardware engineering.

Several months later, I contacted Product Owner Dr. Michael Belford to learn how the team was doing. His response was particularly gratifying:

"First and foremost, running this project with Scrum is an ongoing success. The team has formed nicely and we have proper buy-in from each team member."

"The daily standup is a life saver. As is the board. I can quickly understand where we are and what needs to be done."

"Release planning has been beneficial because it really focused the team onto what our big goals are as opposed to smaller engineering goals."

"I feel that management here in San Jose is excited about Scrum."

For me, those comments captured the essence of the transformation.

The most important lesson from Thermo Fisher Scientific was not that hardware development differs from software development. We already knew that.

The important lesson was that Scrum can work remarkably well in a hardware environment once those differences are acknowledged and accommodated.

The goal is not to force hardware teams to behave like software teams.

The goal is to help hardware teams become more effective at what they already do.

That was the lesson from Thermo Fisher Scientific, and it remains my view of Agile hardware development today.

For those interested in the full case study, the complete Thermo Fisher Scientific story is available on my website at <https://kevinthompsonphd.com/storage/papers/LessonsLearned-AgileHardware-v6.pdf>.