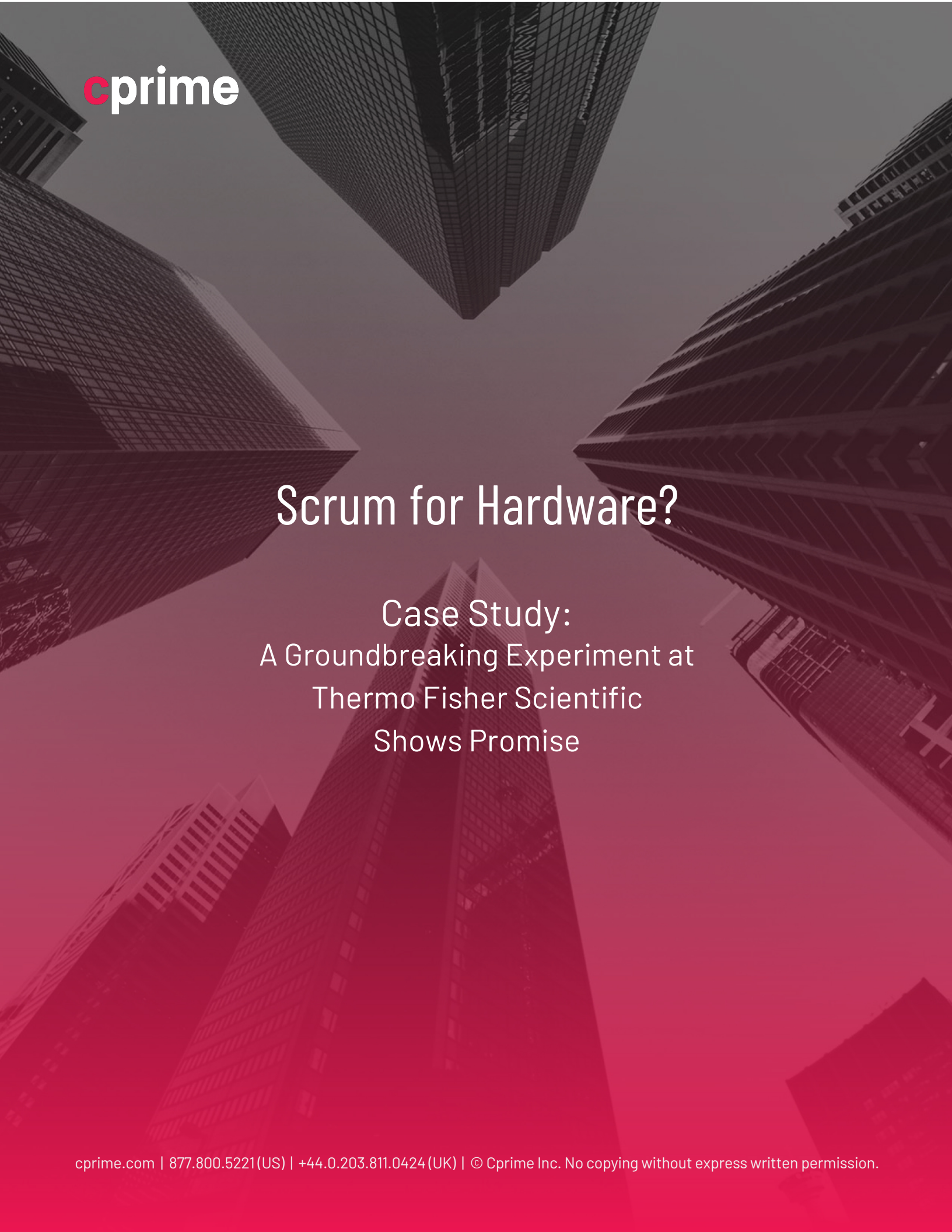

cprime

Scrum for Hardware?

Case Study:
A Groundbreaking Experiment at Thermo Fisher Scientific Shows Promise

Company Details

Industry: Manufacturer of analytical, diagnostic and laboratory equipment

Company Size: 50,000 Employees

Location: Headquarters in Waltham, MA, USA; 50 Countries

Revenue: $17.5 billion a year

As Scrum and Agile spread outward from software development to a wide variety of industries, hardware has been a notable exception. But a recent pilot project at Thermo Fisher Scientific, led by the Agile consulting company Cprime, suggests that Agile can be implemented in hardware-only projects with excellent results.

Overview

The world's largest maker of analytical, diagnostic and laboratory equipment, Thermo Fisher Scientific has revenues of $17.5 billion a year and employs 50,000 people spread across 50 countries. The project in question took place in the company's San Jose, California, office after employees there heard positive comments about some Agile experimentation taking place in the company's office in Bremen, Germany.

"We decided to try it here to see if we could get similar results, such as increased transparency of project status and improved collaboration between marketing and R&D and among different technical functions," says Senior Manager Marisa Richardson. "We also needed a project management process that was acceptable in a regulated industry."

With her background in Lean and Operational Excellence, Richardson said she felt Scrum offered a nice framework for defining and monitoring the work of a complex, multidisciplinary project. "I was also interested in a process that would bring more visual management to how projects were led."

Thermo Fisher was familiar with Scrum, having used it in software, but no one, to the knowledge of either Thermo Fisher or Cprime, had tried Scrum in a strictly hardware environment — particularly when the goal was not a prototype but a fully developed, saleable product. "There was no obvious history to point to of anyone who was already doing this successfully," says Cprime's Agile Practice Lead Kevin Thompson, PhD, PMP®, CSP®, who led the project.

And Cprime, Thompson is quick to point out, hadn't done it either. However, Thompson was an obvious person to lead the project, having written a paper, "Agile Processes for Hardware Development," in which he argued that a Scrum process should be effective for development of electronic and electromechanical devices.

How is Scrum Different in Hardware?

What makes the implementation of daily working Scrum so different in a hardware environment? You might better ask what isn't different, says Thompson. To start with, software development typically proceeds at a fairly rapid pace and is broken down into separate steps or iterations, while it might take three to six months to get to a working hardware component or feature. More specifically:

1. It's not easy to add functionality over time.

In traditional Scrum, features and functionality are added as you go along, but hardware is set from beginning. "You design the entire system to do certain things and then you build components to fit that plan," says Thompson. Because hardware is developed according to strict process models and has to meet specific compliance standards, it's much harder to make changes, particularly late in the process. When late changes do have to be made, the costs can run high.

2. You can't test in the middle.

In traditional Scrum, the team tests product iterations, then makes usability modifications accordingly, without altering the flow of execution. But with hardware, it's not possible to test before all the components are assembled. "Take a car, for example — you can't drive it when all

you've got are tires and a drivetrain," says Thompson. "You've got to put an awful lot of this stuff together before you can check capabilities, so that [checking] happens much later than in the software world."

3. Everyone is learning a new language.

It wasn't just the hardware developers who were learning a new language in being introduced to Scrum. Thompson realized he too needed to listen and learn. "They didn't know how to do Scrum, but they did know how to design equipment," Thompson says. So he started by listening in team meetings and learning how the current process worked. "A hardware team talks about pieces of equipment, they don't think in terms of functionality. So I said, "Let's divide up the world your way, and go with how you think about all this."

How Did They Do It?

In many ways, Cprime's work with Thermo Fisher Scientific followed the same process as any other type of Scrum training. The primary differences were the environment, the deliverables, and the fact that the team members, all hardware engineers, had zero experience with Scrum.

Choosing the team.

Given the pioneering nature of the effort, it was important that everyone involved understood that the project would involve significant challenges, says Thompson. "While Cprime has enormous experience working with a variety of clients, Agile hardware development is new." Even more critical, managers chose members of the nine-person team carefully, looking for innovators who had good tolerance for risk and mistakes.

Making the plan.

To ensure the greatest chance of success, Thompson advised Thermo Fisher to choose for the pilot a medium-sized project that was important to the company, but the success of which was "not a life-or-death matter." They chose a new product known as "Trailblazer" that met these criteria and launched in late July 2015.

Setting a schedule.

- ***Week 1****: Scrum training and writing high-level requirements (epics)*
- ***Week 2****: Drafting the release plan, including setting estimates for all epics, writing user and technical stories in preparation for Sprint 1*
- ***Week 3*** *(first week of Sprint 1): Sprint planning, developing and testing deliverables*
- ***Week 4*** *(second week of Sprint 1): Developing and testing deliverables.*

Defining a release cycle.

Thermo Fisher defined the product life cycle in three phases: engineering prototype, manufacturing prototype, and final design. After consideration, Thompson and the team decided it made most sense to map release cycles exactly to the three main design iterations. Thus the first release cycle was to design a working engineering prototype. The team estimated the full period necessary for the project at 15 sprints, or 30 weeks, or about 7 months.

Lessons Learned: What Worked and What Didn't?

At first, as the team began Scrum training, "there was a fair amount of bafflement and amusement," Thompson says, at this completely different way of working from their previous Waterfall-style project management. But quite quickly things began to come together. Here are the most important lessons Thompson learned about how a typical Scrum process works in a hardware environment.

Lesson #1: The product owner is also a team member.

In software development, product owners tend not to be software developers but to be product managers focused on market and business needs. But in hardware, the people who drive the definition of hardware products will usually be those participating in the development process.

Lesson #2: Writing hardware requirements is a different process.

To break down a project into specifications and deliverables, it's necessary to cover all aspects of the product as currently understood. In software, a typical process would be to define the major user-facing areas of functionality that were important for the first (or next) product release, and write specifications for them. But in hardware the focus is on what the product does and the components necessary to achieve that successfully. This meant that few of the deliverables reflected user experience, and most deliverables were behind the scenes. In Scrum language, there were lots of technical stories and relatively few user stories.

Lesson #3: Release planning is essential.

In typical Scrum, the fundamental development cycle is the two- to three-week sprint, and the much longer time horizon of a release cycle may or may not be part of the process. But because the cost of change is so much higher in hardware development, and the time to usability is much longer, it became clear that release planning was required. Going forward, "we sensed that we need to do more release planning and project risk assessment at the beginning, to be able to give a more predictable overall schedule for the organization to plan by," says Richardson.

Lesson #4: Stories are tested by implementers.

While in software, quality assurance specialists are generally tasked with software testing, hardware deliverables vary so widely (from, as in this case, the design of a circuit board to an actual device or piece of machinery) that developers are in most cases the only ones who understand them well enough to test them.

Lesson #5: Time-based estimation works best.

Traditional Scrum teams collaborate to estimate stories relative to each other, in story points. However, hardware development teams comprise a mix of such highly specialized skills that establishing story point norms as a group proved impossible. The team found time-based estimation much more successful and intuitive.

What Made This Experiment Successful?

Taking the time to plan together allowed the team to communicate and collaborate better than ever before, everyone involved in the project agrees. "The process of planning went more smoothly, with fewer bumps than they were used to," says Thompson.

Now, adds Richardson,

> *"There's an increased level of expertise in the overall product by the whole team because they work closely developing and reviewing sprint deliverables, as opposed to working in functional silos that may not be in sync."*

Team member Michael W. Belford, a research and engineering scientist at Thermo Fisher, agrees, "I make technical decisions with more input from the team than in any other project that I have worked on," says Bellford. "The daily stand-up is a lifesaver, as is the [visual task] board. I can quickly understand where we are and what needs to be done."

Outcome measures were another area where results were almost immediately apparent. "The first sign this was a success was that the burn-down chart was eerily on target; the tracking records showed we were exactly on plan," says Thompson. Both team members and managers appreciated the resulting improvements in predictability.

Scrum has also increased team members' ability to see the larger picture. "Release planning has been beneficial because it really focused the team on what our big goals are, as opposed to smaller engineering goals," Belford says.

Even what had seemed to be a problem has benefits, says Belford. "Grooming is a hassle because it takes so much time and requires us to stop doing and think, but the drawbacks I mention are also the clear benefits."

Thompson emphasizes that one of the most important aspects of the process is that it was clearly defined and purely Scrum, not something Agile but less defined. "I'd say that if we broke ground here, it was in creating a standard way to use Scrum for hardware development. All the i's are dotted, and the t's are crossed. It isn't a one-off custom solution, but a fully reusable standard that I'm now using with other hardware clients."

What Happens Next?

Currently the pilot project is still going strong at Thermo Fisher, with the nine-person team continuing to "push the envelope" in exploring the application of Agile to hardware.

Considering it an experiment in progress, Thermo Fisher Scientific isn't ready to draw any specific conclusions yet. Still, the company hopes to expand the process to other teams and projects. "We would like to experiment with a bigger project, a distributed team, and continue to evolve with these variables," says Richardson. "There are a lot of parallels with Lean, and we've seen Lean adopted outside of the automotive and manufacturing sectors to many different industries, such as health care and the military. It definitely makes sense that similar trends could happen with Scrum."

Meanwhile, as word spreads, Thompson says he's seeing a great deal of curiosity among those in hardware, and several clients have approached him about starting projects of their own. In particular, companies that use Scrum for software are interested in expanding it to hardware with the expectation that it would improve coordination between the two, says Thompson.

"Any time you're building a product and your software and hardware need to get along, then the people who build them need to get along, and now you're going down the path towards Scrum."