Cloud-Native Applications and Their Role in Supporting Agile Hardware Development

By

Brian Herrera

Bachelor of Science in Mechanical Engineering
Stanford University, 2009

Submitted to the System Design and Management Program
in partial fulfillment of the requirements for the degree of

MASTER OF SCIENCE IN ENGINEERING AND MANAGEMENT

at the

MASSACHUSETTS INSTITUTE OF TECHNOLOGY

February 2024

Authored by: Brian Herrera
System Design and Management
January 26, 2024

Certified by: Steven Eppinger
Professor of Management Science and Innovation
Thesis Supervisor

Accepted by: Joan S. Rubin
Executive Director, System Design & Management Program
MIT System Design and Management Program

Cloud-Native Applications and Their Role in Supporting Agile Hardware Development

By

Brian Herrera

Submitted to the System Design and Management Program on January 26, 2024
in partial fulfillment of the requirements for the degree of
Master of Science in Engineering and Management

Abstract

Agile product development focuses on collaboration, iterative development, and responsiveness to change as a mindset and methodology for project teams. Agile has been instrumental in software development and improving overall project outcomes for software teams. Agile has recently been introduced to hardware teams, given the benefits experienced with software teams. While Agile for hardware is still in its infancy, there are many aspects of cloud-based applications (e.g., Jira, Microsoft 365, Zoom, Miro, Google Docs, etc.) that are enabling the use of Agile in hardware development. In this research, we explore how cloud-based applications support Agile development for hardware teams. We reviewed existing frameworks and interviewed nine individuals from eight different organizations. We learned that hardware teams are complex and require a high level of coordination between its team members. Cloud-based applications support Agile project teams through collaboration, speed of iteration, flexibility, and alignment. When utilizing these applications, experienced practitioners consider their organizational structure, the team's physical location, and interdependencies with other groups. While cloud-based applications provide several benefits to project teams, we suggest they adapt these tools to fit their specific needs. Future development and integration of these tools may help reduce the number of total applications used to streamline the coordination process and reduce the overhead of tools.

Thesis Supervisor: Steven Eppinger
Title: Professor of Management Science and Innovation, Sloan School of Management

Acknowledgments

Thank you to Professor Steve Eppinger, my thesis advisor, for your guidance and mentorship throughout this process. Your knowledge of product development and exploring Agile for hardware has given me a new way of looking at development cycles and the organization of teams and projects. You were able to help simplify my findings in a way that gave me structure while giving me direction for the next problem to tackle. Thank you!

I would also like to thank my wife, Christina, who, when I mentioned I was interested in a graduate program at MIT, encouraged and supported me even while she was in the middle of an intense cardiology fellowship and we were new parents with a newborn. When I got into MIT, she gave me enthusiasm and support; it hasn't stopped since. I would also like to thank my daughter, Emilia, whose curiosity, love, and playfulness give me purpose, teach me patience, and make me a proud dad.

I would also like to thank the individuals who took the time to meet with me and let me interview them for this research. They shared great insights and introduced me to new concepts and tools that make this research possible and will help me grow as a leader in product development. Thank you to those who helped connect me to my interviewees – the power of networks is huge. I appreciate all your support.

Thank you to the SDM community – the faculty, TAs, staff, and the SDM 2021 and 2022 cohorts. Your support, knowledge, and encouragement made the MIT experience memorable in so many ways. Although relatively short, it is a time I will cherish.

Lastly, I'd like to thank my family – parents, siblings, and in-laws – whose unconditional support typically flies under the radar, but which makes all the difference.

Table of Contents

1. Introduction and Motivation

In the fast-paced project management landscape, Agile project methodologies have become the standard for software development. The Agile Manifesto (Beck et al., 2001) helped promote the mindset and emphasis on collaboration, iterative development, and responsiveness to change to deliver successful software products. The incremental and iterative approach has been instrumental in getting software products to market (Stellman & Greene, 2014). In contrast, hardware development has predominantly remained rigid, with Waterfall being the dominant project management method (Drutchas & Eppinger, 2022).

Within the past decade, however, many organizations working on hardware development have begun to embrace Agile methodologies to gain the benefits experienced in the software industry for their physical products (Eppinger & Hirschtick, 2023). As hardware development organizations have begun to adopt Agile, it has coincided with the widespread use of cloud-based applications, such as Zoom, Jira, Miro, and Microsoft Office 365. Motivated by the growth of Agile methodologies in hardware, this research aims to answer the question: ***How do cloud-based tools[1] support Agile hardware development?***

We use the terms "cloud-based" and "cloud-native" interchangeably to describe applications. However, there are differences in the architectural design of the applications that impact how developers create, maintain, secure, operate, and price the applications (Saez, 2023). Our research is focused primarily on the access of applications through the cloud, and the research will not discuss the applications' architecture moving forward.

We summarize the dominant project methodologies and cloud-based applications used in hardware product development in Section 2. Section 3 reviews the literature to show past case studies of cloud-based tools and Agile in hardware development to explain the current research. We explain the research methodology in Section 4 and present the results of interviews with practitioners in the industry in Section 5. In Section 6, we present a framework to help understand these results. Section 7 concludes with recommendations to practitioners and areas for future research and development.

2. Background

This section will describe some of the more common product development methodologies and cloud-based tools used in hardware development.

[1] We use the term "tools" and "applications" interchangeably.

2.1. Product Development Methodologies

Product development methodologies have evolved as products and systems have become more complex, and new industries have driven new ways of thinking. Two of the most common approaches used have been the traditional Waterfall method and the relatively newer Agile method (Kramer, 2018). Although other methods are used, such as the V-shaped model, Spiral Model, or Big-Bang Model, in this paper, we will discuss Waterfall and Agile methods only. The focus will be on hardware development. In this case, hardware refers to mechanical and electrical devices that must be physically manufactured and assembled. In this section, we will share an overview of each method and its influence on hardware development.

2.1.1. Waterfall Method

The Waterfall method, sometimes called stage- or phase-gate, is a linear and sequential process for developing products. This method establishes all the planning and requirements before starting any design work (Ullman, 2019). These events happen sequentially, with typical phase-gate reviews occurring after each stage. Figure 1 shows an illustration that captures the typical stages of Waterfall style product development. The illustration shows how each stage leads to the next in a waterfall shape.

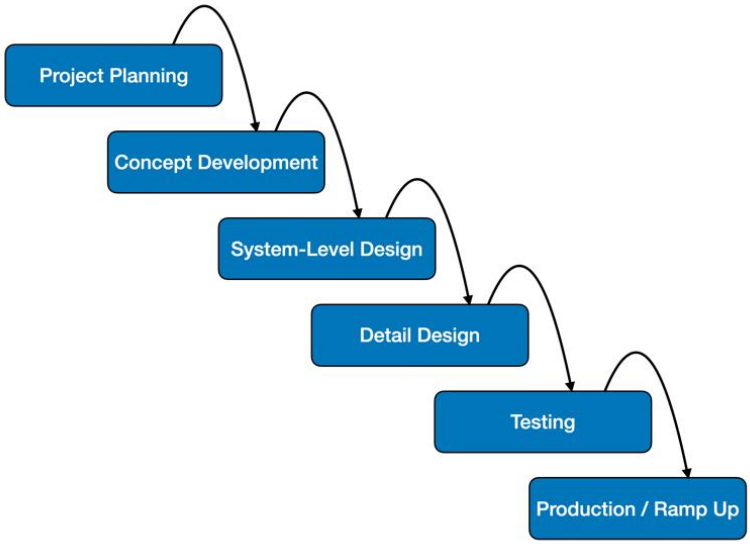

Figure 1: The Waterfall Process. Adapted from Ulrich and Eppinger (2016).

Each phase in the Waterfall method must complete a set of deliverables before moving to the next phase. Typically, a review, or a gate, must be approved after each phase to allow the work to progress to the next phase. This method allows project teams to return to a previous stage if they find issues. However, this impacts cost and schedule as it's considered a deviation from the original plan created at the start of the project (Häring, 2021).

Regarding hardware development, below is a summary of the advantages and disadvantages derived from Kramer (2018).

- Key advantages of Waterfall are that it is (1) simple to understand and manage due to the structure of the stages, (2) predictable and focused on a particular stage at a given time, and (3) effective for small projects with low complexity and low uncertainty.

- Key disadvantages of Waterfall are that it is (1) costly to step back to a previous stage once you begin building and testing, given the long learning cycles, (2) difficult to accommodate new or changing requirements in the later stage of development because requirements are set upfront, and (3) not effective for large projects with high complexity and high uncertainty.

As described, the Waterfall method's rigidity does not make it an ideal solution for all project cases. While the sequential process provides a structured path, it leads to limited flexibility when changes arise, and challenges when adapting to unforeseen issues, which are very common in hardware development.

2.1.2. Agile Methodologies

As an alternative to the Waterfall method, Agile methodologies emerged in software development due to the need for collaboration, adaptability, and flexibility (Pargaonkar, 2023). The Agile Manifesto contains the core values and ideas that guide effective development teams with its focus on "individuals and interactions," "working software over comprehensive documentation," "customer collaboration over contract negotiation," and "responding to change over following a plan" (Beck et al., 2001). Although initially conceived for software development, Agile principles have been adapted to other fields and industries, including hardware development.

Even though Agile has a set of methodologies, it is also a mindset that focuses on principles of sharing in the planning, design, and process, working together to eliminate barriers as projects evolve, and improving and iterating the product and team dynamics (Stellman & Greene, 2014). The Agile methodologies leverage these principles. In this section, we will explore two standard Agile methods, Scrum and Kanban.

2.1.2.1. Scrum

The Scrum methodology uses a lot of specific terminology to describe its process. We will use the process and terminology throughout the paper and have summarized it below. Figure 2 provides a visual overview of the Scrum process (Ullman, 2019).

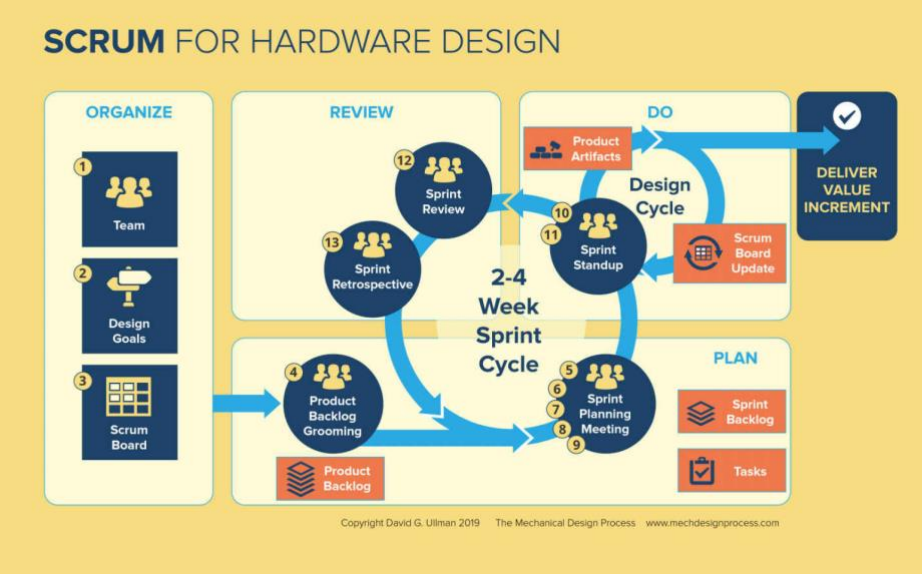

Figure 2: The Scrum process in detail. Image sourced from Ullman (2019).

Scrum takes a project and breaks it down into a series of fixed timeboxed iterations called **sprints**. These sprints typically last 2 weeks, but some organizations may extend them to 4 weeks, given specific organizational requirements or previous learnings. A **Product Owner** will work with customers to write and prioritize the **user stories** that will generate **tasks** the team needs to complete. Grouping a series of user stories or tasks can create an **epic** that aligns with a more extensive project deliverable.

All the different tasks will create a **backlog** of work that a **Scrum Master** will manage with the **technical development team** to map out into different sprints throughout the project. The Scrum Master will list the specific tasks for a particular sprint on the **Scrum board** and capture the completion for the team.

Ahead of each sprint, the Scrum Master will work with their team to do **Sprint Planning**, selecting the specific tasks in the backlog to tackle for that sprint. To appropriately plan, the team assigns **story points** to each task to estimate the effort required to complete it. As the team completes tasks, they measure their **velocity** and how many points they complete per sprint and utilize **burndown charts** to visualize the amount of work completed over a period of time.

The Scrum Master will hold **daily standup meetings** with the team to discuss status and barriers, where teams are encouraged to raise concerns, request help, and acknowledge significant milestones and findings. At the end of each sprint, the team conducts a **sprint review** to discuss each task's sprint outcomes and disposition. Incomplete tasks move out to a future sprint, depending on priority. Lastly, and most importantly for collaboration and continuous iteration, the team holds a **retrospective** to discuss what went well, what needs to improve, and address other aspects of the sprint or project (Stellman & Greene, 2014).

2.1.2.2. Kanban

The Kanban methodology helps improve the product development process for teams by visualizing work, limiting work in progress, and managing flow. It leverages Lean mindset values that originated with the Toyota product development process in the late 20th century (Stellman & Greene, 2014). Lean focuses on (1) eliminating waste - activities that don't add value to the product, (2) building integrity in a product to meet users' needs, and (3) delivering as fast as possible by eliminating barriers (Hammarberg & Sunden, 2014).

Kanban typically works on top of existing project methods but serves to help understand the system and how items flow through the process (Stellman & Greene, 2014). Kanban methodologies use a **Kanban board** to visualize the team's workflow. The items on the Kanban board are **work items** that get moved along the different columns of the board as they progress through the workflow stages.

Work in Progress (WIP) limits are critical to Kanban, enabling teams to control the amount of work at each stage to ensure a smooth flow through the system (Alaidaros et al., 2021). Teams control which stages require limits and determine the limit count based on their historical and continuous improvement experiences. Teams use a **pull system** to add new items and focus on the highest-priority work as the items move continuously through the stages. As seen in Figure 3, the work items flow along the board from left to right, with some columns having a maximum limit of Work in Progress (WIP) to trigger alerts and highlight bottlenecks (Andrei et al., 2019).

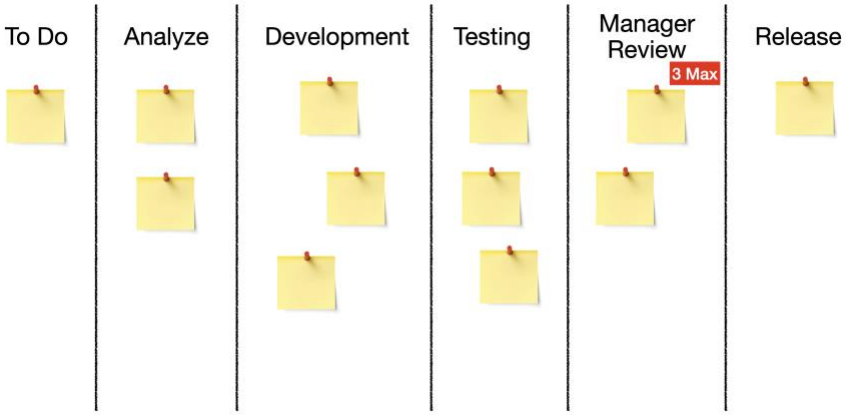

Figure 3: The Kanban board with different stages and work items. Adapted from Hammarberg and Sunden (2014).

Although Kanban and Scrum have distinct elements, they share many of the same principles but practice them differently, as explained in comparative research by Ozkan et al. (2022) and Saleh et al. (2019). For example, Kanban and Scrum both have visual boards to track different stages of work; they similarly share a concept of work limits through WIP Limits and Sprint Planning; they leverage a prioritization of work through their respective pull system or backlog review; and they meet and collaborate to discuss work items and continuously improve their process.

The critical difference between the methodologies is that Kanban is not explicitly timeboxed, like Scrum; however, as work items pile up in specific stages of the workflow process, WIP limits help drive discussions to eliminate blockers.

2.1.3. Tailoring Project Methods

Applying Agile methodologies to hardware development is a nuanced approach, as project teams must understand their projects' size, complexity, uncertainty, and variability (Schmidt et al., 2018). While Agile's flexible, collaborative, and iterative nature allows it to be beneficial in early prototyping and feasibility stages, where failures and changing requirements are common, later stages of programs, especially those requiring regulatory oversight and long lead time items for manufacturing, introduce difficulties with strictly adhering to Agile.

2.2. Cloud-Based Tools

The product development methodologies described in section 2.1. have evolved as new industries and technologies require new structures and processes to be more effective. Alongside the evolution of project methodologies has been the development of cloud-based applications shaping current project management practices. These cloud-based tools are connected online, updated in real-time, and accessible across multiple mobile and desktop devices from anywhere with an internet connection (Calefato & Ebert, 2019). This section reviews the critical cloud-based applications leaders and teams use to run their projects. The three main categories of applications discussed are communication tools, lifecycle management and workspace tools, and design and visual collaboration tools. Each of these sets of tools plays a crucial role in hardware development.

2.2.1. Communication Tools

Hardware development is a collaborative process requiring multiple disciplines to discuss interconnected aspects of their project. The need to collaborate makes communication critical to the success of the project.

Communication tools, such as Slack, Zoom, and Microsoft Teams, have become central hubs that allow teams a wide range of capabilities to discuss and share information (Calefato & Ebert, 2019). Slack is a popular real-time messaging tool that has expanded its feature set to include file transfer, audio and video calls, and integration into other platforms (Slack, n.d.). Slack helped to popularize many communication features seen in other tools today, such as channels, and continues to be a leader in integrating with different lifecycle and workspace tools (Calefato & Ebert, 2019). Zoom is predominantly an audio and video conferencing application that saw increased usage during the COVID-19 pandemic in 2020 when remote work required more video communication (Schmidtner et al., 2021). Microsoft Teams, a platform within Microsoft's 365 suite, continues to expand its features of instant messaging, audio and video calling, conference call capabilities, and file and data sharing (Arimoto, 2023). Teams' seamless

integration with other commonly used Microsoft tools focusing on organization productivity has been crucial in its popularity and use (Schmidtner et al., 2021).

Cloud-based communication tools facilitate real-time discussions and collaboration, providing flexibility regardless of where team members are located.

2.2.2. Lifecycle Management and Workspace Tools

Efficient lifecycle management and workspace tools are critical to managing the complexities of hardware development with multiple teams and various types of documentation needed. The increased complexity of today's projects requiring multiple disciplines (i.e., mechanical, electrical, manufacturing, test, and firmware) and subsystems has made alignment and collaboration critical within teams.

Lifecycle management and workspace tools, such as Jira, Confluence, and Microsoft 365, enhance collaboration, provide visibility into progress, and streamline workflows. Jira, a tool developed by Atlassian, helps to manage project activities, schedule and track work, report and resolve issues, monitor progress, and keep historical information available (Wright, 2023). Jira's use of Scrum and Kanban boards makes it very adaptable to task management for Agile teams, with analytics to capture key Agile metrics, such as burndown charts and project velocity (Calefato & Ebert, 2019). Confluence, a tool developed by Atlassian, is a wiki-type workspace that serves as a knowledge center for teams using pages to store content within a team's space (Atlassian, n.d.). The workspace allows teams to easily create, store, and edit information, especially meeting minutes and critical decisions (Calefato & Ebert, 2019). Microsoft 365 has a suite of tools such as Word, Excel, PowerPoint, SharePoint, and others that allow for robust workspaces and documentation to collaborate in real-time, internally and externally (Microsoft Inside Track, 2023).

Cloud-based lifecycle and workspace tools enable real-time collaboration and accessibility that ensures team members have the information they need to be productive in their tasks and deliverables to complete their work and easily share with their team.

2.2.3. Design and Visual Collaboration Tools

Design and visual tools are critical to hardware development teams by offering opportunities to ideate and visualize concepts in a dynamic digital platform. These tools provide a blank canvas to be creative, organize thoughts, and showcase their work.

Design and visual collaboration tools, such as Miro, Figma, and Onshape, enable teams to sketch and build out designs as they work to develop new workflows, solve new problems, or design user interfaces and physical systems. Miro is a tool that serves as a whiteboard that fosters visual collaboration on product development workflows, brainstorming issues, organizing content, or strategizing future roadmaps with teams (Miro, n.d.c). Miro's ability to integrate

with other tools, such as Jira, allows it to be versatile in providing a blank canvas that feeds tasks and ideas into other cloud-based tools. Figma is a tool that offers a platform to develop user interfaces with teams by prototyping the design of user experiences (Figma, n.d.). Although Figma may be tied to coding and UI/UX, its collaborative nature allows hardware individuals to understand the use case and interdependencies of products that have software UX displays. PTC's Onshape is a cloud-native CAD program that enables easy collaboration between multiple users to simultaneously design and review 3D models, with the ability to allow in-model comments to aid in discussions of hardware design (Walsh, 2021). Although many organizations may still use traditional file-based CAD that requires check-in and check-out processes, cloud-native CAD systems enable real-time collaboration to provide new opportunities for team dynamics and accessibility.

These design and visual tools contribute to collaborative design efforts, more efficient iteration cycles to improve speed, and flexibility with teams to access and view files.

This thesis will explore how different Agile hardware development teams have integrated cloud-based tools into their workflows to support their organizational goals in managing the delivery of new products. In the next section, we will review the existing literature for Agile methodologies in hardware development and the impact of cloud-based tools on Agile teams.

3. Literature Review

3.1. Agile Methodologies in Hardware Development

Before discussing the impact of cloud-based tools on Agile development, it is essential to understand how Agile methodologies have been adapted and applied in the context of hardware development. The literature on this topic is relatively new, given that Agile's use in hardware has become popular in the past decade. Developments in hardware and globally distributed teams have shown that agile methodologies are "adoptable and adaptable" with different groups (Ebert & Paasivaara, 2017). The literature provides challenges and opportunities associated with implementing Agile principles in the physical world of hardware development.

3.1.1 Challenges of Agile in Hardware Development

Several researchers have conducted interviews and case studies with organizations using Agile, or aspects of Agile, within their project management structures. The researchers have captured feedback that sheds light on Agile adoption. Schmidt et al. (2018) described product development as a volatile, uncertain, complex, and ambiguous (VUCA) environment because of the difficulty of planning based on all the unknowns. Agile's ability to "cope with ever-changing boundary conditions" that VUCA environments create leads to its adoption over more traditional Waterfall approaches (Atzberger & Paetzold, 2018).

However, the direct adoption of Agile, designed for software management, does not translate directly into hardware management and reaches various limits that present challenges for teams (Heimicke et al., 2019). Drutchas and Eppinger (2022) identified and narrowed down their findings into three critical challenges of Agile in hardware development: constraints of physicality, learning cadence, and backlog creation.

The constraints of physicality, coined by Ovesen (2012), highlight that building physical objects demands coordination with multiple teams (internally and externally), the interconnectedness of complex systems, and large manufacturing and test equipment that limit the ability to frequently iterate (Drutchas & Eppinger, 2022; Ullman, 2019). Physical products may have connections that, if changed, cause significant ripple effects throughout the system, requiring large changes across multiple teams (Ullman, 2019; Atzberger & Paetzold, 2019). Physical products also have more complex lifecycles involving more disciplines, such as manufacturers, test labs, distribution channels, warranty and support, and end-of-life disposal, making managing them much more challenging (Radeka & Iberle, 2022).

In the case of sprints in Scrum, the learning cadence is a short 2–4-week interval intended to produce a shippable product as its output based on software development practices. Expecting a physical product within a single sprint to produce such an outcome becomes difficult, and establishing similar goals and acceptance criteria to software may lead to ineffective hardware planning (Ovesen, 2012; Atzberger & Paetzold, 2019; Drutchas & Eppinger, 2022). Expanding the sprint length beyond 4 weeks, however, is not the solution as it leads to a lack of focus and prioritization due to work typically expanding to fill all available time, which makes appropriate sizing of tasks a critical challenge (Radeka & Iberle, 2022). In addition, criteria for the “definition of done” need to be established within organizations to evaluate benchmarks and set deadlines on incremental approaches (Michalides et al., 2023). Establishing sprint goals and producing valuable outcomes is the challenge with the learning cadence of hardware teams.

The backlog creation and pull-system for Kanban are important in Agile as they establish the set of tasks and items the team prioritizes and works on throughout the project. However, backlog creation presents a problem with large complex systems because it “doesn’t have a good way of tying related items together,” which impacts highly connected systems (Radeka & Iberle, 2022). For large organizations developing complex products, the level of dependencies increases and significantly influences the synchronization within teams and priorities (Michalides et al., 2023). These issues are exacerbated as teams outside of hardware, such as software, need to integrate with the system to launch the complete product (Drutchas & Eppinger, 2022).

3.1.2. Strategies for Successful Agile Implementation in Hardware Development

Building on the challenges, researchers have provided insights into strategies for successfully implementing Agile methodologies in hardware development.

In evaluating the physical constraints, the research focused on moving away from having to deliver a physical product at the end of each sprint or stage of the process and instead focusing on value to the team. Agile hardware teams must focus on knowledge gaps as the valuable learnings within the incremental steps of the process (Drutchas & Eppinger, 2022; Radeka & Iberle, 2022). Understanding which aspects of the product present the highest risk and prioritizing those learnings helps provide context for the proceeding steps with the project (Drutchas & Eppinger, 2022; Ovesen, 2012). Similarly, project teams need to understand which knowledge gaps come with the highest uncertainty and highest cost of change, and ensure that these "new, unique, and difficult (NUD)" learnings are addressed first (Radeka & Iberle, 2022). By focusing on the highest risk, impact, and uncertainties, teams can tackle critical knowledge gaps by leveraging the iterative aspects of Agile to eliminate long, slow loopbacks.

In evaluating the challenge of learning cadences, the research suggests that there are several adjustments that teams can make to traditional Agile software development based on the specifics of the team and the complexity of the physical product. Teams can leverage a style of decoupling from other internal groups to run their sprints off-cycle in which they run a 4-week sprint while their closely integrated group runs sprints in a 2-week cycle (Drutchas & Eppinger, 2022). Drutchas and Eppinger (2022) argue that this style allows teams to effectively partner while understanding that different groups, given specific components or manufacturing complexity, may require additional time to generate the knowledge needed for a sprint or portion of the project. Teams may also want to avoid being time-bound and focus on the flow of work through the development process, in which case they leverage more Kanban approaches than Scrum sprints (Drutchas & Eppinger, 2022). Similarly, the daily stand-up may not necessarily be appropriate for teams with tasks that typically don't change daily, which has led teams to switch to every other day or three times a week stand-ups based on the technical constraints of the team (Radeka & Iberle, 2022). Sprints can also be coupled to showcase a series of events that are closely connected, such as completing simulations, generating drawings, partnering with manufacturers, and completing manufacturing quality checks as individual valuable goals for sprints, all of which provide knowledge to the team and build confidence as they proceed through the process (Drutchas & Eppinger, 2022; Radeka & Iberle, 2022).

In evaluating the challenge of backlog creation, the research has focused on a higher-level project planning period to help create and make Agile effective for physical systems. Combining aspects of Waterfall and Agile in an "Agile-Stage-Gate" hybrid proved to support overall Agile use in physical products (Cooper & Sommer, 2018; Ullman, 2019). In both Cooper and Sommer (2018) and Radeka and Iberle's (2022) research of a hybrid approach, the project leaders should focus on creating a multilevel plan that leverages a waterfall phase-gate approach as a high-level roadmap, decomposing it to major milestones with partners, and decomposing this to planning for specific knowledge gap activities. The individual activities at the lowest level generate the value that flows up to the major milestones and phases. This helps tie multiple teams' activities together and build awareness of interconnectedness and clarity for the larger group (Cooper & Sommer, 2018; Radeka & Iberle, 2022). Drutchas and Eppinger (2022) also identified how integration points in a project plan allowed for a better understanding of the

future planning of sprints to prioritize work effectively to target those dates. As teams understand the higher-level and longer time horizon goals, they can understand and work towards creating “emulators and simulators” as part of the backlog planning to generate learnings ahead of major milestones (Radeka & Iberle, 2022).

Understanding these challenges and strategies is crucial to enable the benefits of Agile – collaboration, iterative development, and responsiveness – in hardware development. It is equally important to contextualize these challenges to understand how teams leverage cloud-based tools to mitigate them.

3.2. Impact of Cloud-Based Tools on Agile Software Teams

The literature on cloud-based tools predominantly focuses on their impact on Agile software development. Agile software development can provide an understanding of the use of cloud-based tools for general Agile needs but will lack information and discussion around the implementation when physical hardware products are involved.

For many years, distributed teams have leveraged cloud-based tools to work more effectively with team members in different locations (Calefato & Lanubile, 2016). However, the onset of the COVID-19 pandemic in 2020 generated an even greater need for cloud-based tools, given the changes that knowledge teams and workforces had to make to continue to meet business demands in light of globally tragic events (Mancl & Fraser, 2020).

According to Calefato and Lanubile (2016), critical to team dynamics, particularly distributed teams, is the need for individuals to have situational awareness of their environment. Situational awareness provides individuals with an understanding of their workflows, coworkers’ availability, knowledge of project priority, and the necessary information to effectively complete their tasks (Calefato & Lanubile, 2016). Cloud-based tools focused on communication and workspace workflows enable this situational awareness within their platforms through their collaborative nature and compilation of information (Calefato & Ebert, 2019).

The research highlights that no single tool captures an Agile team's needs but a combination of tools to support various processes and activities of teams (Calefato &. Lanubile, 2016; Ozkan & Mishra, 2019). The integration between multiple tools through the development of APIs has enabled teams to improve their efficiency and speed, further leveraging the overarching principles of Agile (Ozkan & Mishra, 2019). However, one of the concerns raised with the number of tools and features in use by a group is that teams and individuals may have “information fragmentation” and “channel overload” if organizations do not establish rules for modes of engagement (Calefato & Lanubile, 2016).

The research on the impact of COVID-19 on Agile teams focuses on the adoption and improvement of cloud-based tools to enable remote collaboration (Mancl & Fraser, 2020; Schmidtner et al., 2021). COVID-19 drove teams to work in remote and hybrid settings that necessitated the need for cloud-based tools, where otherwise such exploration would not have

occurred (Schmidtner et al., 2021). Collaboration and communication, fundamental principles of Agile methodologies, led groups to quickly adopt communication-level tools, such as Zoom and Slack, as substitutes for face-to-face interactions (Mancl & Fraser, 2020). The tools also expanded with scaled solutions, improved usability, and competitive costs, making them more accommodable and effective for teams (Schmidtner et al., 2021).

3.3. Literature Summary and the Current Research

Implementing Agile methodologies within hardware development brings the benefits experienced in software development – collaboration, responsiveness to change, and flexibility. However, adopting Agile is nuanced and must be adapted by teams to overcome the challenges of physical constraints, learning cadence, and how the backlog is created and prioritized. Cloud-based tools, however, are enablers that support the Agile process through their collaborative design. The growth and adoption of cloud-based tools post-COVID-19 pandemic have made them ubiquitous with many teams, and their features are universal to improve productivity and enable a wide range of workstreams.

Though there is individual research on Agile hardware teams and cloud-based tools for Agile software teams, there is a lack of research on how Agile hardware teams utilize cloud-based tools. Using interviews with practitioners within Agile hardware development teams across multiple organizations, we will attempt to create a framework to capture how cloud-based tools support Agile methodologies in hardware development teams.

4. Research Method

Our research focused on understanding the impact of cloud-based tools on Agile hardware product development teams. We utilized a qualitative case study approach to conduct our research. We held semi-structured interviews with individuals from eight hardware organizations whose teams were using Agile methodologies. The interviews focused on:

(1) How the organization used Agile in its product development
(2) What cloud-based tools are used for team dynamics and project management
(3) What benefits are experienced from the tools in enabling Agile processes
(4) What challenges are experienced from the tools in hindering Agile processes

We collected data between September and December 2023 through interviews with nine individuals across eight different organizations, which we summarize in Table 1. The organizations interviewed span various industries, company sizes, and Agile styles. One of the individuals interviewed provided feedback from two different company experiences, and their two experiences are listed separately in the table.

We focused on interviews with team members explicitly working on the hardware development of products, although many of the organizations operated both hardware and software teams. Not all interviewees were currently working at the organization they discussed but were previously employed by it.

The interviews were conducted on a non-confidential basis to encourage open discussion. Participants were assured that their responses would be anonymized to ensure the confidentiality of individual perspectives. Interviews were recorded, transcribed, and reviewed to identify tools, styles, mechanisms, and themes. In some cases, we conducted follow-up interviews to elaborate on additional points to better understand the structure and influence of cloud-based tools on teams.

Section 5 shares the results as case studies, with quotations and insights from the interviews. Section 6 introduces the framework and captures the more significant discussion points from the interview case studies. Section 7 provides conclusions for practitioners and further research into this area.

Table 1: Case and Interview Details

Case	**Industry**	**Interviewee**	**Agile Method**
Siemens	Technology and Automation	Engineering Program Manager	Scrum
iRobot	Consumer Robotics	Manager, Hardware Testing	Scrum
Ultraleap	Consumer Electronics	Hardware Team Lead	Scrum
Inertia	Product Development Engineering Consulting	President	Scrum
		Product Development Manager	Scrum
Dell	Consumer Electronics and Information Technology	Principal Product Development Engineer	Kanban
		Principal System Architect	Kanban
Symbotic	Warehouse Robotics	Manager, Robotics Test Engineering	Kanban
Volta Labs	Biotechnology	Technical Program Manager	Kanban/Scrum Hybrid
Root AI	Agricultural Robotics	Director of Hardware Development	Scrum

5. Results

All quotes presented within each specific case study are attributed to the unnamed interviewees from their respective companies. The anonymity of participants is maintained to uphold confidentiality and encourage candid responses. Quotes are used with explicit

permission from the interviewees, who were informed about the research objectives and the intended use of their statements.

In addition, images of cloud-based tools are shared as representations of the interviewee's use-case. The content of the images is not related to any work by the interviewees and serves only as context for the cloud-based tool's features and capabilities as used by practitioners.

5.1. Siemens

5.1.1 Overview and Agile Details

Siemens is a large multinational conglomerate that delivers systems and services across various industries, such as infrastructure, manufacturing, transport, and healthcare. Siemens has been in business for over 175 years and, as such, has gone through transformations throughout its existence. One transformation has been the transition to Agile for its hardware development teams within the past decade.

The Engineering Program Manager (EPM) interviewed was part of the Agile transformation within the hardware group, developing large complex systems that required a multidisciplinary hardware team in a highly regulated industry. A significant driver to adopt Agile was that "hardware teams needed to acquire [Agile] ways of working […] to keep up with the software side because the systems [Siemens] was developing had a very tight coupling between the software and the hardware." Software teams were already well structured to operate in Agile Scrum processes, whereas hardware teams ran rigid waterfall phase-gate processes. Cadence was the critical motivator for change due to differences between software and hardware teams. "[Hardware teams] wanted to avoid the big gap that we had when our software team was ready, but [hardware] was not ready with the design, […] sometimes with a six-month gap." The team's challenges in integrating software and hardware led to a top-down decision from leadership to go through an Agile hardware transformation prior to 2018.

Critical to adopting Agile was the organizational changes and training requirements needed from all team members to go through formalized Scrum training. Team members were extensively trained in the methodologies, and Siemens created roles such as Scrum Master and Product Owner across the organization. The hardware team ultimately operated a Scrum process that used 4-week sprints. With software teams standardized with a 2-week sprint cadence, the organization would end up doing "double sprint" integration, where every 2nd sprint for software (every 4 weeks) was an integration point with hardware.

Scrum teams vary between 5-6 members to 10-12 members, depending on the exact technology and risk level associated with the project. Individual scrum teams are distributed, with different technology expertise in different areas; however, most team members are in the same place and can easily meet in an office or lab space. Distributed teams require similar types of test and hardware prototypes regardless of which area or office they are located in. The large complexity of the systems in development also leads to breaking down teams into subsystems.

The various Scrum teams operate independently with “almost daily” standup meetings, individual backlogs, sprint reviews, and retros. However, the Scrum Masters would meet multiple times weekly in Scrum of Scrum meetings to discuss interdependencies.

However, the highly regulated nature of the products in development leads project managers to create more extensive multi-level project plans with phase-gate events. Siemens follows a similar approach to the Agile-Stage-Gate method described by Cooper and Sommer (2018), using stage-gates throughout the project while using Agile Scrum at the task completion level. Siemens’ program managers create the higher-level phase gate plan that decomposes to the individual Scrum levels of hardware and software teams. This ensures quality, clarity, and visibility in developing complex systems. The Engineering Program Manager highlights that “the most effective place to use Agile is in the early phase to speed up the process, […] when you’re in that early feasibility phase where there’s too much unknown, you want to fail fast.” A purely Agile process without phase gates, however, is difficult because Siemens “can’t compromise the phase gates […] phase gates are put in place to make sure that we have quality […] we have to keep the highest standards in place because of our [customers].”

5.1.2. Cloud-Based Tool Strategy

Siemens' Agile hardware teams leverage cloud-based tools to collaborate across large, interconnected, distributed groups. A list of the top-used tools is below:

- Microsoft 365 suite for workspace documentation
- Microsoft Teams for communication
- Siemens proprietary software for project management and issue tracking
- Proprietary CAD for hardware design

The Microsoft 365 suite, with Word and PowerPoint, allows the team to collaborate on critical documentation regardless of geographical barriers. At the same time, Microsoft Teams provides the flexibility and speed to enable a quick chat or call to discuss items that are barriers to progress. Figure 4 shows an example of the Microsoft Teams channel interface and how teams organize communication channels for different projects.

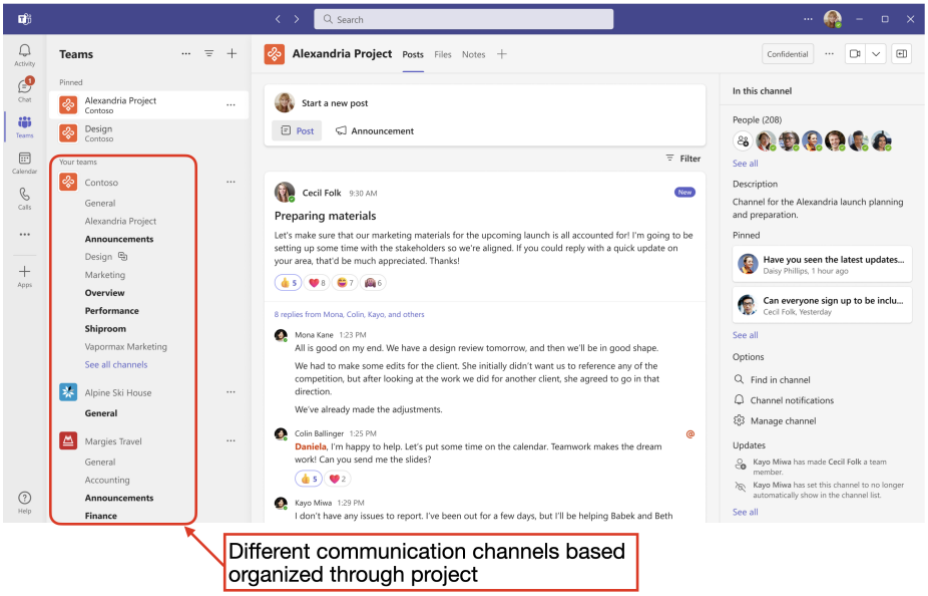

Figure 4: An example of Microsoft Teams channel communications with listings of different channels and the team members tied to the channels. Image source edited from Microsoft Tech Community (n.d.).

In addition, team members consistently provide updates to their tasks in their proprietary Jira-like software that gives all team members visibility to metrics and status updates on project progression in custom-built dashboards. Program managers utilize the dashboards created in their Jira-like software daily to determine topics to discuss in standup meetings and to quickly identify the issues that will escalate to executives at different levels. By connecting tasks to different sprint goals and project goals, dashboards automatically update progress on charts as teams complete tasks, allowing program managers to identify idleness in task completion and see comments to understand the issues. The automation in the tool provides Siemens alignment in a central hub of information and speed in identifying barriers and risks that require attention and immediate action.

The EPM specifically mentions how Microsoft Teams has been able to make conference calls more engaging. Before COVID, team members would call into meetings, typically joining in from a conference room if located with a team or joining in remotely. There was variability in how team members joined the meetings – not all had access to video or the ability to share a screen. Post-COVID, feature improvements within Microsoft Teams made the meetings more consistent, and the cultural norms of video calls made it so “everybody goes on video.” Figure 5 shows a representative Microsoft Teams video call with team members discussing a retrospective as an example of the tool's experience (Wong, 2020). Post-COVID, the Scrum meetings became “a lot easier” because the team members leveraged the communication tools to collaborate more effectively.

Figure 5: An example of a Microsoft Teams video call with multiple team members and retrospective discussion. Image source from Wong (2020).

Cloud-based tools supported Siemens' Agile transformation in their hardware team by providing them collaboration, speed, flexibility, and alignment within their team, allowing them to gain Agile's benefits regardless of their teams' global distribution.

5.1.3. Current Challenges and Ongoing Improvements

The EPM mentioned that utilizing cloud-based tools within Agile frameworks has not come without challenges.

"The way that people communicate, the way people write, can be left to many interpretations." This is particularly an issue as status updates and key findings are primarily communicated within the Jira-like cloud-based tool, and any subsequent discussions exist in Teams' channels or direct messaging. Suppose someone is not working on a particular task but is adjacently connected to it. In that case, they may open the ticket, see the comments, and "not know what is meant by [the final closing comments]," suggesting that "sometimes people just click, click, click, because they want to close it out to move on to the next." They emphasize that it is "more and more important to have the right picture captured and communicated upwards."

The EPM's pain point with Agile for hardware is assigning the "right commitment" to a sprint when estimating the story points for a task. In Scrum, story points are intended to reflect the complexity, effort, and risk involved with the tasks the team is working on. If story points are assigned incorrectly, it can lead to missed sprint goals and task completions. Even though this EPM would sit with the team and try various estimation methods, such as t-shirt sizing (estimating using a relative comparison to other tasks) or Fibonacci sequence (assigning story points with Fibonacci sequence values only), they would "mostly not" be correct. "Sizing work […] and sizing sprints is always the challenge." The EPM suggests that a tool utilizing artificial intelligence (AI) by gathering historical data across similar projects and risk profiles from teams to right-size work and right-size sprints would be valuable for teams. This can drive focus to solving the technical problems while the tools provide recommendations on planning.

5.2. iRobot

5.2.1 Overview and Agile Implementation

iRobot is a consumer robotics company predominantly located in the Greater Boston area. One of the company's most popular products is the robotic home vacuum cleaner, Roomba. iRobot was founded in 1990 and develops hardware and software for its products.

The Manager of Hardware Testing interviewed was part of a highly integrated team between hardware and software, whose work was reliant on assembled hardware and associated developed software to support the integrated test of the systems. iRobot has a "strong software presence," and its "ability to work hardware into an Agile methodology" allowed it to "move fast in terms of development."

iRobot teams are co-located in the Greater Boston area and, before COVID, worked predominantly in-person. The hardware teams are 8 – 15 members that span the typical hardware disciplines of mechanical, electrical, and test engineering.

The team uses Scrum with sprint planning, daily standup meetings, sprint reviews, and retrospectives. Like Siemens, they utilize higher-level program timelines with strategic milestones for product goals, and hardware milestones for custom tooling, arranged in phase gate developments. The team must be aware of the interfaces of its system to plan accordingly for "embedded software that's going to have milestone deliveries that are based on hardware," as it can impact the layering of tasks and interdependence within teams.

The physical systems introduce a lot of "snags" into the Scrum process because of the difficulty in "dealing with external teams getting materials in, shipment, receiving, and quality for something custom you've designed." The hardware teams have had to learn to "look at a project, break down the work, and then create relatively sized buckets to do your estimation for point-based systems."

5.2.2. Cloud-Based Tool Strategy

iRobot's Agile hardware teams leverage cloud-based tools to collaborate across their multidisciplinary teams and provide visibility, alignment, and speed. A list of the top-used tools is below:

- Jira for workstream and project management
- Microsoft SharePoint for documentation
- Microsoft Teams for communication

As part of a critical integration point for the robotic systems, the Manager of Hardware Testing notes that Jira is fundamental to their workflow as it is the "place where we can basically

collaborate online, post both [test] logs and other data that might be leading to root cause being hardware or software." The test team can run their tasks and easily collaborate with other teams in their organization to triage issues. Those teams, whether software or hardware, will then link the Jira items identified as issues into their own sprint backlogs. Although a lot of the "software development and hardware development is handled separately in multiple [projects] of Jira, […] you can still link the development tickets […] and go to the multiple stories that were generated regarding the root cause." In Figure 6, Modus Create LLC (2020) provides an example of a workflow similar to iRobot's Test team approach where Test creates issues and then dispositions them accordingly with tracking and automated workflows available within the tool. Jira provides collaboration, visibility, and alignment within teams to share information and gain awareness of how work and tasks are interconnected.

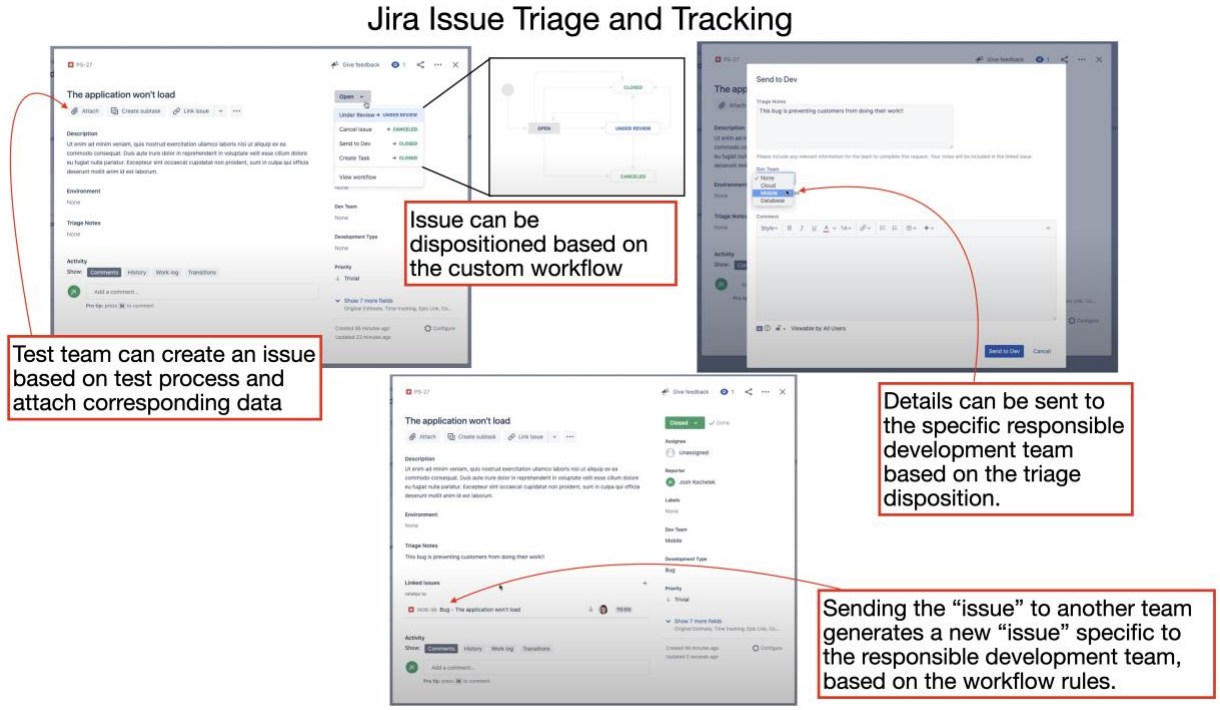

Figure 6: Jira issue triage and tracking with custom workflows between interconnected teams. The screenshots were sourced and edited from the Modus Create LLC (2020) video.

Similar to other cases with communication tools, a lot of what was considered "desk conversation" transitioned to Microsoft Teams conversations even when team members were co-located. The Microsoft Teams channels created for different groups enabled work groups to discuss "quick conversations" amongst their team without leaving their computer. Microsoft Teams provides flexibility and speed of iteration to the hardware team with instant access in a centralized space for communication.

5.2.3. Current Challenges and Ongoing Improvements

The Manager of Hardware Test noted that Jira is a “powerful tool” for “enabling companies in use for a lot of different workflows,” however, there is a “lack of clarity of usage and a lack of usage across multiple teams.” The pain point is that the tool requires its team members to follow a process with rules to adhere to Agile methodologies. The tool provides the features and visibility, but ideas such as “definition of done,” “required [details] to go into the stories,” and “where work comes from and where it goes,” “are things that the tool really can’t help with […] you need to know that.” Ultimately, while the tools enable the team to run Agile, entire organizations, not just single teams, still need to understand the Agile mindset and methodology to maximize the tools' effectiveness in multidisciplinary teams.

5.3. Ultraleap

5.3.1 Overview and Agile Implementation

Ultraleap is a consumer electronics company with distributed team members located in Silicon Valley (US) and Bristol (UK). Ultraleap focuses on hand-tracking vision systems for augmented and virtual reality (AR/VR) and automotive industries. Its Leap Motion Controller 2 allows users to utilize hand-tracking tools for computer applications.

The Hardware Team Lead we interviewed helped implement Agile practices within their hardware team at the onset of development for their recent Leap Motion Controller 2 product. Agile adoption was driven by various stakeholders' needs to “balance risk and investment,” reduce downstream risks by early engagement with manufacturing and quality teams, and leverage a methodology that values exploration and iteration for design teams (Mills, 2023a).

The Ultraleap tracking hardware team is primarily co-located with a corporate hybrid work setting. The team size is 6 – 7 members operating Scrum practices in 2-week sprints. The team removed phase-gates except for the initial project kick-start and created 6-week phases that culminated in a review of more significant project milestones with leaders and larger cross-functional teams. The Scrum team used backlogs, sprint planning, daily standups, sprint reviews, and retrospectives to run the Agile process.

The Ultraleap team utilized epics to track critical milestones, such as “Proof of Concept” builds and “first trial runs.” Additionally, these epics used goals based on Technology Readiness Levels (TRLs) to estimate project maturity.

The Hardware Team Lead used a Gantt chart to communicate critical dependencies, critical path scenarios, and overall milestones to set realistic expectations in project planning to react quickly to unknown risks (Mills, 2023b). “We did have a Gantt [chart] as well, but what I tried to instill with people is that it was not a plan in terms of a fixed thing. It is a tool we use for planning scenarios and likely outcomes, but everything within it is planned agilely in those sprints.” The

Gantt chart also helped partner with external manufacturing teams not leveraging Agile and with longer lead times.

5.3.2. Cloud-Based Tool Strategy

Ultraleap's use of cloud-based tools was important in their collaboration as they transitioned to Agile methodologies. Although they used most of the cloud-based tools before the Agile transformation, the tools enabled the team to utilize Agile methodologies successfully. A list of the top-used tools is below:

- Jira for workstream and project management
- Confluence for information documentation
- Slack for communication
- Miro for workstream and visual collaboration
- Microsoft Teams for video conferencing
- Microsoft 365 for documentation and high-level project management

The Hardware Team Lead found that sprint planning "actually works brilliantly on [Microsoft Teams], and to some extent, better than it does in a meeting room." They noticed that when team members are in a conference room together doing sprint planning, "they're not focused necessarily on the screen" or may not have a good view; however, when the project lead shares the Jira backlog and the Scrum board in a virtual conference call "everyone has a really clear view of all of the information." The virtual video conference calls enable the team to focus hybrid days in the office on hands-on hardware reviews and face-to-face interactions with team members that build team rapport instead of sprint planning, which they can effectively do remotely. Jira and Microsoft Teams enable the team to be collaborative and flexible in ways that elevate beyond an in-person experience.

Similarly, Miro boards are big for collaboration. The team uses Miro for all retrospectives, strategic planning meetings, and structured approaches to issue root-cause analysis. Templates on Miro make it easy, fast, and collaborative to set up team discussions. For example, during root cause analysis brainstorms, the team uses Fishbone diagrams to discuss all possible approaches to understand the problem (see Figure 7); Miro's API with Jira then automatically generates Jira tickets to input into the backlog (see Figure 8). Miro's features enable the team to collaborate and reduce overhead to improve the efficiency and speed of iteration in updating other connected tools.

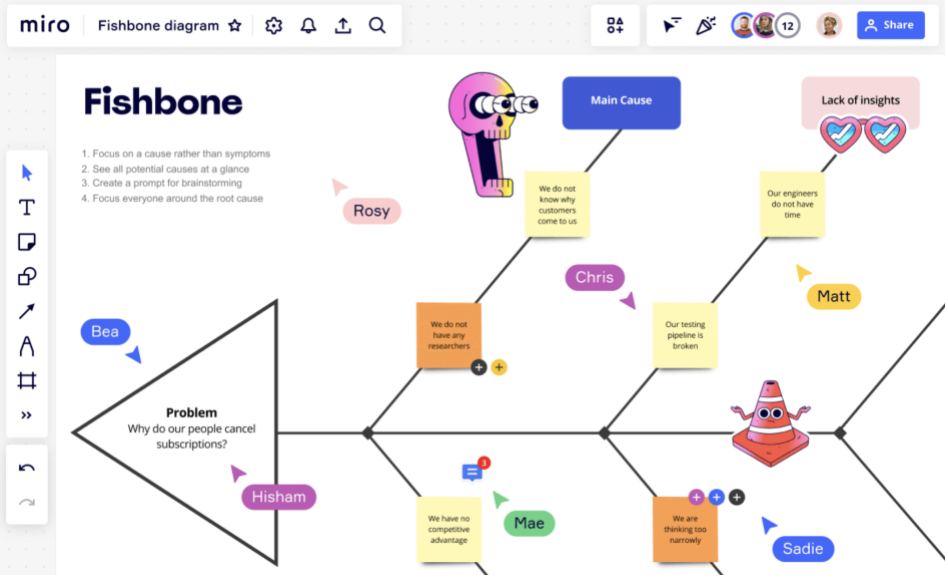

Figure 7: An example fishbone diagram using Miro for root-cause analysis of a problem or issue. Image source from Miro (n.d.b).

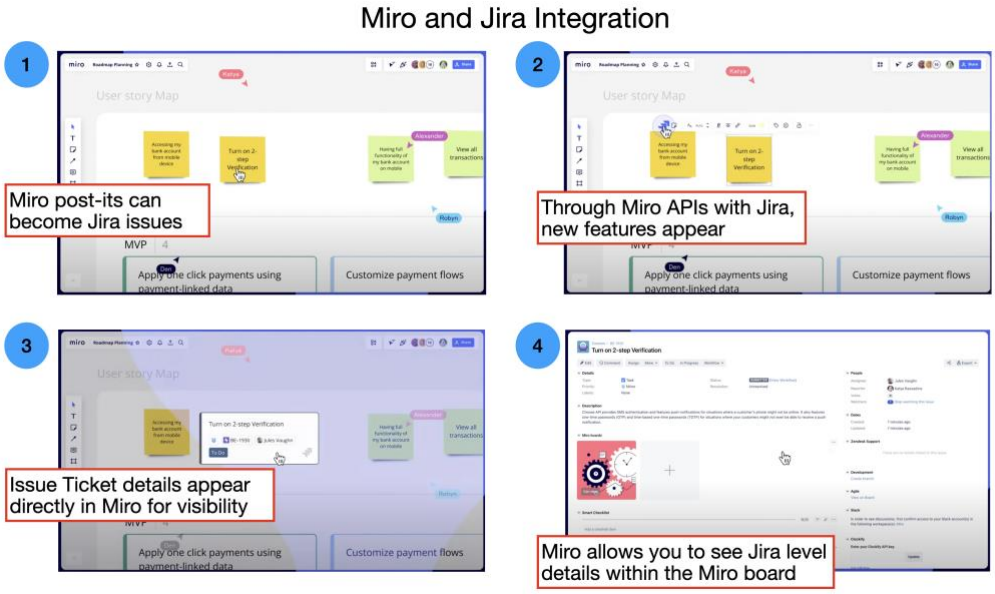

Figure 8: An example of Miro and Jira integration helps turn Miro items directly into Jira items using APIs. Steps 1-4 share the creation of Jira tickets through the Miro board. The screenshots were sourced and edited from the Miro (2022) video.

The team's cultural norms are strong on communication, embodying it in how they utilize Slack. The team does not wait for daily standups to communicate issues, and Slack can typically capture the issues and raise concerns for team support. The team is “always talking and always communicating,” which is critical for short 2-week sprints. In addition, they capture sprint reviews with a brief headline update in the Slack channel and a link to a Confluence page with all the review notes for a simple, clear message with the ability to dive in to learn more. The Slack and Confluence combination supports the team through collaboration, speed, and alignment.

5.3.3. Current Challenges and Ongoing Improvements

The constraints of the physical are a constant challenge for the team, especially in areas that are out of their control, such as external manufacturing. External manufacturers are looking for schedules and dates to understand when designs will freeze for their own planning, which is difficult if running Agile Scrum processes. This is the driver to utilize a Gantt chart to support external teams. Additionally, when issues arise in manufacturing that require team support, it may have to wait until the next sprint if there are other milestones. This creates a lot of "friction" between the team members and manufacturers depending on how closely Agile processes are adhered to.

The combination of conference rooms and virtual meetings has also created difficulties in collaborating and aligning aspects of hybrid work. The Hardware Team Lead suggests that conference rooms need to have "good mics and enough mics," "cameras with some intelligence that will focus on the presenter or talker," "easier to join and share content from your laptop," and better-placed TV monitors to have a productive and engaging meeting. They noted that a lot of recently released technology improves conference room setups for hybrid work. Without those improved setups, the team's ability to collaborate diminishes even if the cloud-based applications provide a virtual collaboration space.

5.4. Inertia

5.4.1 Overview and Agile Implementation

Inertia is a product design, development, and manufacturing company based in Toronto, Canada. Inertia serves a wide range of customers by providing a fully integrated product development experience.

We interviewed the President and Product Development Manager, who helped implement Agile methodologies and practices for the organization. Inertia is a distributed team, with a majority of its team in Toronto and a global client base. The team uses Scrum practices with 2-week sprints, standard sprint planning, daily standups, sprint reviews, and retrospectives. A typical Scrum team consists of 7 – 9 members across a multidisciplinary group of firmware, electrical, mechatronics, and mechanical engineers. Scrum team size and engineering discipline makeup vary depending on the project profile. Projects may also involve several Scrum teams and variable project lengths depending on the project's scope. Additionally, depending on the client's request, the team leads may host reviews with the client as frequently as once a week.

Serving a wide range of clients across a spectrum of services leads to managing various project plans tailored to the customer's needs. For larger projects, the project lead will create a multi-level project plan showcasing all expected sprints with interconnected work items, high-level project goals for each discipline, and forecasted sprint goals. These high-level goals and

activities ultimately drive the creation of low-level tasks managed through the daily Scrum process.

The entire team's participation and ownership throughout the sprint process empowers individual team members to discuss goals, tasks, and issues while aligning on expectations.

5.4.2. Cloud-Based Tool Strategy

At Inertia, the cloud-based tools play a fundamental role by being the hub that almost all information and communication pass through to drive project management. A list of the top-used tools is below:

- Miro for project planning and visual collaboration
- Jira for workstream tasks and project management
- Microsoft 365 suite for content creation and file management
- Microsoft Teams for communication
- SmartSheet for high-level project scoping

Miro has been “fundamental to the success of the project and getting buy-in from everyone.” In Inertia’s multi-level project planning described above, Miro is the hub for collaboration and alignment among the team members. The Product Development Manager explained:

> *“As the project gets more complex, the more I really need to see the tasks laid out in a visual format [in Miro] to show the dependencies in the flow. From there, I will export into Jira and track tasks, but we always come back to the Miro board to get our grounding [to answer questions like] where are we in the context of the bigger project? The task that you are looking at right now, what follows it? What did it follow?”*

Team accessibility to the board gives the team ownership and drives cohesion throughout the project. Figure 9 provides an example of the capability to leverage Miro to visually connect items across teams (rows) and sprints (columns) for team alignment.

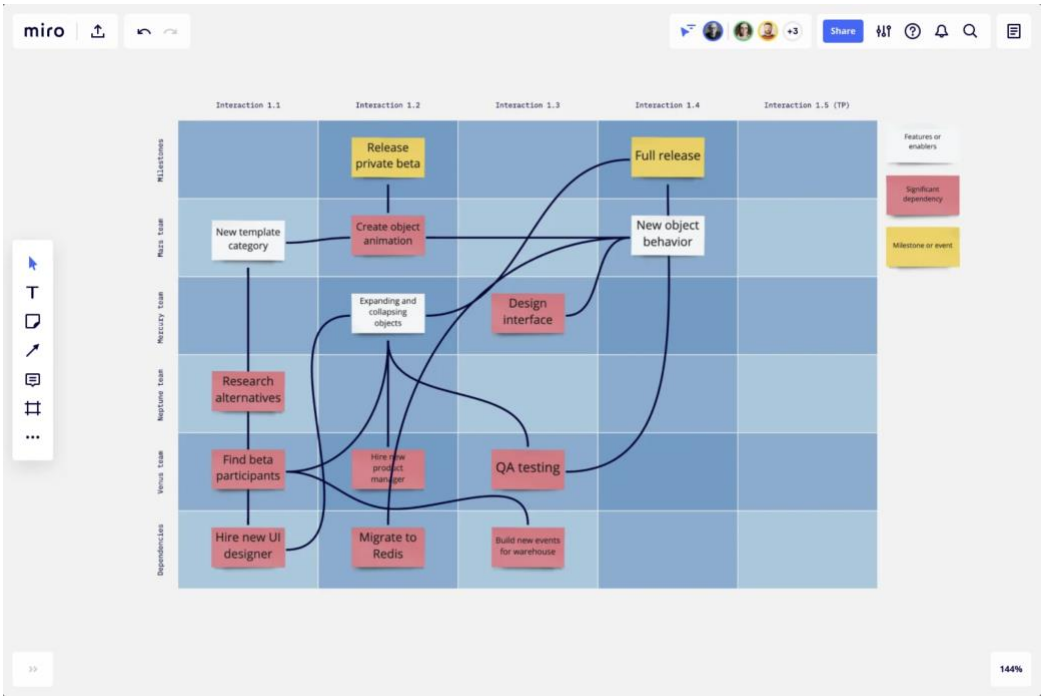

Figure 9: An example of a Miro board used to project plan across multiple disciplines (rows) and sequential sprints (columns) to track dependencies and visually align. Image source from Miro (n.d.a).

Miro also serves as their visual presentation to clients. The team curates Miro with individual snapshots of the trackers and technical content to serve as visual aids when sharing updates with clients. Miro's infinite whiteboard canvas then enables these artifacts to be laid out side by side to quickly and visually track the historical progress shared. This provides the team speed in identifying content to discuss and alignment with client expectations.

Similar to other organizations interviewed, Inertia also found that with distributed and hybrid teams, the experience of a video conference call is "a much better experience for everyone" than having a hybrid meeting of some team members in a conference room and others joining remotely. Inertia established a practice of requesting teams to use virtual video calls if one or more team members are unavailable in person. Although in-person is preferred, when it is not possible, cloud-based video communication tools enable teams the flexibility to continue to collaborate.

5.4.3. Current Challenges and Ongoing Improvements

Inertia's current pain point is the use of many disparate cloud-based tools. They have a "Gantt style tool [for project scoping], an infinite whiteboard with Miro, separate PowerPoint deck trackers, design files in SolidWorks, Jira, Teams, and so many others." This use of tools has generated "tension" with project leaders, but they've struggled to find an efficient solution. They mention that "one-stop shop" tools are "never perfect" and typically lack a specific feature that leads the team to integrate another tool. The Product Development Manager describes

using multiple tools as an “art” where they are constantly “figuring out what things to lean into and which things to pull back from.”

5.5. Dell Technologies

5.5.1 Overview and Agile Implementation

Dell is a large multinational technology company known for products such as personal computers, servers, computer accessories, and security technology. The Experience Innovation Group (EIG) within Dell's global team focuses on modernizing, transforming, and reimagining customer experiences.

The Principal Product Development Engineer and Principal System Architect interviewed are part of the EIG team focused on new innovative products. While the New Product Introduction (NPI) team leads Dell’s mass-produced products, the EIG team focuses on showcasing products internally and externally to understand the feasibility of new technologies and use cases.

The EIG team consists of a multidisciplinary distributed group with a team size of roughly 10 people, meeting twice a week. The EIG team uses Agile Kanban methodologies to drive their processes as “traditional waterfall was not workable for innovation.” When developing a new product or experience, they will “set up several milestones, separate the new product out into four or five parts […], and each of those parts will become their own project with teams working in parallel to prove the function.” Each subset team has a short schedule to explore various options. The teams are “not focused on time; [they] are focused on technology maturity.” Although they do not tie themselves to timeboxed sprints, they typically spend 3 to 4 weeks researching various technologies. They constantly switch and run parallel discussions with multiple contract manufacturers to understand feasibility and maturity.

The team uses a Kanban board to move work items through their research workflow. Because the team works with many suppliers on novel technologies, they are regularly subject to the specific supplier's response time, which makes Kanban methodologies helpful for visualizing how the various options are progressing in their project.

A few months ahead of significant business events, such as the Consumer Electronics Show (CES) or internal events, they use more rigid project planning principles to create team integration points. Teams will typically leverage their Kanban board findings and “cherry-pick the top features” to make the system-level integration milestones by “back-casting” from the “time-to-experience” that best fits the organization's goals.

The Kanban process enables the team to work around their supplier's schedules while giving them the flow and transparency to track the progress of many technology options before focusing on more integration-based execution.

5.5.2. Cloud-Based Tool Strategy

Dell's distributed team leverages cloud-based tools to collaborate and align on critical points. A list of the top-used tools is below:

- Jira for workstream and project management
- Microsoft Teams for communication
- Microsoft SharePoint and 365 for documentation
- Miro for visual collaboration
- Zoom for video communication

They use Jira and Microsoft 365 products daily to create content and track updates. Since the EIG group is globally distributed, they have the added burden of working in different time zones that don't overlap during regular working hours. The team uses the comments and direct tags within these tools and documents to provide feedback and questions as they work asynchronously throughout their days. "Now that the files are online, the document sends you a notification […] that you need to update something." The Principal System Architect notes that this "really speeds up the work" and avoids keeping people up for a conference call outside work hours. Figure 10 is a representative example of how Dell utilizes PowerPoint and Microsoft 365's real-time collaboration to edit and comment on a single file to improve their communication and speed of completing documentation.

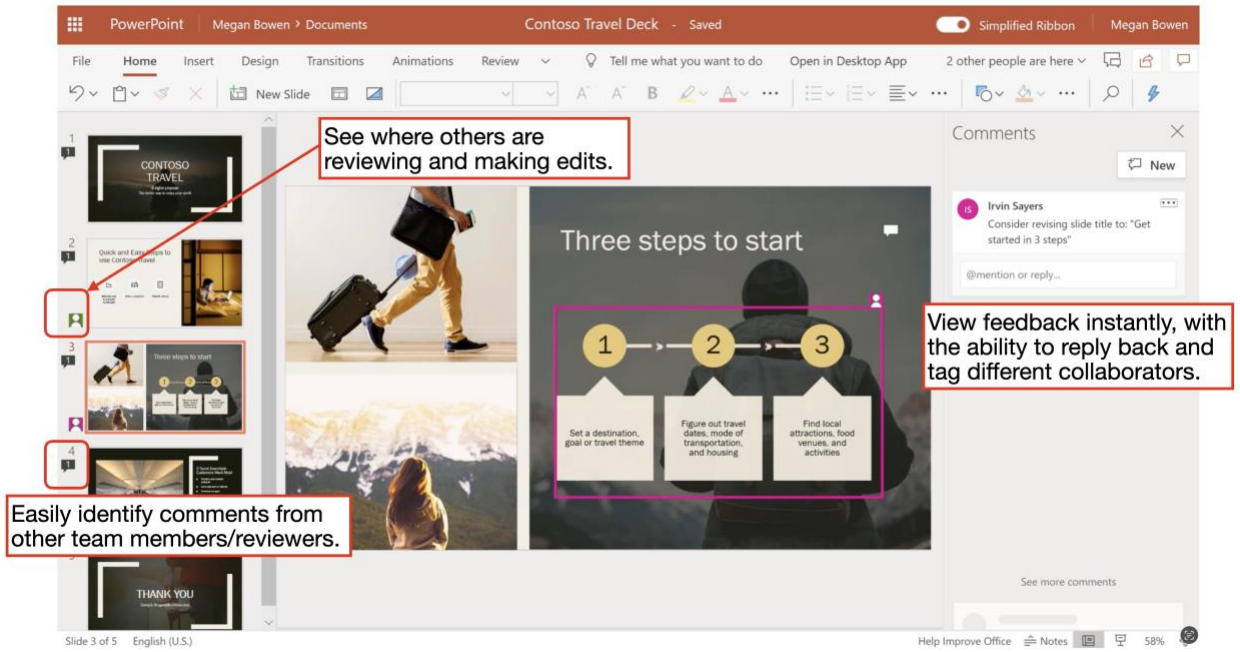

Figure 10: An example of Microsoft PowerPoint's real-time collaboration with multiple users simultaneously editing, reviewing, and commenting in a live file. The screenshot was sourced and edited from the Microsoft Support (n.d.) video.

Dell EIG uses PowerPoint for a lot of presentations between members to discuss their findings and share technical information with the larger group; however, when they "want to discuss about a user experience, [they] will use Miro board" by setting up a Zoom and having their team join in a brainstorming discussion. The team uses a lot of Miro boards to strategize more significant questions. The discussions help them to overcome barriers with their team regardless of location.

These added levels of communication enhance the team's ability to collaborate with the flexibility of both synchronous and asynchronous speed.

5.5.3. Current Challenges and Ongoing Improvements

Dell has benefitted from cloud-based tools by enabling their teams to continue to be productive and collaborative regardless of location. However, one of the issues mentioned by those interviewed was the necessity to have high bandwidth internet speed. Both interviewees spend significant time at manufacturers' locations, understanding their technology and process as they researched various solutions. Because of this, they are vulnerable to low internet speeds and the need to VPN into their corporate networks to access corporate credentials. This causes many tools to work poorly, slow the process, and cause frustration. The need to always be connected leads to issues with productivity in these situations.

5.6. Symbotic

5.6.1 Overview and Agile Implementation

Symbotic is a robotic automation company that develops systems that improve the operating efficiencies of warehouses with headquarters in the Greater Boston area. Symbotic was founded in 2007 and, similar to iRobot, develops software and hardware for its robotic systems.

The Manager of Robotics Test Engineering interviewed is part of a highly integrated team working with both the software and hardware development teams to test and validate the systems.

The project team uses the Kanban process with a team size of 8 – 15 members that span mechanical, electrical, and test engineers. The teams are co-located in the Greater Boston area but work across multiple buildings in the corporate campus. The project lead hosts standups 3-days a week.

The Robotics Test Manager highlights that Kanban's advantage at Symbotic, specifically with hardware, is the ability to move activities (work items) into different states. For example, they can move an activity into the state of "material pending," estimate when that item will arrive, and go to work on another activity. Because the hardware teams rely on external vendors, the ability to "place activities on the shelf" instead of waiting allows them to move on to other activities in the backlog that are not blocked. This keeps the team members productive and

activities flowing across the Kanban board. It is one way to mitigate the challenges of the physical constraints of working with hardware in an Agile way.

5.6.2. Cloud-Based Tool Strategy

Symbotic's teams use cloud-based tools to collaborate and provide visibility and alignment to all team members. A list of the top-used tools is below:

- Jira for workstream and project management
- Microsoft SharePoint for documentation
- Microsoft Teams for communication
- Miro for whiteboarding

Jira is a critical project management tool to align team activities and work. Along with providing a visual flow of the Kanban board for a team manager, Jira also provides the additional benefit of tracking the work across multiple projects. By establishing specific templates and criteria to populate various fields in Jira, the Robotics Test Manager can create a Jira query to gather data specific to their team and automatically generate dashboards based on available data. Figure 11 shows an example of how this may be portrayed with various Jira gadgets and information to monitor and track different forms of information. This level of automation, using customizable fields, creates a powerful tool that provides the manager visibility, alignment, and speed to monitor and iterate team productivity.

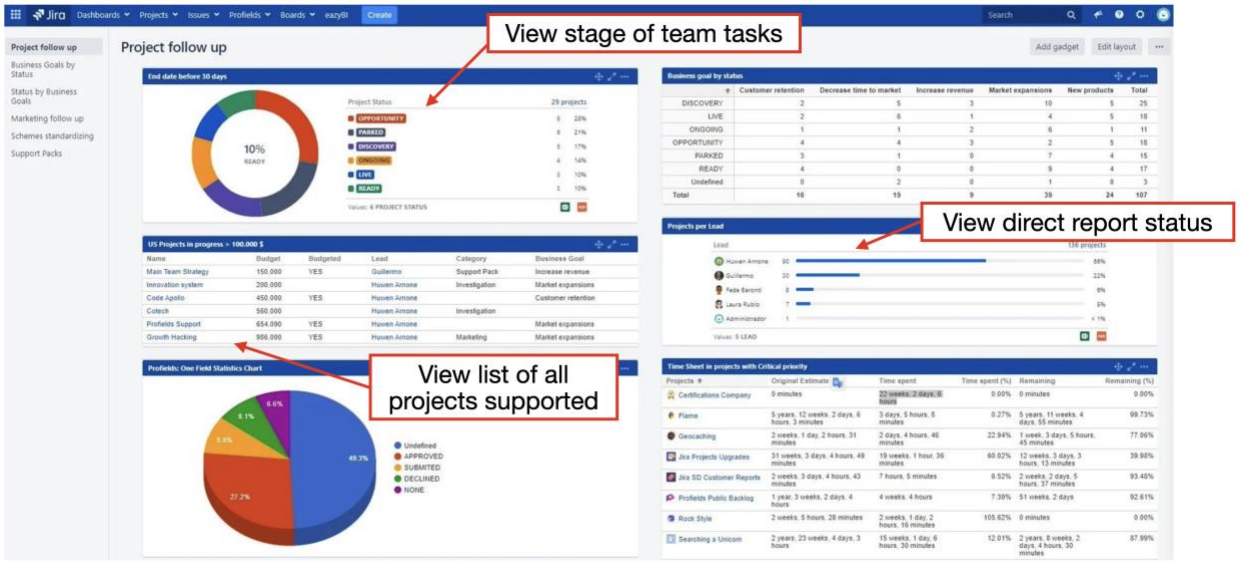

Figure 11: An example of a Jira dashboard pulling information from multiple projects. Multiple views and gadgets provide data in various forms for speed and flexibility. Image source edited from Diaz (2023).

5.6.3. Current Challenges and Ongoing Improvements

Although cloud-based tools have enabled increased levels of collaboration, the Robotics Test Manager mentions that "whiteboard is super hard to replicate virtually for brainstorming." They explain that "in-person it can be a 30-minute brainstorm vs. 1-hour brainstorm over video" due to the difference in remote engagement within the team.

> *"The [physical] room is a good control of the conversation, and every time I leave an in-person whiteboard session, people want to take a picture of the whiteboard because they can finally see what they had been trying to visualize. It was the team effort and the discussion that caused it. People like to see how they got there."*

This challenge contradicts what other interviewees have shared on brainstorming via cloud-based tools such as Miro. Different practitioners have leveraged Miro effectively as a collaborative virtual tool. This example brings to light that the tool can empower its users but may require more training and team adaptations to realize the benefits.

5.7. Volta Labs

5.7.1 Overview and Agile Implementation

Volta Labs is a biotechnology company developing scalable automation for genetic engineering in the Greater Boston area. As a new company founded in 2018, the project management team has had to create project management procedures to stay organized and keep projects on track as the company and project expectations grow.

The project teams are structured into three tightly coupled groups: hardware, software, and sciences. We interviewed the Technical Program Manager (TPM) leading the hardware team.

Volta Labs uses an Agile process, predominantly Kanban, but with aspects of Scrum across hardware and science teams, while the software team uses the more traditional Scrum approach. They did not share the exact team size, but all team members are co-located in the Greater Boston area and use activities that follow Sprint Planning, Daily Standup, and Sprint Review processes, although they do not use that exact naming. The team discusses and reviews goals for their "sprints" that tie into their Kanban work items. The team does not use story points or time allocation for their work items.

The TPMs of the different groups – hardware, software, and science – meet weekly to discuss interdependencies and issues between the teams. The TPMs also partner to do higher-level project planning to tie in long lead time items and critical milestones that are then taken and driven through their Kanban/Scrum hybrid process.

Most teams are in-person since a significant amount of work deals with physical systems and laboratory work that is only available on-site.

The unique Agile process allows the team to be flexible to change as variability in scientific work is difficult to timebox; however, leveraging certain practices and mindsets enables the team to iterate and progress at their current maturity.

5.7.2. Cloud-Based Tool Strategy

Volta Labs' cloud-based tools enable them to collaborate by providing centralized locations for documents and flexibility to work from anywhere and anytime. A list of the top-used tools is below:

- SmartSheet for project management
- Jira for workstream and issue tracking
- Google Meet for video communication
- Slack for communication
- Google Docs for documentation
- Figma for visual collaboration

They use SmartSheet as the primary high-level project planning tool. It tracks the dependencies between different groups and serves as the primary tool that TPMs utilize to "figure out what the hard deadlines are and work backward to see [where to plan it]." The SmartSheet stays high-level and is not used to capture the granularity of the daily tasks discussed in daily standups. Figure 12 provides an example of the capability of SmartSheet to collaborate with multiple users for high-level planning.

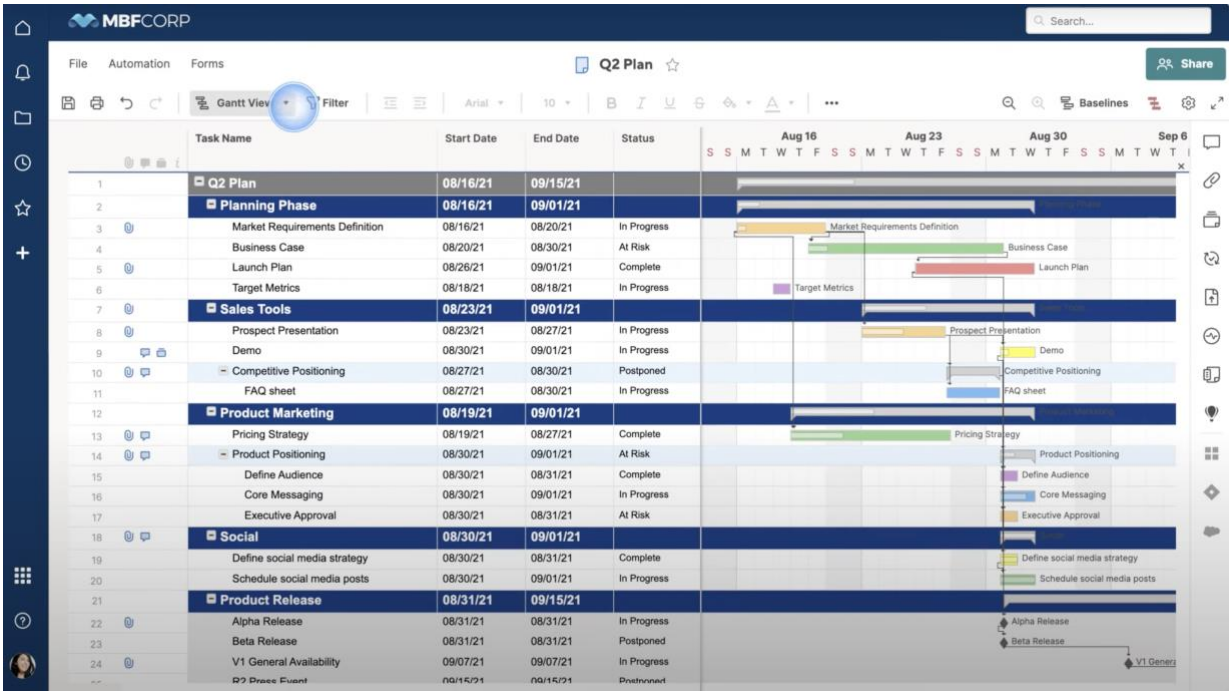

Figure 12: An example of a SmartSheet Gantt chart with collaborative capability for overall project planning and dependency mapping. The screenshot was sourced from the SmartSheet (n.d.) video.

They use Jira as a critical tool for collaboration between teams. All teams utilize Jira for issue tracking and reporting, which is essential to improve the speed of information sharing and notification of issues. For example, suppose the science team or the hardware team finds an issue. In that case, they generate a ticket in Jira, tag the respective team members supporting that issue, and begin discussing solutions within the tool. They raise the issues in the daily standups and weekly sprint reviews to find a resolution.

Additionally, using Jira and Google Sheets as real-time collaborative tools enables team members to work asynchronously. The TPM mentions that “it's super great doing async conversations; if I need quick answers, it’s not as helpful [and I’ll use Slack], but if I’m reviewing something on my own time and the person can get back to me within the afternoon or within the next day or two, then the comments have worked great.” The tool allows the team flexibility to work and manage their time in specific workstreams.

5.7.3. Current Challenges and Ongoing Improvements

The major challenge the TPM experiences is managing the project plan timeline with the existing SmartSheets tool. When making changes to re-baseline the timeline because of new dependencies or new findings that adjust the length of existing activities, the SmartSheet fails to acknowledge the new information. In working on a new product platform, the team constantly discovers new findings where “the scope of the task has to change because we realized something [new] as we’re developing, designing, or building that is actually more complicated.” These issues have led to frustrations and inefficient workarounds like cloning

documents to make new edits to preserve historical information. The TPM requires a more robust tool with additional flexibility for high-level scheduling, similar to the weekly task tracking offered by tools such as Jira.

5.8. Root AI

5.8.1 Overview and Agile Implementation

Root AI was an agricultural robotic company focused on robotics for indoor farming in the Greater Boston area. Its Virgo harvesting robot enables the automated picking of multiple crops. The company was founded in 2018 and acquired by AppHarvest in 2021. The interview includes a team member who previously worked at Root AI but transitioned out of the company prior to its acquisition.

We interviewed the Director of Hardware Development, who was an early member of the organization. From the onset of development, they implemented Agile Scrum practices to integrate the software and hardware of the system.

The small organization was co-located in the Greater Boston area. The 5 – 7 person teams were split between hardware and software and organized in a Scrum process with 2-week sprints. Although teams were co-located, the organization operated in a hybrid structure of in-office and remote work on certain days of the week.

Critical to success was the collaboration between software and hardware. During the project planning stage, which would continuously happen throughout the project as new findings emerged, the team leaders would work on making the project “a little bit more modular” to plan integration points better and answer questions such as “when should we expect to have features,” and “when should we expect to need software support for certain features,” “coordinating back and forth on what the [sic] the features were and what the interfaces are.” Collaboration and coordination were critical to align groups in an ambiguous new platform and new product for a new organization.

5.8.2. Cloud-Based Tool Strategy

Root AI’s use of cloud-based tools was important in their collaboration and use of Agile methodologies. A list of the top-used tools is below:

- Jira for workstream tasks and project management
- PointingPoker.com for team sizing and story point allocation
- Confluence for information documentation
- Zoom for video communication
- Slack for text communication

Root AI's first attempt at task tracking was using Trello. However, the team ran into issues as project details and tasks evolved that required "more horsepower" because of "the need to recategorize [tasks] and put [them] into epics." Root AI transitioned to Jira for its "great power of bulk change" and its "power to manage thousands of tickets to figure out what they all are and why they matter." Jira gave them a tool with the speed to match the level of learning and iterations that Agile offered to their product development process.

Root AI also adopted Confluence to document information and learnings. Confluence is the central hub for information outside of direct tasks that are heavily relied on. In Confluence, "everything feels alive," whereas in "Google Documents, it's like somebody else wrote it, so I [feel the need] to have to clone it before I go and add to it." The "version history is great" and "every document is a living document; […] if you want to tear up the theory of operation of the whole machine because you are doing a big revamp of everything, you can do that!" Confluence gave the team the ease of adaptiveness to make changes without feeling unable to revert or pivot. Figure 13 shows an example of the collaboration capabilities of Confluence to identify teams, edit tasks, and directly edit and comment on content.

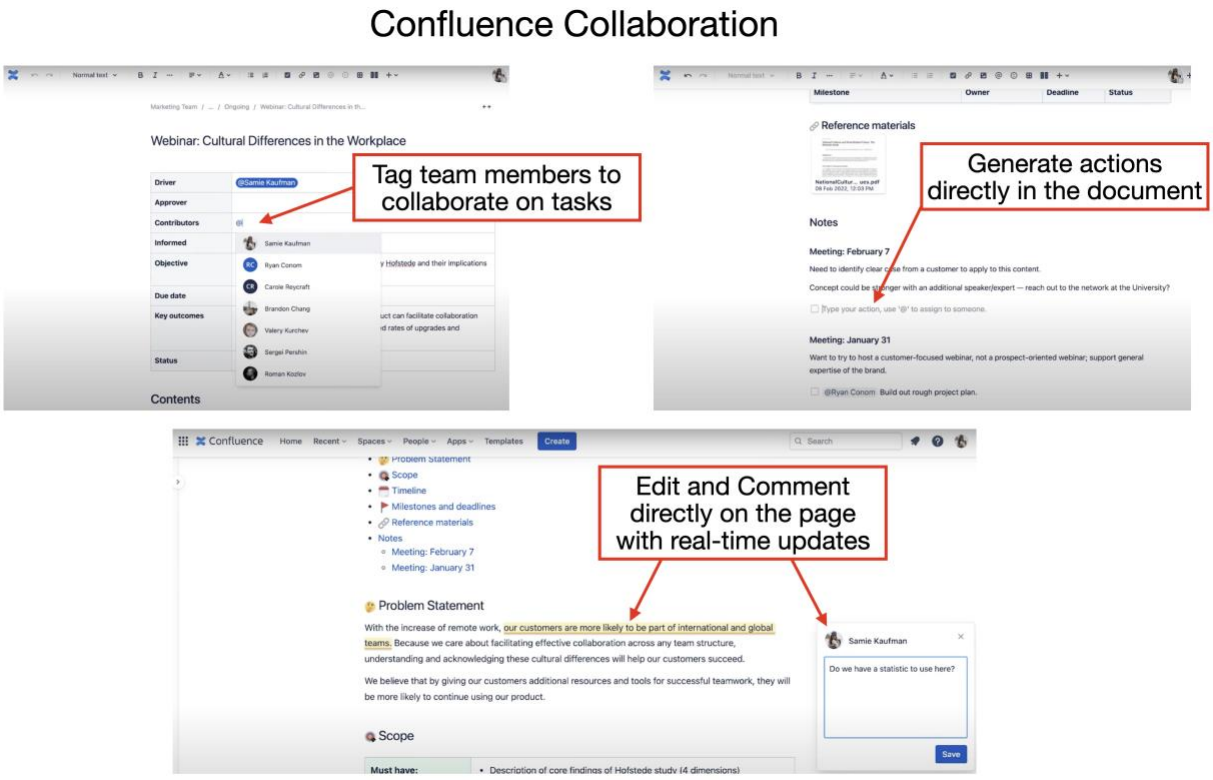

Figure 13: An example of a Confluence space used for collaboration and editing. The screenshots were sourced and edited from the Gliffy by Perforce (2022) video.

Regarding sprint planning, the Director mentioned that the "most important tool" was pointingpoker.com. PointingPoker.com is an online tool that determines the story points to apply to tasks as Scrum teams plan sprints and manage their backlog of work. It follows the game of "Planning Poker," or "Scrum Poker," where team members collaborate to assign points

to tasks using the Fibonacci sequence to determine story point complexity for improvements on sizing. With a new organization, new team members, and a new product, PointingPoker.com helps provide the Scrum team with an easy and collaborative way to align on task estimation to support their Agile methods. Integrating this activity into Jira could be beneficial in gathering data for improved estimation and planning.

5.8.3. Current Challenges and Ongoing Improvements

As has been a trend through this research, Root AI team members would need to review tasks in the backlog when planning and "discuss the definition of done." The pain point focused on the design and release of components: "You always have to describe what done means. You always have to ask, 'Are we going to just finish the CAD, or are we going to finish the CAD and finish the drawing and finish ordering it?'" Modularizing was helpful, but with 2-week sprints, it would take "10-12 weeks of using it […] to really get velocity to a meaningful number." Even an experienced practitioner will experience learning curves with new platforms and products in an Agile hardware environment.

One of the tools beneficial to the Director's design reviews but not native to the SolidWorks CAD program was Five Flute (www.fiveflute.com). Five Flute enables users to share a CAD model with internal and external teams and add markup comments and feedback directly on the model, as shown in Figure 14. Additionally, it enables interactive presentations while avoiding the need for screenshots in slide deck presentations. Users can use the 3D model directly while retaining all comments in a presentation mode. This was particularly helpful since "design reviews always devolved into people just spinning the CAD; having it start [in that state] in the first place" made preparation and reviews easier. This style of cloud-based tool allowed for additional collaboration and speed with the team to better communicate with a visual backdrop. Although some CAD programs may allow comments, the on-model and presentation mode features enabled higher levels of collaboration and alignment with the team.

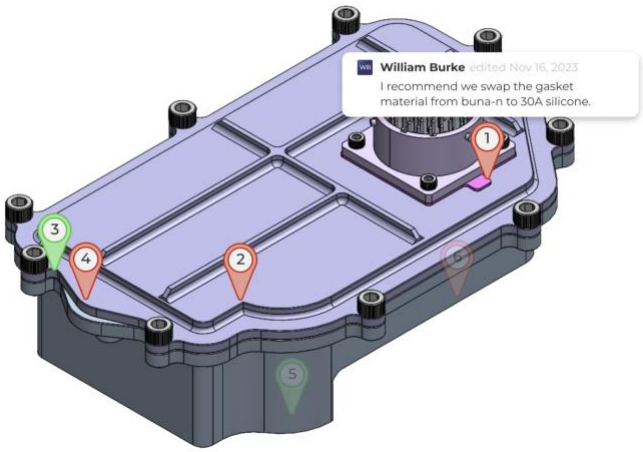

Figure 14: An example of a Five-Flute CAD design review with comments directly on the model. Image from Five Flute (n.d.).

5.9. Organizational Adoption of Cloud-Based Tools

The analysis of the organizations in the case studies reveals a diverse adoption of cloud-based tools for Agile hardware development. Across the organizations, they have employed various tools to facilitate collaboration, enhance speed and flexibility, and ensure alignment within Agile teams. Although teams may use additional tools to meet work expectations, the interviewees identified the tools mentioned as critical to their collaborative project management methodologies. Figure 15 illustrates the distribution of tool adoption across organizations and coupled within the category of tools previously identified: communication tools, workspace and lifecycle management tools, and design and visual collaboration tools.

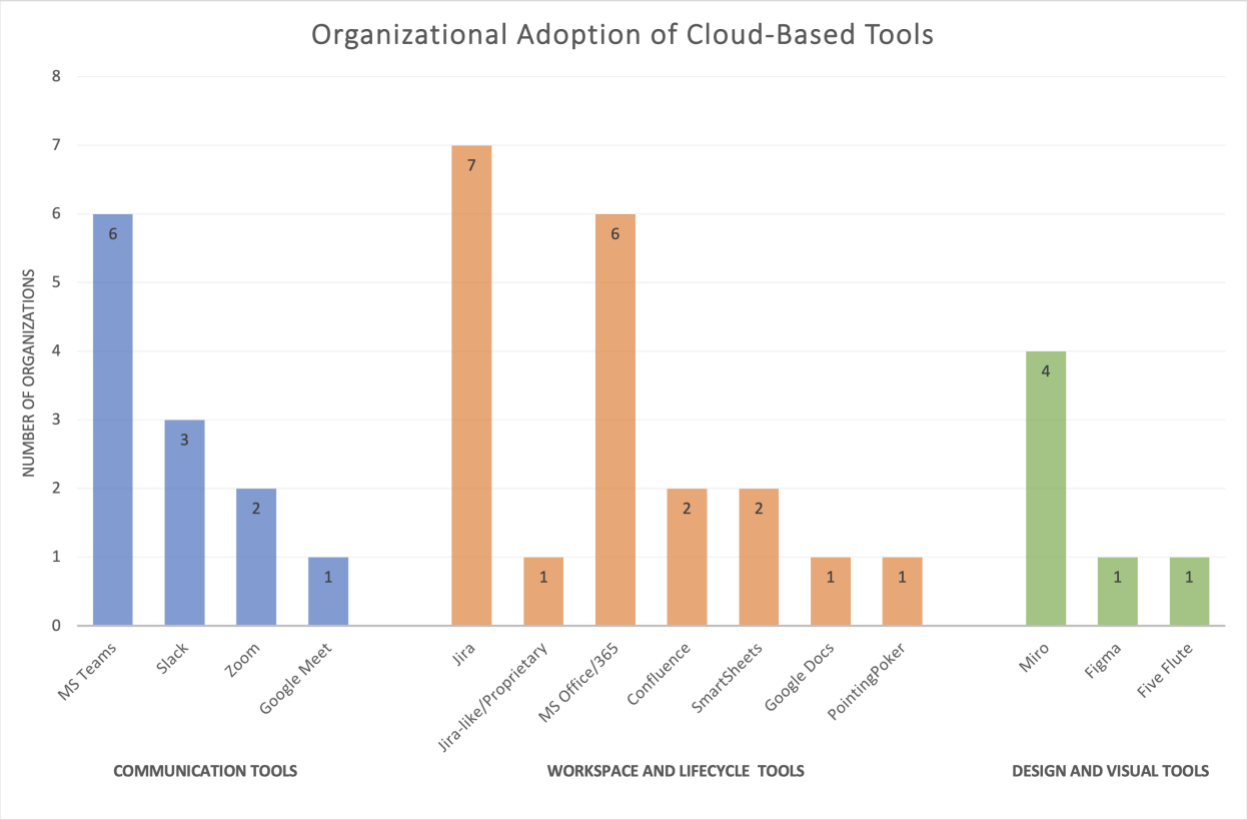

Figure 15: Distribution of cloud-based tool adoption by case study organizations.

Our research captures only a small subset of hardware organizations using Agile; however, the diversity of tools underscores the adaptability of cloud-based tool ecosystems in catering to teams' various requirements. The high utilization of Microsoft products (i.e., MS Teams and MS Office/365) highlights the ecosystem's importance in seamlessly connecting applications.

In the next section, we propose a framework to capture our case study analysis results.

6. Theoretical Framework and Discussion

6.1. Theoretical Framework Overview

Hardware development organizations are embracing and adopting Agile methodologies to manage their projects in an iterative and collaborative nature. By exploring existing literature and in-depth case studies, we've delved into the intricacies of Agile methodologies, hardware development processes, and the integration of cloud-based tools. In response to these insights, we propose the following framework to capture the ways in which cloud-based tools support four enablers of Agile for hardware development teams: Collaboration, Speed of Iteration, Flexibility, and Alignment. See Figure 16 for details.

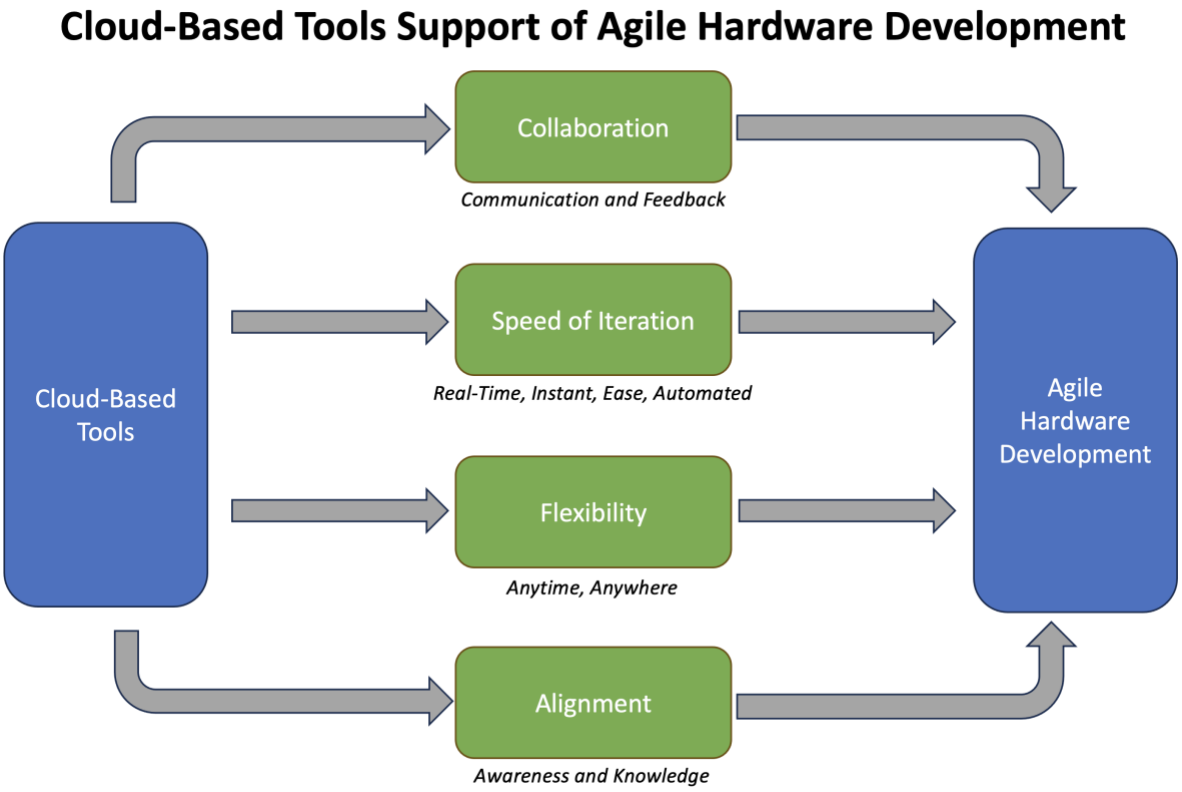

Figure 16: The framework for cloud-based tools supporting enablers of Agile methodologies.

The framework suggests that cloud-based tools, including Jira, Microsoft 365, Miro, Zoom, and others, enable enhanced collaboration, iteration speed, flexibility, and alignment – all fundamental principles of Agile methodologies.

The following subsections will define and discuss each aspect of the framework.

6.1.1. Collaboration

In our framework, we define collaboration as the partnership with others through communication and feedback loops. Collaboration is critical in Agile methodologies; cloud-based tools that enable collaboration through their features and interface embody Agile and allow the team members to be Agile.

Throughout the interviews with Agile hardware practitioners, the commonality between all cloud-based tools was “collaboration.” By design, the tools enable multiple users (team members) to come together to communicate, discuss, share, design, and build together in the virtual spaces. The tools are no longer native to a single person's workstation but shareable with others to elicit feedback and development.

Teams experience collaboration through the “quick conversations” on Slack and Teams, the “virtual” daily standup meetings on Zoom and Teams, and the “brainstorming” sessions on Miro.

Collaboration is seen throughout the case study examples, and, in particular, iRobot’s use of Jira as a central hub for integration and issue resolution showcases Jira’s ability to enable collaboration. The iRobot hardware test team generates tasks based on the latest testing, describes the issue, shares test data, and directs it to the appropriate team for follow-up and feedback. The project teams can then comment, direct message, or share more documentation to keep a historical reference of all communication and feedback within Jira for easy search and retrieval.

6.2.2. Speed of Iteration

In our framework, we define speed of iteration as the ability to operate quickly by sharing information, automating tasks, or ease of completing activities. Speed enables the team to iterate faster, move quickly, and respond to changes more effectively, which are all benefits that Agile methodologies offer.

Throughout the case studies, we highlighted numerous examples of iteration speed brought by using cloud-based tools. For example, Ultraleap's use of Miro boards for their root-cause analysis with customized fishbone templates automated the transition between brainstorming solutions to task assignment within Jira, with built-in APIs. This level of automation and connection between the cloud-based tools enables an easy transition along the workflow of root-cause analysis, leading to faster iterations to close knowledge gaps.

Similarly, Root AI would spend time curating images for a PowerPoint presentation of the CAD for design reviews. However, utilizing the cloud-based Five Flute tool that allows for comments and highlights within a CAD model simplified the process and enriched the design review experience without the waste of creating numerous PowerPoint slides. The speed at which they

could iterate by reducing the team's burden preparing for reviews enabled them to communicate clearly with the team while reducing effort.

6.1.3. Flexibility

In our framework, we define flexibility as the ability to access information and people anywhere, anytime, to improve productivity. Flexibility supports the collaboration of distributed teams and workplace impacts caused by COVID-19. Flexibility enables teams to stay connected and engaged when responding to change on the go.

We see flexibility throughout the case studies, with teams that are both globally distributed or have hybrid workforces that require Zoom and Slack for direct calls and messaging and access to Jira, Confluence, and Microsoft 365 to collaborate and create content.

Specifically, both Dell, a globally distributed team, and Volta Labs, a hybrid co-located team, emphasize the benefits of asynchronous work to complete document and design reviews when other team members are unavailable. Using in-text comments and tagging individuals elicited a flexible and collaborative approach to work.

Along those same lines, Siemens, a globally distributed team, leverages Microsoft Teams to have synchronous meetings for daily standups. At the same time, their cultural communication norms make it easy to direct message team members or groups to remove barriers.

6.1.4. Alignment

In our framework, we define alignment as a team member's awareness and knowledge of the project goals and status. Alignment is essential to Agile, as teams operate in fast iterations and can quickly respond to change. Alignment enables teams to move synchronously towards common goals.

Alignment is seen throughout the case studies, as project managers leverage cloud-based tools to visually capture the intricacies of the projects, with dashboards, Scrum boards, and Kanban boards to align priorities and actions.

Inertia's Product Development Manager creates multi-level project plans to visually capture the goals, dependencies, and activities the team will need to execute. The team's active participation in revising, updating, and editing information provides ownership of project goals for all team members and builds cohesion and confidence in the path to project completion. Team members align as they collaborate and work together.

Similarly, the Robotics Test Manager at Symbotic can leverage ongoing project updates in Jira across the entire organization to build queries and dashboards specific to the test team's own activities to align easily, even though the test team supports numerous ongoing projects. Jira's

capability of pulling tasks and data across multiple projects provides visibility and awareness that would be difficult to manage if a manager had to join meetings across all the supported projects.

7. Conclusions and Recommendations

7.1. Conclusions

Our research presents the methods and practices that hardware teams use to implement Agile processes and proposes a framework to explain how cloud-based tools enable and support those same Agile methodologies. Through literature review and interviewed case studies, we learned how hardware teams adopt Agile to run their projects and how cloud-based tools tie into their workflows.

We provided background knowledge on various project management forms, from Waterfall phase-gate processes to the differences between the two dominant Agile practices in hardware, Scrum and Kanban. In the literature review of studies on Agile hardware groups, we identified that there is no “one size fits all” approach to applying Agile to hardware, but instead, there are adaptations needed within the traditional Agile software development to accommodate the aspects of working with physical systems. We learned that the three major challenges impacting Agile adoption in hardware are constraints of the physical, learning cadence, and backlog or task creation. Researchers provide several strategies to help resolve these challenges and successfully implement Agile. The main strategies are the following: prioritizing high-uncertainty, high-risk items at the start of the project to answer the most critical knowledge gaps of the product; modularizing physical products down to smaller increment tasks to feed faster iterations; and leveraging multilevel plans to introduce phases with higher level project goals and integration points decomposed to lower-level individual tasks.

We also provided background knowledge on cloud-based tools and separated them into communication, workspace and lifecycle management, and design and visual collaboration categories. In our literature review, we discussed how cloud-based tools have impacted Agile software development teams. We discussed how tools provide software teams with situational awareness to enable them to be Agile. We also discussed how the onset of COVID-19 propagated the higher use of cloud-based tools and their increased features and capabilities.

Our interviews and case studies of nine individuals across eight organizations utilizing Agile hardware development methodologies gave us insight into how organizations structure their Agile teams, what suite of cloud-based tools they use, and a spotlight of specific cases of cloud-based tools enabling Agile practices. The interviews led us to create a framework describing how cloud-based tools support Agile hardware development by enabling enhanced collaboration, accelerated iteration speed, flexibility, and alignment. Collaboration arises seamlessly as team members communicate and exchange feedback through facilitated channels

within these tools. The speed of iteration is driven by real-time collaboration and the immediate accessibility of connected information within the tools. Flexibility arises from the cloud-based nature of tools, granting users accessibility anytime, anywhere. Alignment is achieved through enhanced visibility and knowledge-sharing facilitated by these tools. These characteristics tie into critical Agile methodologies that are the cornerstone to successfully adopting Agile to reap its benefits of collaboration, iteration, and responsiveness to change.

Although literature focused on cloud-based tools predominantly discusses only their use for Agile software development (Calefato & Ebert, 2019; Mancl & Fraser, 2020), we've shown that these tools can be leveraged by Agile hardware development, as well as enrich the development process.

7.2. Recommendations for Practitioners

7.2.1. First-Time Agile Adopters

For practitioners planning to transition to Agile methodologies from Waterfall-type methodologies, they should consider the overall structure of projects based on their complexity and risk level. Within an organization, it may be favorable to leverage full Scrum, full Waterfall, full Kanban, or a hybrid of approaches for different aspects of the project.

In the literature, we found that for low-uncertainty, low-risk projects, Waterfall was a preferred method since it followed a structured plan that most teams could follow. However, once uncertainty arises, Agile methodologies become useful for effective project management. Along these same lines, Siemens used textbook-style Agile Scrum methodologies during early feasibility stages when unknowns and uncertainties were high; however, they transitioned to Waterfall phase-gate approaches towards the end of the project when the risks stabilized, and quality in a highly regulated industry became crucial.

There is no "one size fits all" approach to Agile; however, if you acknowledge the project profiles and adequately train your workforce on the proper Agile methodologies, your organization can benefit from collaboration, iteration, and response to change in its project management.

7.2.2. Agile Users Leveraging Cloud-Based Tools

For practitioners who are currently utilizing Agile and are evaluating cloud-based tool adoption to support processes, they should consider their existing workflows. In our case studies, cloud-based tools worked effectively when structured workflows existed in the organizations and teams they served. Practitioners should understand the tool's capabilities in successfully supporting that workflow. Many cloud-based tools have a large set of features and capabilities that span multiple workflows. The specific workflow that a tool will be used for should be clear to set the appropriate rules of engagement for team alignment and awareness.

7.3. Recommendations for Future Research

Our research and case studies were not extensive and additional information is needed to fully understand the impacts of cloud-based tools on Agile hardware development. While our research was born out of the need to build knowledge specifically about cloud-based tools regarding Agile hardware development, this is only a starting point to build a wealth of knowledge in this field.

Our research raised the following questions that can be used for future research:

- What structure do organizational leaders use when adopting cloud-based tools for Agile hardware development teams?
- What Agile hardware development characteristics are driving the development of new features in cloud-based tools?
- What features and capabilities are hardware teams searching for when adopting cloud-based tools?
- What is the impact of interruptions from cloud-based communication tools on productivity?

8. References

Alaidaros, H., Omar, M., & Romli, R. (2021). The State of the Art of Agile Kanban Method: Challenges and Opportunities. Independent Journal of Management & Production, 12(8), 2535-2550.

Andrei, B., Casu-Pop, A., Gheorghe, S., & Boiangiu, C. (2019). A Study on Using Waterfall and Agile Methods in Software Project Management. Journal of Information Systems & Operations Management, 125-135.

Arimoto, P. et al. (2023, October 23). "Microsoft Teams Service Description." Microsoft Learn. https://learn.microsoft.com/en-us/office365/servicedescriptions/teams-service-description.

Atlassian. (n.d.). "What is Confluence: A Brief Overview | Atlassian." Available online at https://www.atlassian.com/software/confluence/resources/guides/get-started/overiew#about-confluence

Atzberger, A., & Paetzold, K. (2019, July). Current Challenges of Agile Hardware Development: What Are Still the Pain Points Nowadays?. In Proceedings of the Design Society: International Conference on Engineering Design (Vol. 1, No. 1, pp. 2209-2218). Cambridge University Press.

Beck, K. et al. (2001) "Manifesto for Agile Software Development." 2001. Available at http://agilemanifesto.org/.

Calefato, F., & Lanubile, F. (2016, August). A Hub-and-Spoke Model for Tool Integration in Distributed Development. In 2016 IEEE 11th International Conference on Global Software Engineering (ICGSE) (pp. 129-133). IEEE.

Calefato, F., & Ebert, C. (2019). "Agile Collaboration for Distributed Teams [Software Technology]." *IEEE Software*, 36(1), 72-78, https://doi.org/10.1109/ms.2018.2874668

Cooper, R., & Sommer, A. (2018). Agile-Stage-Gate for Manufacturers: Changing the Way New Products Are Developed Integrating Agile Project Management Methods into a Stage-Gate System Offers Both Opportunities and Challenges. Research-Technology Management, 61(2), 17-26.

Diaz, L. (2023, April 28). Using Jira Dashboards to Track Multiple Projects. Deiser Blog. https://blog.deiser.com/en/use-jira-dashboards-and-gadgets-to-track-multiple-projects

Drutchas, J. & Eppinger, S. (2022). Guidance on Application of Agile in Combined Hardware and Software Development Projects. Proceedings of the Design Society, 2, 151-160. https://doi.org/10.1017/pds.2022.16

Ebert, C., & Paasivaara, M. (2017). Scaling Agile. IEEE Software, 34(6), 98-103.

Eppinger, S., & Hirschtick, J. (2023). Transitioning to Agile Development: How to Develop Hardware Like Software. PTC White Paper, May 2023. Available online at https://www.ptc.com/-/media/Files/PDFs/corporate/transitioning-to-agile-how-to-develop-hardware-like-software.pdf

Figma. (n.d.). Figma Collaboration: The All-in-One Design Tool. Figma. https://www.figma.com/collaboration/

Five Flute. (n.d.). Design Better Products in Less Time. Five Flute. https://www.fiveflute.com/design-reviews/

Gliffy by Perforce. (2022, February 17). Collaborate in Confluence Cloud | Basic Confluence commands [Video]. YouTube. https://www.youtube.com/watch?v=vVZySaFnhOM

Hammarberg, M., & Sunden, J. (2014). Kanban in Action. Manning Publications.

Häring, I. (2021). Technical Safety, Reliability and Resilience. Singapore, Springer Nature.

Heimicke, J., Niever, M., Zimmermann, V., Klippert, M., Marthaler, F., & Albers, A. (2019, July). Comparison of Existing Agile Approaches in the Context of Mechatronic System Development: Potentials and Limits in Implementation. In Proceedings of the Design Society: International Conference on Engineering Design (Vol. 1, No. 1, pp. 2199-2208). Cambridge University Press.

Kramer, M. (2018). Best Practices in Systems Development Lifecycle: An Analyses Based on the Waterfall Model. Review of Business & Finance Studies, 9(1), 77-84.

Mancl, D., & Fraser, S. (2020, June). COVID-19's Influence on the Future of Agile. In International Conference on Agile Software Development (pp. 309-316). Cham: Springer International Publishing. https://doi.org/10.1007/978-3-030-58858-8_32

Michalides, M., Bursac, N., Nicklas, S., Weiss, S., & Paetzold, K. (2023). Analyzing Current Challenges on Scaled Agile Development of Physical Products. Procedia CIRP, 119, 1188-1197.

Microsoft Inside Track Staff (2023, July 12). “A Foundation for Modern Collaboration: Microsoft 365 bolsters teamwork.” Inside Track Blog, Microsoft, https://www.microsoft.com/insidetrack/blog/a-foundation-for-modern-collaboration-microsoft-365-bolsters-teamwork/

Microsoft Support (n.d.). Create a Presentation with Others. Microsoft. https://support.microsoft.com/en-us/office/video-creating-a-presentation-with-others-e7f9bba4-c6c4-4ada-adf5-70bcdf2aafa9

Microsoft Tech Community (n.d.). Now in Public Preview & Targeted Release: New Channels Experience in Teams. Microsoft. https://techcommunity.microsoft.com/t5/microsoft-teams-public-preview/now-in-public-preview-amp-targeted-release-new-channels/m-p/3825179

Mills, N. (2023a, July 26). Leap Motion Controller 2 – An Agile Electronics Case Study – Part 1. LinkedIn. https://www.linkedin.com/pulse/leap-motion-controller-2-agile-electronics-case-1-neil/

Mills, N. (2023b, August 2). Leap Motion Controller 2 – An Agile Electronics Case Study – Part 2. LinkedIn. https://www.linkedin.com/pulse/leap-motion-controller-2-agile-electronics-case-neil/

Miro. (2022, September 23). MiroXJira [Video]. YouTube. https://www.youtube.com/watch?v=mD9GgNmvXrc

Miro. (n.d.a). SAFE Program Board Template & Example for Teams. Miro. https://miro.com/templates/safe-program-board/

Miro. (n.d.b). What is a Fishbone Diagram? Miro. https://miro.com/diagramming/what-is-a-fishbone-diagram/

Miro. (n.d.c). What is Miro: Product Overview. Miro. https://miro.com/product-overview/ .

Modus Create, LLC. (2020, October 22). How to build a creative triaging solution in Jira [Video]. YouTube. https://www.youtube.com/watch?v=ht42VlNzHWY

Ovesen, N. (2012). The Challenges of Becoming Agile: Implementing and Conducting Scrum in Integrated Product Development [PhD Thesis]. Aalborg University.

Ozkan, N., Bal, S., Erdogan, T., & Gök, M. (2022, September). Scrum, Kanban, or a Mix of Both? A Systematic Literature Review. In 2022 17th Conference on Computer Science and Intelligence Systems (FedCSIS) (pp. 883-893). IEEE.

Ozkan, D., & Mishra, A. (2019). Agile Project Management Tools: A Brief Comprative View. Cybernetics and Information Technologies, 19(4), 17-25.

Pargaonkar, S. (2023). A Comprehensive Research Analysis of Software Development Life Cycle (SDLC) Agile & Waterfall Model Advantages, Disadvantages, and Application Suitability in Software Quality Engineering. International Journal of Scientific and Research Publications, 13(8), 120-124. https://doi.org/10.29322/IJSRP.13.08.2023.p14015

Radeka, K., & Iberle, K. (2022). When Agile Gets Physical: How to Use Agile Principles to Accelerate Hardware Development. Washington, Chesapeake Research Press

Saleh, S., Huq, S., & Rahman, M. (2019, February). Comparative Study Within Scrum, Kanban, XP Focused on Their Practices. In 2019 International Conference on Electrical Computer and Communication Engineering (ECCE) (pp. 1-6). IEEE.

Saez, A. (2023, September 4). "Cloud-based versus cloud-native: what's the difference?" Cloud Native Computing Foundation. https://www.cncf.io/blog/2023/09/04/cloud-based-versus-cloud-native-whats-the-difference/

Schmidt, T., Weiss, S., & Paetzold, K. (2018). Expected vs. Real Effects of Agile Development of Physical Products: Apportioning the Hype. In DS 92: Proceedings of the DESIGN 2018 15th International Design Conference (pp. 2121-2132).

Schmidtner, M., Doering, C., & Timinger, H. (2021). Agile Working During COVID-19 Pandemic. IEEE Engineering Management Review, 49(2), 18-32. https://doi.org/10.1109/EMR.2021.3069940

Stellman, A., & Green, J. (2014). Learning Agile: Understanding Scrum, XP, Lean, and Kanban. California, O'Reilly Media Inc.

Slack. (n.d.). "Slack Features: One platform for your team and your work." Slack, https://slack.com/features

SmartSheet. (n.d.). How to Use Gantt Charts [Video]. SmartSheet. https://help.smartsheet.com/learning-track/level-1-get-started/basic-gantt-view

Ullman, D. (2019). Scrum for Hardware Design: Supporting Material for the Mechanical Design Process. Oregon, David Ullman LLC.

Ulrich, K., & Eppinger, S. (2016). Product Design and Development. Sixth edition. New York, NY: McGraw-Hill Education.

Walsh, D. (2021, May 12). “How Cloud CAD Improves Collaboration with Your Extended Internal Product Design Team.” PTC Onshape, https://www.onshape.com/en/blog/how-cloud-cad-improves-collaboration-with-your-extended-internal-product-design-team

Wong, K. (2020, September 23). Microsoft Ignite brings enhancements to Teams, including support for up to 20,000 participants. HardwareZone.com.sg. https://www.hardwarezone.com.sg/tech-news-microsoft-ignite-brings-enhancements-teams-including-support-20000-participants

Wright, R. (2023, December 14). Learning Jira (Cloud Edition) Online Class. LinkedIn Learning. https://www.linkedin.com/learning/learning-jira-cloud-edition-19890900