

An Interactive Case Study: Rising Above Failure – Part 2

What happened next...

The consensus between the coaches and the temporarily reassigned executive from corporate was that the 2 pilot teams performed well in their first 3-4 sprints. Teams delivered working software that met their Definition of Done. They worked through challenges of branching and merging code. The manual tests they wrote and executed gave them confidence in the code's quality. Managers and participants in the User Advisory Group (UAG) were getting comfortable with Sprint Reviews, the concept of the prioritized product backlog, and incremental delivery.

The decision was made to add 4 more Scrum teams, bringing the total up to 6 Scrum teams who would be working together. But, there were not enough Scrum Masters, Testers, or Coaches for the new teams. So the subsidiary's executive team agreed to provide more funding for these roles. Since, in some cases, 2 teams could work from 1 backlog, it would be acceptable for a Product Owner to work with more than 1 team. With a solid resource plan, the same enablement approach for the 2 pilot teams was used to launch the new 4 Scrum teams.

However, once scaling began, several conflicts started to arise, or at least, become more apparent, causing significant churn.

From a resource perspective, there was more attrition in the Product Management group. 2 members left for medical purposes, leaving only 2 Product Owners (PO). 1 PO would be full time for 1 team (he was brand new to the company), and the other PO only had time to be a consultant for remaining teams. The good news is that a new Product Manager was hired, and there would be investment over the next year to rebuild the Product Management group.

The Scrum Masters noticed certain senior-level team members on all 6 teams started disappearing during the day. They were in standup, but not in the team room. The tasks they pulled in were not getting done. Blockers started to form, and they were nowhere to be found.

Interestingly, nearly all the help desk tickets for fixes and support for live products were being ignored...for about 2 months. The small issues that the team members would typically fix weren't getting resolved. These blossomed into heated escalations, not only within subsidiary's executive team, but also the customer's executive team as well. To satisfy the customer's executives, the subsidiary's executive team required the teams to create a comprehensive detailed 90-day plan, with estimates for all work included.

As the escalations trickled from executives to the Managers, the managers began to not show up to the Enablement Team twice weekly standups. There were fires to put out, and the Managers needed ensure that the issues were being resolved quickly and to their satisfaction. Managers were actively poaching their direct reports out of the Scrum teams to fix the issues. One manager stated, "They are my people. I will do with them what I want."

Furthermore, Managers began to not attend sprint reviews, and were getting frustrated that the Scrum Masters would not let them attend daily standups. One Manager wanted daily standups to be staggered so he could visit them all. Many Managers felt out of touch with the teams and what they were specifically working on.

Ultimately, 5 of the Managers required a weekly status report, each tailored to their own interests. So, a Program Manager from the PMO created 5 different status reports with similar information that went to the respective manager every week. Of course, this required the Scrum Masters to provide the info weekly to the PMO so that they could create the reports.

Technical challenges began to surface. In the pilot, with 2 teams integrating within a sprint, there was enough collaboration, coordination, and communication to quickly and easily resolve issues. Now with 6 teams branching and then trying to merge code, this was another story. Conflicts arose during merges, leading to “code freezes” within sprints, several days before the end of each sprint. It took nearly the entire set of developers on all the teams to debug and get the code to work as an integrated product.

Additionally, the 6 Scrum teams had technical dependencies on 4 more other teams, who were working in a waterfall fashion. So, even if the 6 Scrum teams completed their work, there would still be additional work required from the other 4 teams before the product could be rolled out. In other words, it would take outputs from all 10 teams to have a working product.

For Quality, testing within the initial 2 pilot teams was manual and somewhat manageable for the scope of work they completed. There was a conscious decision from Managers to put off automated testing until later sprints. Now with work completed by 6 teams every sprint, the impact of testing went beyond teams; there needed to be a view point of quality as an integrated product. Furthermore, since more scope was getting completed every sprint, the amount of time to do manual regression testing increased exponentially. It was not unusual for testing to trickle into the next sprint.

Round 2 Exercise:

With the folks at your table, answer the following questions (keep at “headline” level):

1. Now what are the top 1 or 2 potential failure points? Why?
 - a. Are these the same as your answers from Part 1?
2. What would you do to lessen the churn at this point?