Thermo Fisher Scientific: Lesson #11 – Why Sprint Planning Is Different in Hardware Development

One of the predictions from my Agile Hardware research was that Sprint Planning would need to account for the specialized nature of hardware-development teams.

The Thermo Fisher Scientific engagement validated that prediction.

We began Sprint Planning in much the same way that many software teams do. The Product Backlog was ranked according to Product Owner priorities and dependencies, and the team estimated Stories during Sprint Planning.

After that, however, the process diverged from the typical software model.

Many software teams plan Sprints primarily at the team level. Stories are estimated, a forecast of team Velocity is established, and Stories are selected until the Sprint reaches its capacity limit.

This approach works because many software teams consist largely of generalizing specialists. Although team members have different areas of expertise, work can often be redistributed when necessary to balance workloads.

The Thermo Fisher team operated differently.

As discussed in previous articles, the team consisted of highly specialized experts in areas such as optics, electronics, firmware, physics, and mechanical engineering. Most Stories required expertise possessed by only one or two team members.

As a result, assigning work to the Sprint was not simply a matter of staying within the team's overall Velocity.

We also had to consider the capacity of individual team members.

A Sprint might contain fewer Person-Days of work than the team's forecasted Velocity and still be infeasible if too much of that work required the same specialist.

For example, if several Stories required optics expertise, it was possible to overload the optics engineer while leaving other team members with excess capacity.

This meant that Sprint Planning required a second level of analysis.

In addition to monitoring overall team capacity, we had to forecast the availability of individual team members and evaluate how work would be distributed across the team.

As Stories were considered for inclusion in the Sprint, we sometimes found that the workload assigned to a particular specialist exceeded that person's available capacity. When this occurred, Stories had to be swapped in or out of the Sprint Backlog until a workable balance was achieved.

The process was not particularly difficult, but it was different from the simpler team-level planning model often described in Scrum literature.

The underlying reason was straightforward.

Unlike many software teams, hardware teams often consist of specialists whose skills overlap only minimally. Specialized expertise creates planning constraints that do not disappear simply because the team is using Scrum.

This experience reinforced another conclusion from my Agile Hardware research.

Lesson #11: Hardware Sprint Planning must account for both team capacity and individual capacity. A Sprint is feasible only if neither the team nor any individual specialist is overloaded.

The lesson is not that Scrum planning fails in hardware development. Rather, it is that hardware teams often require an additional layer of capacity planning to account for highly specialized skill sets.

In the next article, I will discuss another consequence of specialization: why swarming and parallelization behave differently in hardware development than they do in 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.