Agilent's agile transformation journey

How agile helped Agilent reduce errors, improve quality, and increase product capacity

BY DAVID VERMETTE

In 2015, Agilent's Software and Informatics Division was in trouble. In the midst of a major new product release, the team was going to miss its release date. It wasn't the first time; the division only met about 20 percent of its releases on time.

The missed release dates created tremendous pressure on the software teams. "When teams are under pressure they make bad decisions," said John Sadler, VP and GM of Agilent's Software and Informatics Division. "They sacrifice quality and end up paying for it on the back end. Overall, the team was relatively demoralized and had a hard time delivering anything on time."

Part of the challenge was that Agilent's Software and Informatics Division included 150 employees on two continents who collaborated with contractors on a third. It was a sprawling division of engineering, marketing, quality assurance, tech support, and sales professionals. The company needed to establish new team practices to help it to deliver value faster, with greater quality and predictability, and better respond to change. In short, they needed an agile transformation.

Embarking on an agile transformation

In 2015, Agilent mapped out an agile transformation with the following goals:

- Focus on predictability
- Develop a reliable release cadence
- Create a reproducible delivery cycle
- Drive continuous improvement

The agile approach puts quality first, a reproducible cadence second, and scope third. "When teams don't follow these priorities, they often put scope first, the timeline gets imposed in the form of a deadline, and quality goes begging," Sadler said.

Agilent wanted to establish agile processes that baked

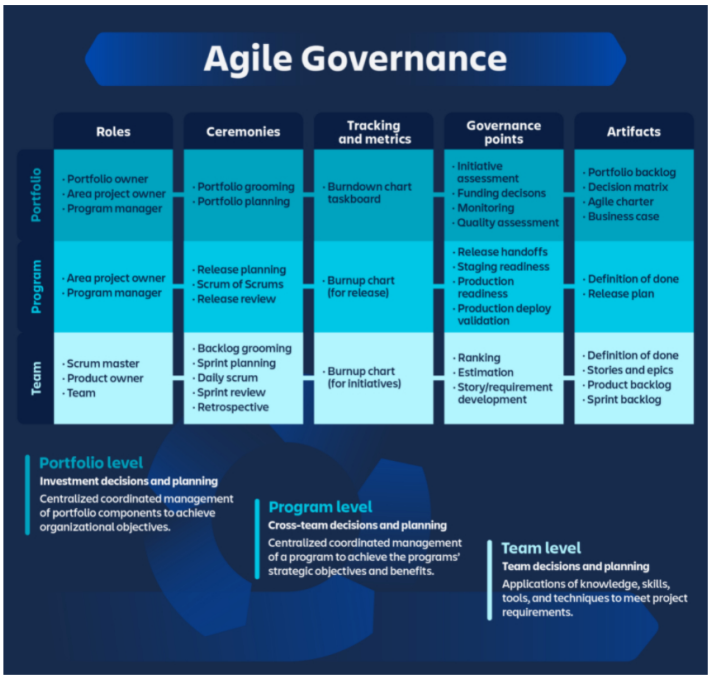
priorities into their software development. Establishing continuous improvement as a primary value entailed a cultural shift that created context for the transformation. Putting quality first meant improving satisfaction for both external and internal customers. Agilent was willing to sacrifice scope in order to meet customer expectations in the field.

The second priority was for the division to create a predictable release cycle that could be fine-tuned over time. A shorter, more reliable development cycle meant the team would avoid problems and mitigate risks early in order to give adequate response time.

Agile transformation: step-by-step

Beginning in August 2015, Agilent partnered with cPrime, a consulting company experienced in helping organizations through agile transformations, and later with TCGen, a leading product development consultancy.

Implementing the agile product management tool Jira Software was the first step. Jira Software now provides one system of record for all Agilent's development work, across distributed global teams. It creates transparency with a single version of the truth that reveals agenda features, when they are expected, and what was accomplished across teams.



Adapted with permission from cPrime, Inc.

Agile training was the next priority, with goals to teach agile principles and concepts and establish a shared terminology. cPrime outlined its recipes for Agile Governance in the Enterprise (RAGE) model that outlines key ceremonies including release planning meetings, team Scrum of Scrums meetings, product owner Scrum of Scrums meetings, release backlog grooming meetings,

release review, and release retrospective meetings.

Agilent also established area product owner roles and program manager roles, and measured progress with burnup charts. Agilent also adopted lightweight decision-making techniques, executed ceremonies, used agile scrum artifacts, and adopted other agile practices.

Agile in action

In November 2015, Agilent's Software and Informatics Division held Sprint Zero, a two week agile planning session with fourteen trained teams. A product release plan was developed spanning eight months for the OpenLAB chromatography data system.

Sprint Zero activities included three categories:

- Educate teams on the business and technical goals, drivers, and high-level requirements for OpenLAB through presentations and posters.
- Use the newly learned techniques to develop requirements and estimate stories, epics, and defects.
- Lay out stories, epics, and defects on the Release Planning board, and mark all cross-team dependencies.

Agilent soon found that their agile improvements extended beyond the pilot project. One of the first indicators of success was internal. "Because we had to partner with other Agilent divisions, doing what we said we were going to do was immensely important," Sadler said.

Agilent also saw external improvements. Customer feedback became instrumental in the Agilent teams' increased responsiveness to changes in the market. By including customers early in the development process, Agilent increased software quality while reducing market and integration risk.

One of Agilent's insights was to include in the definition of done that upgrade stories be included in every agile epic. Babita Jain, director of software quality and Stefan Weiss, manager of software integration at Agilent, led the transformation implementation and helped the teams understand the total scope of the project. "We did not consider the epic done unless it includes the upgrades," Jain said.

The agile transformation not only improved quality, but also reliability. In 2016, the OpenLAB chromatography data system was delivered on time. Since the agile transformation, Agilent has delivered software in a steady cadence and field-reported defects have declined.

Measuring success

Agilent defines and measures the success of its agile initiative with “low field failure rates and high customer loyalty,” said Sadler. Customer retention is also key. In the mature, competitive market in the life sciences industry, customer attrition is a danger. The ability to sell old customers a new system is essential.

The following are four critical metrics enabled by Agilent’s Jira-based capacity model:

Burndown/burnup charts

Agilent previously measured work in engineering hours, person days, or story points. Now everyone uses burnup charts to track completed work and the total amount of work. Burn down charts show how much work remains.

Percent of capacity delivered to market (payload)

The ability to model and track capacity plays an essential role in how Agilent measures success. Jira Software enabled Agilent to build a detailed capacity model. “This model allowed us during the planning phase to tell marketing how many story points they had to work with for a release. Marketing gets to rank the backlog to fit that capacity,” Sadler said.

Count and median time to deliver fixes

The capacity model provided a more granular view that allowed Agilent to see the capacity spent fixing critical or serious field-reported defects versus defects that the quality team discovered and reported.

Percent of time spent fixing defects discovered by manual testing

The capacity model helps Agilent measure time spent making features that customers value versus time spent maintaining or sustaining existing products.

In just over three years the division more than doubled the percentage of capacity focused on value-added work, increasing it from 30 percent to 65 percent. As software quality improved, driven by the new agile approach, there were also fewer defects to repair later.

A year after starting their agile transformation process, Agilent team members describe various "before" and "after" scenarios:

Before: Performance was measured in engineering hours, person days, or story points. **After:** Everyone agrees on how to measure velocity and burndown.

Before: Teams with different levels of agile training had different definitions for epics,

stories, and subtasks.
After: Everyone in the organization speaks the same language.

Before: Scrum masters wrote code, led team meetings, and weighed in on feature priorities.
After: Scrum masters are task masters who report up to product managers.

Before: Features with bugs could get approved, carrying defects into subsequent development phases.
After: A universal definition of “done” ensures that bugs are eradicated before the next sprint.

Before: Problems often occurred at the final hour, causing panic and substantial delays.
After: A shared set of tools used across all teams prevents surprises and helps maintain focus.

Before: There was no instrument for measuring a team's capacity to do work.
After: There is agreement on how to predict and measure capacity on a daily basis.

The agile future at Agilent

Like any culture that values continuous improvement, Agilent’s agile transformation is by no means complete. The next steps include continuing to shorten cycle time and hone the ability to gain early insight from market feedback, some key elements of DevOps metrics.

Agilent has already cut cycle time dramatically, breaking yearly releases into two six month cycles – even though the company commercializes releases annually. The company plans to cut cycle time in half again, moving to a quarterly cadence.

Getting customers involved early and often in the development process is also a subject of ongoing improvement. Sadler reports that his group finds that there is less arbitration among competing interests in discussions about product scope when there’s clear evidence from market feedback. Maintaining quality and usability for customers in the field is an ongoing priority, and ongoing engagement with users helps maintain the company priorities: quality first, then release cadence and scope.

Lessons learned

Sadler credits success to his team, including the leaders

Jain and Weiss, who embraced agile development as a way to drive rapid and continuous improvement. The right tools, like Jira Software and Confluence, allowed the team to consolidate work in one place and measure progress.

Agilent found that an agile transformation requires a sponsor who oversees research and development, inbound marketing, and quality. “If you can’t transform all three, you will not succeed,” Sadler said. “You can’t get an agile transformation done through R&D alone.” What’s most important is not the work that’s done within the functions, but the way they work together.

Agilent also found it’s essential to suspend the assumption that a perfect understanding of customer needs resides inside the building. An agile approach is to release products according to a reliable cadence and then directly receive customer feedback. This approach requires the attitude that release dates are sacred, and when the bell rings, the team ships the product as is.

Finally, advises Sadler, the agile framework makes problems visible. For example, agile exposes gaps in capacity. It reveals the non-value-added portions of the process that cause delays and reveal quality miscues. Shifting from a waterfall approach to an agile approach not only requires changes in how teams work, it’s also a cultural shift. Agile drives a culture of continuous improvement that is downright contagious.

DAVID VERMETTE

David Vermette is a freelance writer, researcher, and editor originally from Massachusetts. He often writes about the management of product development, including Agile methodologies, product strategy, and product portfolio management. A conservatory trained musician and jazz bass player, David is also the author of a history book called A Distinct Alien Race (Montreal: Baraka Books, 2018).

Scaling agile in large organizations

How can you realize the benefits of agile at the team level across all levels of your organization? Learn more about scaling agile.

ARTICLE

An inside look into release-ready teams

Reliable, consistent releases are a crucial part of a software team's success (and survival!). Learn how to build a release-ready team.