Getting Agile

A Case History

Joseph Hillery
Scott Pigman
Who We Are

Scott Pigman - I started my professional career as a mechanical engineer performing finite element analyses and then switched over into mechanical design using Unigraphics (NX). Along the way I became interested in software development and managed to eventually make the transition into my current career at Northrop Grumman. My primary focus is Teamcenter architecture and development.

Joe Hillery – I have been working in software development since the 70s, and with (Westinghouse) Northrop Grumman since the early 80s. Over the years, I’ve worked with various PDM systems: Sherpa, Teamcenter Enterprise, and now, Teamcenter.
What are We Doing?

This presentation is not about *what* we’re doing, but…

We have an existing Teamcenter 10.1.4 being used for CAD Model Management and Manufacturing Support. Our initial goal is to:

- Upgrade to Teamcenter 11.2
- Add Aerospace and Defense (A&D) module
- Interface with a legacy Configuration Management System (load data and keep synchronized)
Why use Agile Scrum?

Three key factors converged in the beginning of 2016:

• IT Management began to push the use of Agile.
• Agile Tools Became Available
• We were stuck.
The Agile Scrum Methodology

Global Product Data Interoperability Summit | 2016

- Features to stories
- Stories estimated
- Release plan
- Release definition of done

- Stories to tasks with hours
- Story definition of done

**Release 0 (Project Startup)**
- ~4-8 weeks
- Identify team
- Product vision
- Training
- Initial schedule/cadence
- Initial architecture
- Features
- Product roadmap
- Infrastructure setup

**Release Planning**
- 3 month cycle
- Prioritized actions for improvement

**Sprint Planning**
- 2 week cycle

**Sprint Scrum**
- 15 minutes
- Tasks completed
- Regular collaboration

**Sprint Retrospective**

**Sprint Review**
- Demonstrate completed stories
- Stories accepted
- Review status
- Potentially shippable

**Release Closeout**
- Features complete and ready for release

**Project Closeout**
The Agile Scrum Methodology

- Business / Management
- Solution Architect
- Dev Team
- Scrum Master
- Stake Holders
- Product Owner
- ADP
- AGILE CoP DIRECTORY
- SERVICES
- ABOUT
Collective experience of team on Agile Scrum Methodology

Certified Scrum Masters
What resources are available within the company?

- Agile Center of Excellence
- Agile Community of Practice
  - Web site with links to online resources
  - Bi-weekly lunchtime online meetings
- Agile coaching
- Training
Starting from Scratch

Essential Scrum: A Practical Guide to the Most Popular Agile Process (Addison-Wesley Signature Series (Cohn)) by Kenneth S. Rubin (Author)
Our approach to getting started

- Agile 101 Training
- YouTube videos
- Find a Coach
- Pick Teams
- Decide Roles

- Recruit potential Scrum Masters
- Recruit Product Owners

- Bang out product backlogs, while learning TFS
- Figure out next step when we get there
Challenges

Team members in three time zones
Organizing

What we want to accomplish in Release 1

• Support all existing functionality for Manufacturing and CAD integration in new system.
• Load Parts and Documents from legacy system and keep them in sync. The primary tool used for this function is eQube/MI.

Our initial plan was to have three Scrum teams:
• Manufacturing
• Design Data (CAD)
• ERP integration
Who is the Product Owner?

The challenge was to find a single person for each Scrum Team who was willing to define the Product Backlog, knowledgeable about the product, and could speak for everyone in our multi-campus coast-to-coast environment.
Who is the Scrum Master?

We looked for people who could

- Help the other team members with the Scrum process
- Advocate for the team and shield from external interference.
- Work to eliminate impediments to team’s progress
- Don’t Annoy! We’re doing a lot of new things in new ways. A calming presence would help.
The Development Teams

We did not have a pool of developers experienced with the technologies we were using (Teamcenter, eQube/MI, git). We tried to balance the experience levels across the teams.

The teams each had representatives for development, test and validation, and system administration.

We had some people as members of multiple teams. That became a problem, so we adapted.
A Sprint is a time-boxed period of one month or less during which the Development Team produces a useable, potentially releasable, product increment.
How long is the Sprint?

We started with a 4-week Sprint. That seemed too long, and out-of-sync with what we were hearing was the norm around the company.

Then we “adapted” to a two-week Sprint, which we thought was more the norm. We found that was too short, and we weren’t finishing our Product Backlog Items. Plus, it was difficult to manage meetings with three Scrum Teams.
# How long is the Sprint?

Finally, we moved to a three-week Sprint. We could finish Stories, and scheduling meetings wasn’t such a nightmare.
Although it is not an Enterprise standard, our sector is centering on Microsoft Team Foundation Server (TFS) for managing Scrum. It maintains:

- Separate Team Areas
- Product Backlogs
- Sprint
  - Backlogs
  - Team membership and capacity
  - Tasks and the Task Board
  - Test Plans and Test Results for the Tasks
  - Burndown chart.
- Reports and Queries (e.g. Velocity Chart).
Early Mistakes

• We skipped Release Planning!
• We should have had a Sprint 0 for each development team where the environmental issues were ironed out.
Incorporating other Agile tools (trying anyway)

We wanted to do more than just break the Product Backlog into Sprints. So, we wanted to include

- **Test driven development**
  - Teamcenter private classes makes this a bit difficult.
- **Pair programming**
  - In our case, a learning tool for inexperienced developers.
- **Test automation**
  - So far only manual, but TFS tracked.
- **Build automation**
  - Semiautomatic creation of deploy packages.
  - Continuous integration is our goal.
After about five parallel Sprints, we decided that the Manufacturing and the Design Integration Development Teams had a lot of commonality and it made sense to merge them.
A Mid-Release Adjustment

At that time we also rearranged roles, moving two former Product Owners to Stakeholders.

And, the Solution Architect, who really was the one who knows the application became, also, a Product owner.
Challenges

How the Developers have taken to Scrum

• Lots of additional meetings, especially if someone is on more than one team.
• Developers reporting to different management chains might have different levels of support.
• Many new things all at once: TFS, Teamcenter, git, eQube. Can be overwhelming.
• Scrum Lingo
Challenges

Getting the Test and Validation Team members involved early. We want to avoid two weeks of inactivity and boredom, followed by a week of furious activity.
Challenges

“Deliverable” Product at the end of the Sprint?

We have to really bend the definition of “Deliverable” for our two projects, especially the legacy data load.
Velocity and Burndown

- Keeping remaining hours up-to-date
- How do we handle Stories we don't finish:
  - Split up into parts we did and didn't finish?
  - Just carry forward and average velocity?
- Try to make stories that better match the sprint capacity
Challenges

Breaking free from traditional patterns of behavior:

• Someone has to be the boss vs team responsibility
• Reluctance to create tasks and assign to others
• Unit Manager Identity Crisis
• Upper Management still wants to track with traditional metrics (MS Project).
We are now eight months into this process. How has it been?
Where are we on the path to DevOps?

- Plan
- Code
- Build
- Test
- Release
- Deploy
- Operate
- Monitor