

Thermo Fisher Scientific: Introduction

Many Agile practices that work well in software development break down when applied directly to hardware engineering.

This should not be surprising. Hardware development is not simply "software development with longer lead times." The two disciplines operate under fundamentally different constraints involving dependencies, specialization, prototyping, integration, manufacturing, and the cost of change.

Over the years, I have worked with organizations developing products such as mass spectrometers, telecommunications equipment, and aircraft systems. One of the recurring themes across these engagements has been the need to adapt Agile practices to fit the realities of hardware development rather than attempting to apply software-oriented practices without modification.

The benefits of doing so can be substantial. Organizations that successfully adapt Agile methods to hardware development often experience improvements in schedule reliability, earlier identification of critical risks, better cross-functional coordination, and reduced confusion about priorities and deliverables.

Between 2013 and 2015, I conducted research into the application of Agile methods to hardware development. At the time, very little information was available on the subject. Most Agile literature assumed a software-development environment, and many practitioners believed that Agile methods were either impractical or impossible to apply in hardware organizations.

My research led to a surprising conclusion.

Scrum turned out to be a viable process for hardware development, but only after several assumptions inherited from software development were reconsidered. Many concepts that seem obvious in software organizations either behave differently or become much less useful when applied to hardware engineering teams.

One of the most interesting opportunities to test these ideas came through an Agile transformation at Thermo Fisher Scientific.

Thermo Fisher develops sophisticated scientific instruments and laboratory equipment, including mass spectrometers. The engagement provided an opportunity to apply Scrum in a hardware-development environment and observe what worked, what failed, and what needed to be adapted.

The lessons learned during that effort shaped much of my subsequent work in Agile hardware development.

In this series of articles, I will describe the Thermo Fisher engagement and the insights that emerged from it. Many of the lessons directly contradicted conventional software-oriented Agile thinking, yet proved essential for success in a hardware environment.

In the next article, I will describe how the engagement began and why everyone involved understood that we were venturing into largely unexplored territory.

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>.