ptc®

DIGITAL TRANSFORMS PHYSICAL

TRANSITIONING TO AGILE DEVELOPMENT:

HOW TO DEVELOP HARDWARE LIKE SOFTWARE

Steven Eppinger
Professor of Management Science and Innovation
MIT Sloan School of Management

Jon Hirschtick
Chief Evangelist
PTC

WHITE PAPER

May 2023

Contents

Hardware teams are changing the process of how they develop their new products. Increasingly they are adopting Agile Process, the same type of process that has become the standard method to develop software. It is happening for several good and timely reasons and brings many benefits to product development. Agile is being adopted by some teams fully, by some teams partially, and by some teams who do not even realize they are using Agile processes.

It is important today for every hardware product development leader to understand what Agile Process is, its potential benefits, why now is the time for agile, and how it can be adapted to hardware – despite some of the obvious differences between hardware and software development. Many in the manufacturing space believe, due to their tangible nature, that physical products cannot be created and refined at nearly the same pace as software.

That said, many of these same organizations hold daily team meetings, use cloud-based tools to coordinate their work, refine product designs in real-time remote collaboration, and benefit from frequent customer feedback on their ongoing product development projects. Each of these workflow actions, while not spelled out as Agile development, is indicative of the larger Agile mindset. Although some in physical product creation still believe Agile is a tool for software, the reality is that many hardware teams are already using certain Agile principles, even if they are unaware of the terminology. Others are fully embracing the Agile mindset and process. Potential characteristics and benefits of Agile development, include but are not limited to:

- **Ease of distributed team collaboration**
- **Reduced uncertainty and risk**
- **Incremental value delivery**
- **Continuous, iterative product improvement**
- **Regular stakeholder feedback**
- **Process status checks**
- **Clearly defined team leadership roles**
- **Responding to a rapid rate of change**

This article will primarily focus on the first three listed benefits of Agile product development – collaboration, risk management, and value delivery. We illustrate how Agile methods have impacted manufacturing organizations that have embraced this methodology in their hardware creation processes. We chose to highlight these benefits due to their relationship to cloud technology, particularly cloud-native software as a service (SaaS) solutions. While none of the three highlighted benefits is directly dependent on SaaS solutions, today's cloud-based tools have dramatically reduced the friction of collaboration – and Agile is built upon collaborative team efforts.

Agile methods are applicable for nearly every type of engineering organization, not just software development teams. The principles of Agile development are so critical today, that hardware creators who are not starting to embrace Agile are at risk of being left behind in terms of competitive speed and value creation.

Why Agile Now?

Before exploring the highlighted benefits of Agile product development, it is important to first consider four key factors that are driving Agile process adoption today in hardware product development:

SOFTWARE: Many hardware developers today are also developing software. This software is developed using Agile processes. So even if the organization does not use agile for hardware, it is increasingly familiar with Agile and its benefits.

WORKFORCE: The young generation within today's workforce thinks of computing, communication, and collaboration in very different ways. Gone is the "paper-based" thinking of memos and emails – replaced by the mindset of people who think in short video clips, text messages, and collaborative gaming. The new generation workforce is culturally aligned with Agile, and quite comfortable with rapid iteration and feedback.

CHANGE: Almost every aspect of the world surrounding product developers is changing at a faster pace than ever. Supply chains, prices and availability of raw materials, volumes of production and demand, workforce composition and location, which products to develop, war, pandemics, environmental and social governance, are just some of the factors organizations must now consider in their corporate planning. Product developers simply must respond to almost constant change, and Agile product development works very well in this tumultuous environment.

TOOLS: Today the product developer has a range of tools available to support the agile process for hardware. Cloud-native engineering tools are finally available for CAD, PDM, PLM, simulation, etc., and they are perfect for Agile. Even more solutions, including non-engineering-specific tools like JIRA, Slack, Miro, Trello, and Smartsheets enable Agile for hardware teams

An Acceleration Toward Distributed Workflows

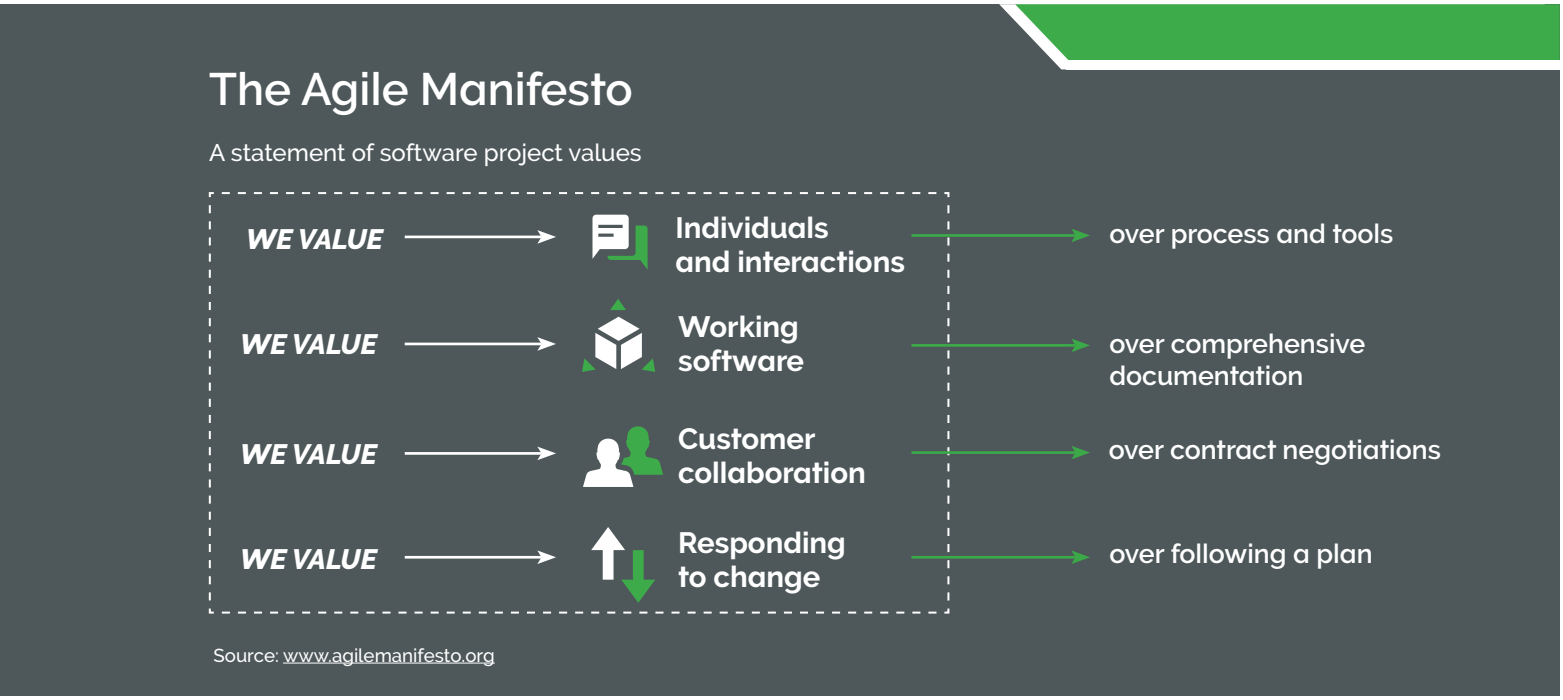
The COVID-19 pandemic added intense pressure to shift professional workplaces to more flexible ways of working. Many organizations found themselves suddenly operating in highly distributed workflows, all while serving customers with radically changed behaviors. While some frontline employees remain essential for on-premises workflows, many others (including a large percentage of knowledge employees) have shifted to remote, distributed work. Teams that used to function in the same room must now operate – with equal or greater efficiency – from multiple locations, collaborating largely through a shared digital infrastructure.

Fortunately, when the pandemic happened, digital transformation technologies had evolved. It would have been impossible to imagine large-scale deployment and utilization of programs like Miro, Zoom, Microsoft Teams, and Onshape 10 years ago. Now these digital technologies are ready, and each contributes to enabling distributed workflow. All of them, while not Agile-specific, also support many of the principles and values comprising Agile product development.

Agile: A Revolution Born in Software

The ideas and concepts underlying Agile software development circled for several years before the creation of the Agile Manifesto in 2001. The document outlined four core values and 12 principles, but all with one common purpose: To redefine software development in a way that values the developers and their customers, rather than the tools and contracts, and focuses on collaboration, communication, and frequent interaction – all with an eye on creating value, responding to changes, and continuously improving the development process.

Perhaps the largest shift in thinking underlying Agile is the redesign of engineering development work, which traditionally happened in long stages or phases, into shorter cycles called sprints. Sprints are oriented toward long-term goals, but focused on completing work and delivering value in frequent increments. Agile teams report accomplishing more progress in less time. With sprints lasting from one to four weeks in duration, deliverables can be reviewed and deployed with each sprint. After several sprints, planned major milestones and releases can deliver larger increments of value at a deliberate pace.



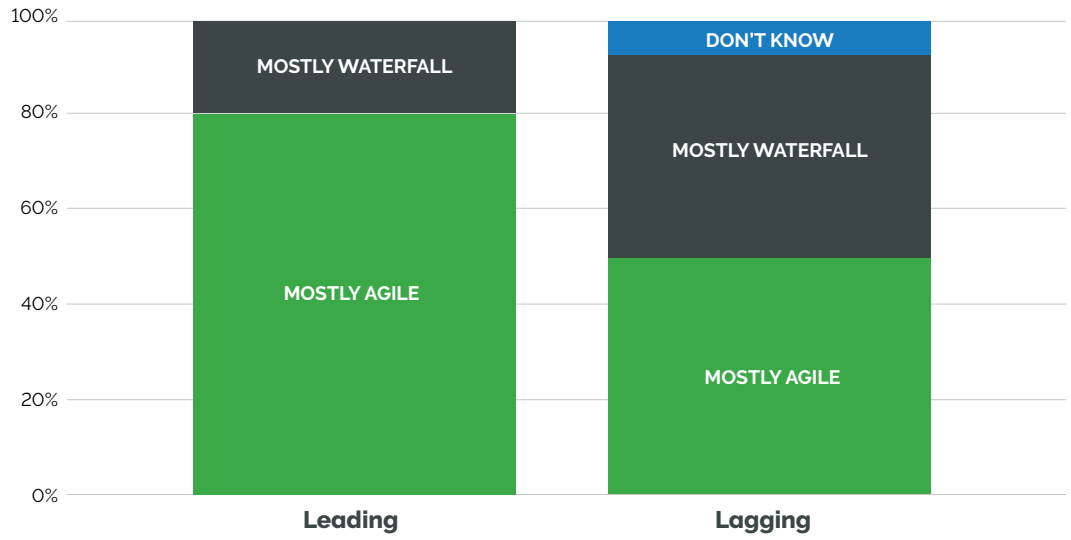
Over the past two decades, Agile development has revolutionized the software industry, starting out as a fringe idea, and becoming a nearly ubiquitous approach to software creation today. While numbers vary, research studies have reported as much as 94% of IT organizations and software teams using Agile development. Digital.ai's 2022 State of Agile report found 80% of software respondents stated they used Agile as their primary development process structure.

Data gathered in PTC's 2022 Executives vs. End Users Survey also showed the prevalence of Agile development. The report questioned an audience that reported itself as 36% IT-focused, 32% product development focused, and 32% manufacturing focused. Overall, 68% reported using Agile. This means that Agile usage has grown well beyond IT to incorporate some of the product development and manufacturing focused activities.

That same survey noted that self-identified leaders (respondents who reported they believed their organization was leading the competition in regard to technology platforms and solutions) were much more likely to say they were using Agile, with 80% answering they had adopted the technology, as opposed to 50% of laggards who said the same. These data reinforce similar findings from McKinsey & Company, which showed that organizations adopting and embedding Agile development reported higher success metrics than those who did not.

EMBEDDING AGILE DEVELOPMENT INTO YOUR ORGANIZATION

Q: How does product design and development typically operate within your organization?



N=123 leading, 78 lagging; ***Source:*** *PTC 2022 Executives vs. End Users Survey*

Agile development, the revolution born in software, has not stayed in the IT industry. As time has passed, and the methods proved effective, more organizations have applied Agile principles and methods in technical areas outside of software and IT. Manufacturing as a segment is one of the more recent frontiers that seeks to implement Agile product development.

Agile Product Development at a Glance

Agile product development refers to adopting and adapting Agile software development techniques for physical product creation. While certain aspects of Agile easily make the transition from software to hardware, other elements require some adjustments to be applied in the realm of hardware. Perhaps the most important adjustment relates to how hardware engineering teams choose to embrace the iterative cycle of short development sprints.

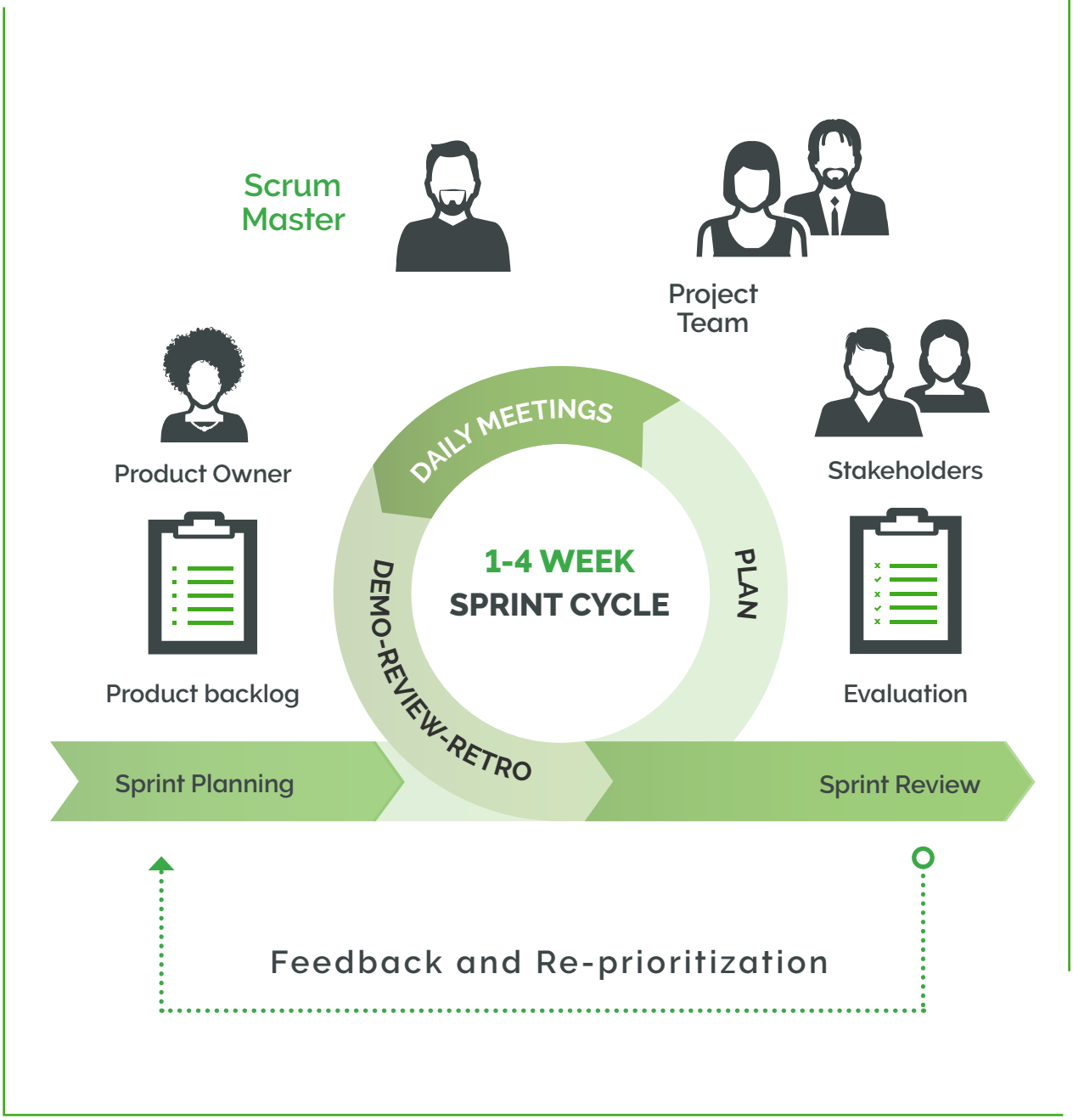
Product iterations in the software space can be developed and tested very quickly, where there are no issues surrounding material cost, supply chain, or assembly time. The idea that a two-week sprint can end with a usable product is more than possible; many software organizations accomplish this regularly through Agile development. Physical products, however, cannot be engineered and produced as quickly, so trying to adopt weekly sprints in hardware design seems impractical or even infeasible. At least, this is the mindset of many engineering teams, and this must change.

While software sprints often end with a potentially releasable increment for users, this is not generally the practice in agile sprints for physical products. Sprints can end with any type of measurable achievement. In the sprint review, the team assesses whether they achieved their sprint goal(s). This is to ensure that progress is made in each sprint. In this way, even developers of complex systems such as automobiles and highly regulated products such as medical devices can and do use Agile product development. Multi-year development programs can be planned as a sequence of three- to six-month milestones, and these further decomposed into multiple short sprints. With appropriate milestone planning and sprint goals, the Agile structure has been readily adapted for physical product creation.

This "adjustment" for agile applied to hardware engineering — reviewable increment rather than shippable increment – is essential, yet it is not the only adjustment that is helpful. Many engineering teams have found it useful to modify the Agile practice of a fixed sprint cadence. Instead, they allow different teams to utilize different sprint durations. For example, a hardware team could use three-week sprints while the related software team could use one-week sprints, and they sync up with a joint demo every three weeks.

Another type of adjustment to standard agile process involves alternating between development sprints and operational sprints. The development sprints involve the work of product creation – design, engineering, testing, etc. Operational sprints, by contrast, focus more on the overhead – product support, customer visits, strategic planning activities, etc. When product developers also need to perform support functions, some organizations find that alternating between these modes can be very effective, eliminates the daily need to switch roles, and reduces the risk of burning out the creative team.

AGILE PRODUCT DEVELOPMENT SPRINTS



Cloud and Agile Empower Distributed Collaboration

Organizations do not need to adopt cloud-based tools to use Agile processes, nor do they need to pair Agile principles with their cloud-native workflows. The two concepts are not directly linked. That said, they are extremely complementary and together provide incredible value, especially when the workforce is distributed.

Take, for example, the common Agile practice of the daily team meeting (a.k.a. morning standup, daily scrum, etc). This 15-minute event maintains effective communication and collaboration within a project team. All team members meet and share what they are working on, the progress they are making toward the sprint goal, and identify any challenges or concerns they may be facing. By incorporating cloud-native platforms, such as Zoom, Microsoft Teams, and Slack, employees from anywhere can share a virtual space for their meeting, thus making it easier to connect on a daily basis.

Another common Agile practice is the use of a visual display of the backlog. By making the future, current, and completed tasks visible to the team, everyone can see the team's progress, what tasks need to be done in the sprint, and how their work relates to that of others. Maintaining the product backlog using a cloud-based graphical tool (such as Jira or Trello) allows product owners and/or product managers to communicate sprint goals and task priorities in an efficient manner – even when priorities change over time – and for the team to refer to the backlog board during their daily meetings.

Yet perhaps one of the greatest examples of Agile and cloud enriching each other occurs with collaboration tools like Google Docs, Sharepoint, Miro, Figma, and Onshape. One of the central Agile concepts is iterative value delivery, and one of the greatest challenges for distributed teams is to maintain alignment and avoid unnecessary work. Multiple employees working on different versions of the same file inevitably leads to confusion and wasted effort.

Building workflows around cloud-native solutions, however, can drastically reduce these failures of coordination. Authorized employees, regardless of location, can access and simultaneously edit the same files, with changes reflected in real-time. Not only does this accelerate the general pace of work, but it also virtually eliminates the risk of errors and rework due to multiple file versions. This allows teams to easily collaborate, rapidly iterate, and then effectively deliver their results with minimal confusion.

Reducing Risk and Uncertainty with Agile

Successful product management goes well beyond simply envisioning the best version of the product possible. Organizations do not necessarily need to be risk-averse but risk-aware. A radical new concept has the ability to derail a company, even the one that eventually succeeds in the market. When teams begin to anticipate the ways a product can fail, they can start to address the potential failure modes as the project develops.

Agile sprints are then as much about exploring and eliminating failure modes as they are about features, performance, and forward progress. In this way, the risk of the project is significantly reduced as uncertainties are addressed in each sprint.

Case Study: Lumafield

Founded in 2019, **Lumafield**, which developed the world's first accessible X-Ray CT scanner for engineers, needs to ensure that every product they create continues to help their organization grow. Their hardware development process is notably fast and they begin by building a simple proof of concept out of off-the-shelf parts. After this, they spend anywhere between two weeks and one month developing and iterating on rapid prototype. The philosophy behind this design process is simple: Engineers have to build products to fully learn lessons, not simply rely on abstracts.

Eduardo Torrealba
Co-Founder
and CEO

> ***"Iterating on prototypes faster than you're comfortable with shows you where the limits are.. It actively stops you from getting bogged down in distractions. Building the whole experience allows you to discover where the bottlenecks are, anticipate them, and ultimately reduce or avoid them all together."***

While Lumafield does not always describe its hardware development process as Agile, it is built on iterative value delivery, frequent check-ins and project benchmarks, and feedback from not just stakeholders within the company but clients. Lumafield is applying many Agile principles and values to reduce the risk and uncertainty in the project design, without adding costly time and delays to the project.

Agile Helps Teams to Deliver Value in Frequent Increments

Following the agile approach to product development, teams can deliver valuable results sooner. While in many software systems, this may entail releasing usable updates to working systems, for hardware it's different. It should mean being able to see incremental progress at the end of every sprint. In a complex system, there are so many components to design, so many suppliers to qualify, so many parts to integrate, and so many tests to run. In each sprint, value is created by handling the most important next steps in a very long sequence of development work.

To achieve incremental value, while also ensuring that the critical end point is reached on time, planning must be done from the top down. Senior leaders identify the major milestones, such as demonstration at a trade show, delivery of a working system to a key customer, or ramp-up of scale production. Working together with product managers and agile team product owners, they can jointly define a series of sprint goal which lead up to achievement of the major milestones.

Case Study: Cummins

Cummins has found success with the agile development approach in delivering incremental value with each sprint, even in larger systems. Cummins produces diesel engines and power systems for use in a range of operational environments almost everywhere in the world. Many system designs require regulatory approval from various entities. These "approval gates" serve as major milestones, with multiple tests, executed in sprint cycles, leading up to each of the major ones. With each successful test result along the way, value is being created, risk is reduced, and confidence is built within the team and its leadership.

Joan Wills, executive director of Cummins Software and Electronics Engineering, has led an expansion of agile usage for embedded software along with working to navigate and define the interface with more traditional stage-gate hardware development programs.

Joan Wills
Executive Director

"Cummins has over 100 years of experience designing and producing our products, but that doesn't stop us from innovating, especially when the advantages are numerous and quickly apparent. Agile software development with clear connections to hardware development programs has allowed us to break down the development of complex systems – not simply into subsystems and components, but also into periodic sprints representing a constant stream of incremental progress."

Scaling Agile for Systems Engineering

Agile product development is not limited to a single team. The concept of Scaled Agile utilizes the structure of time-boxed sprints in a larger product development context in which multiple teams work on portions of the overall system while managing the interfaces between them. Most Scaled Agile organizations address cross-team interfaces through a multi-team planning process.

Case Study: Ocado Technology

UK-based **Ocado Technology** has a very effective implementation of scaled Agile. Ocado's main product is a comprehensive platform for grocery e-commerce fulfilment and logistics. Ocado's technology division designs the custom mobile robots which operate within Ocado's warehouse fulfilment centers to automate grocery order processing. A key aspect of Ocado's internal development is the iterative improvement of these robots, building upon their overall efficiency and capacity.

To implement scaled Agile for development of a complex product, most engineering organizations would assign an agile team to each of the major components or subsystems comprising the product. Ocado allocates roughly 10 teams to 3-week sprint projects defined during the sprint planning process. These projects may involve multiple subsystems or just a few components and result in the "release" of a new bot design every three weeks. For example, one team might be fixing the failure of a spring after 100k cycles. Another team might be improving the belt path for easier maintenance access. Each team's work involves analysis, ideation, design, testing, integration – of a small part of the robot in a 3-week sprint. As the work of each team is continuously integrated into the bot design, internal testing continues on an ongoing basis.

As Ocado's engineering director, Matt Whelan, explains it, "Agile provides great flexibility to unblock the critical development path by defining sprint goals according to the current development status. Perhaps we face an urgent supply chain problem, or we need to provide data or an interface spec to another department, we can put that into the next sprint. We adapt to ensure that one subsystem does not mature before another. This allows for holistic pieces of work to be completed rather than one component at a time." When asked on what particular challenges scaled Agile can address, he went on:

Matt Whelan
Ocado's engineering director

"A challenge with multidisciplinary teams is the fact that different types of engineers can talk different engineering languages. For example, 'sigma' can mean any of: mechanical stress, electrical conductivity, chemical bond type etc. Putting two engineers from different backgrounds in a room and asking them to come to a decision is not always simple, and the conversation needs to start with deciding upon a common language. Our ad-hoc sprint teams mean the mixture of crafts is created at the start of the project, the engineers have the time to understand each other, work with each other, and value each other, such that at the end of the project and point of decision, the conclusions can be sympathetic to each other's perspective."

How to Start Implementing Agile Effectively Today

Any organization or engineering team considering Agile product development should keep this in mind: There are no Agile police. There is no framework demanding that anyone adopting Agile product development must utilize every single aspect of Agile software development, changing as little as possible. Organizations should, first and foremost, prioritize the aspects of Agile that are most relevant to improving their product development efficiency. Here is a straightforward roadmap:

1 Start small when implementing Agile product development

It is important to start small. No organization can go from Agile product development novices to expert practitioners within six months –– not without significant risk of disaster. Agile product development can be used throughout an organization, but should start within one project, in one team, with one clear overall set of objectives. After that team has successfully transitioned to Agile product development, then the organization can start to ramp up the scale, bringing on any and all appropriate expertise necessary to achieve a smooth transition.

2 Choose a relatively easy place to start

Organizations should prioritize starting with a project that will easily transition to Agile development. For example, an organization has two main initiatives. The first is a relatively simple product upgrade, while the second is the development of an entirely new complex hardware system with multiple components – each with its own supporting team. Choosing the former will enable a smaller team to gain valuable Agile experience, which they can then share as the company begins to scale its Agile usage.

3 Try to learn quickly

Agile – while flexible – does have its core concepts and established practices. An engineering team that wishes to adopt Agile but is not willing to utilize time-boxed sprints, backlog management, or frequent feedback from the client will not learn much about Agile. On the other hand, a team which is willing to try out the complete agile toolkit will learn a lot – and can make adjustments in each sprint.

4 Showcase the results of Agile product development

It can be difficult to implement a process change that has impact on the working culture, such as the shift to Agile development. Opponents within an organization will be quick to point out shortcomings or simply note an apparent lack of impact for all the resources that have been applied. To that end, it is very important to showcase the process and results of Agile development. This provides evidence of its value and will attract employees from outside the team with an interest in learning more about Agile development.

5 Ensure the support of executives and managers

Senior management support is essential. Effective organizational change needs motivation and protection from the top down. Senior managers, many of whom are also now becoming very familiar with Agile, can provide valuable goal setting to keep the implementation on track. They can also offer some measure of "protection", allowing the initial agile experiments to succeed with few interruptions.

6 Secure Agile product development experience and expertise

The shift to Agile development may require external expertise. If organizations do not feel they have the appropriate internal knowledge, there are many excellent Agile experts and trainers available outside of the company. Alternatively, key employees from outside of the team but still within the organization (often ones with Agile software experience) can provide coaching throughout the Agile transformation.

7 Understand successful Agile product development takes time

It is important to remember that results take time to be recognized – even with Agile. Teams will not go from inexperienced to experts after just one sprint. Practice is essential, not only for improving efficiency but also for building a strong knowledge base within the organization. This is especially true for any group looking to deploy scaled Agile to develop complex projects.

Are You Ready for Agile?

For many working in physical product development, Agile principles and values have already begun circulating within the organization. This foundation naturally mixes and benefits from the shift to cloud-native software tools and platforms. Simply put: the tools are ready, even if the understanding is still not fully present. There are many in the hardware space already using certain aspects of Agile development without knowing it, but every single engineering team should at least evaluate the potential. Agile can work, but only if you learn the principles, make appropriate adjustments, and give it a try.

Learn More About PTC

Learn more about PTC's cloud-native tools, ideal for Agile product development, here.

ptc®

DIGITAL TRANSFORMS PHYSICAL

ptc.com

PTC, Inc.
May 2023