

## Thermo Fisher Scientific: Lesson #2 – Release Planning Is Essential

One of the conclusions from my original research into Agile hardware development was simple:

**Failing to plan at the Release level means planning to fail.**

The Thermo Fisher Scientific engagement reinforced that lesson almost immediately.

The need for Release Planning in software development varies from organization to organization. Some teams place a great deal of emphasis on Release Planning, while others rely primarily on Sprint Planning and maintain only a rough roadmap beyond the current Sprint.

The primary drivers for Release Planning in software are coordinating dependencies, setting expectations, and making feature tradeoff decisions over time. A software release plan is often built around a cadence model, such as a release every three months or every six Sprints.

Hardware development operates under a different set of constraints.

The cost of change in software is generally manageable. While software defects and design mistakes can certainly be expensive, modifying software is usually far easier than modifying hardware.

In hardware development, design changes often require new prototypes, new circuit boards, additional testing, supplier coordination, manufacturing changes, and repeated verification activities. The further a project progresses, the more expensive these changes become.

As a result, hardware organizations need visibility into future work much earlier than software teams typically do.

Key architectural decisions, system constraints, and minimum viable feature sets often need to be understood months in advance. Many decisions must be made early and adhered to consistently, because reversing them later can dramatically increase both schedule and cost.

This reality also places a greater emphasis on risk management.

While software teams certainly manage risk, hardware teams must identify and address major technical risks as early as possible. Failure to do so can result in costly redesign efforts late in the development cycle.

The management team at Thermo Fisher understood this instinctively. They expected longer-term planning than a two-week Sprint horizon could provide, and they were absolutely right.

Together, we developed a Release structure consisting of three major development phases:

### **Functional Prototype**

The goal of the Functional Prototype phase was to demonstrate that the concept worked.

At the end of this phase, the team would have a working device that proved the feasibility of the design. However, the prototype was not intended to be suitable for manufacturing.

### **Manufacturing Prototype**

The Manufacturing Prototype phase focused on moving from a proof of concept toward a design that could actually be manufactured.

This represented the team's first serious attempt to address manufacturing realities, supply-chain concerns, production constraints, and other practical issues associated with delivering a commercial product.

### **Manufacturable Design**

The final phase produced the design that would ultimately be transferred to the manufacturing organization.

At this point, the product design was sufficiently mature to support production activities.

One aspect of this structure is particularly important.

Unlike many software Release Plans, these Releases were not based on a fixed cadence.

The Releases were defined by what they delivered, not by how much time had elapsed.

The Functional Prototype, Manufacturing Prototype, and Manufacturable Design phases all required different amounts of time. Attempting to force them into identical time boxes would have created an artificial constraint that did not reflect the realities of the work.

This proved to be a much more natural model for hardware development.

The lesson was clear.

### **Lesson #2: Release Planning is not optional in hardware development.**

The higher cost of change, longer development cycles, manufacturing considerations, and increased importance of risk management all require a planning horizon that extends well beyond a single Sprint.

Just as importantly, Release Planning in hardware often looks very different from Release Planning in software. Hardware Releases are frequently organized around major engineering milestones and deliverables rather than fixed calendar intervals.

Understanding that distinction is one of the keys to successfully applying Agile methods in a hardware environment.

In my next post, I will address how scope decomposition differs between hardware and software development.

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>.