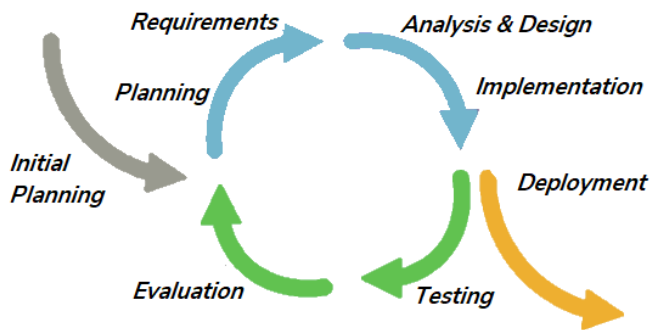


Article Series



Agile for HW Vs. Concurrent Engineering

INTRODUCTION

In the world of hardware product development, there are a range of prominent methodologies that are vying for pre-eminence as the “best” approach to navigate the complex process of bringing innovative products to market. In this article, we’ll address two of these: Concurrent Engineering and Agile for Hardware. While both approaches share a common goal of improving efficiency and reducing time-to-market, they do so in distinct ways. The first question one might ask is, “Are they complimentary or competing?” In this article, we will explore the key elements of a product development process and compare how they are addressed in Concurrent Engineering and Agile methodologies. Along the way, we’ll look for similarities and differences and proceed to answer the question posed, “Are they complimentary or competing?”

By Dorian Simpson

A QUICK REVIEW OF EACH PROCESS

Before diving into the details, let's establish a general understanding of these two approaches. Concurrent Engineering is a methodology that emphasizes cross-functional collaboration and parallelization of tasks. It aims to streamline the product development process by breaking down silos and ensuring that all stakeholders are involved from the early stages. Development progresses through a series of cross-functional design efforts in parallel (concurrent) and relies heavily on techniques pulled from QFD, DFM, FEMA and others, as well as traditional project management. Generally, it can be considered a rather heavy process and success is often dependent on the enabling tools such as digital CAD systems that can be aligned and integrated across teams.

Agile for Hardware also embraces these principles, but centers on a more incremental approach that focuses on adaptability, customer collaboration, and delivering value in short cycles. It can be considered more of a framework based on key principles with flexibility to incorporate tools that make most sense for a given project. It's important to note that many readers may associate "agile" with the pure software-focused variants such Scrum and SAFe™. However, the MAHD (Modified Agile for Hardware Development) Framework™, which will be the variant used for our comparison, is built on agile principles, but modifies commonly understood tactics to address the needs of hardware development. The MAHD Framework also incorporates some of the elements of QFD, Lean and design-thinking, but relies more heavily on team-driven milestones based on iterative cross-discipline planning and backlog management vs. traditional project management techniques.

Table 1 summarizes several of the key elements of a product development process and see how Concurrent Engineering and Agile for Hardware with the MAHD Framework tackle them.

The MAHD Framework also incorporates some of the elements of QFD, Lean and design-thinking, but relies more heavily on team-driven milestones.

Key Elements	Concurrent Engineering	Agile for Hardware (MAHD)
Requirements Gathering	Detailed upfront planning and specification	High-level requirements with room for flexibility
Design	Sequential process with distinct phases	Iterative design and continuous improvement
Prototyping	Sequential prototyping after design completion	Early and frequent prototyping throughout
Testing	Sequential testing after prototyping	Iterative testing and integration
Documentation	Comprehensive documentation for each phase	Light documentation, emphasis on demonstrable output and documentation as the solution evolves
Feedback and Iteration	Limited feedback loops and iteration opportunities	Frequent feedback loops and iterative improvements
Communication	Formalized communication channels and reporting	Cross-functional collaboration and face-to-face interactions
Project Management	Linear project management approaches	Self-organizing teams and adaptive project management

Table 1: Comparing Common Process Elements

A DEEPER DIVE

While there are many similarities in how these key elements are addressed, there are notable differences between Concurrent Engineering and Agile for Hardware. Let’s delve deeper into some of them:

- 1. Requirements Gathering:** Concurrent Engineering relies on detailed upfront planning and specification (similar to a waterfall-type process), aiming to define complete product requirements early on and have the design validated during early prototype stages. The MAHD Framework using agile principles, on the other hand, embraces changing requirements and allows for flexibility, focusing initially on high-level requirements that are refined as the project progresses through learning cycles. A key agile principle is that “requirements” are only finalized once critical customer needs are validated as satisfied.
- 2. Design:** Concurrent Engineering attempts to follow a concurrent design process, but the reality is that the design must be somewhat complete in early stages so that parallel work can continue. In the MAHD Framework, design is also an iterative process with milestones based on project-team determined “IPAC” learning objectives. Adjustments to the plan are then made based on customer feedback, iterative learning and changing market conditions.

Concurrent Engineering places significant emphasis on comprehensive documentation for each phase, ensuring traceability and accountability.
- 3. Prototyping:** Concurrent Engineering typically involves sequential prototyping after requirements are established. The MAHD Framework encourages early and frequent prototyping throughout the development process based on a “vertical slice” of the system under development to identify and refine "requirements." This leaves the definition of prototype as flexible based on the stage of the project and learning needs and allows for rapid feedback and adjustments.
- 4. Feedback and Iteration:** Concurrent Engineering often includes feedback loops and iteration opportunities with an emphasis on technical validation, as the focus is on completing each design iteration before moving forward. The MAHD Framework emphasizes frequent feedback loops, with an emphasis on stakeholder feedback enabling continuous learning and adaptation.
- 5. Documentation:** Concurrent Engineering places significant emphasis on comprehensive documentation for each phase, ensuring traceability and

accountability. The MAHD Framework promotes lighter documentation and prioritizes demonstrable output, such as digital models or the partially integrated vertical slices of the system over extensive paperwork. Documentation is more focused on stakeholder needs and communicating outcomes vs. pre-development definition.

- 6. Project Management:** Concurrent Engineering often employs linear project management approaches, with predefined timelines and milestones. The MAHD Framework, however, favors self-organizing teams and adaptive project management based on team-driven learning and execution cycles, allowing for flexibility and responsiveness to changing circumstances.

Both methodologies attempt to fix one of the fundamental problems of traditional waterfall processes, which is that processes such as Stage-gate are good at providing an overall governance structure, but provide little to no guidance on how teams should work together. However, Concurrent Engineering falls into a similar trap where detailed requirements are assumed to be accurate in the earliest stages of development, and this is rarely reality.

A SIMPLE WAY TO COMPARE

Generally, the more unknowns there are in a project, the more value and benefits can be derived from an agile principles-based process such as the MAHD Framework. However, similar to waterfall approaches such as Stage-gate™, teams using Concurrent Engineering often find comfort in establishing detailed requirements up front.

One simple way to consider these two processes is that Concurrent Engineering can provide an overall approach for iterative design and development, while the MAHD Framework provides a more flexible agile framework to enable faster starts with more unknowns and then using iterative (IPAC) learning and execution milestones to establish adaptive design targets. The design toolsets that are more defined in Concurrent Engineering can also be brought into an agile framework as necessary.

IN CONCLUSION

Concurrent Engineering and Agile for Hardware (based on the MAHD Framework) offer distinct, but overlapping approaches to hardware product development. While both approaches focus on cross-functional collaboration, parallelization and incremental design, Concurrent Engineering requires early, nearly complete product definition and collaborative design tools, while the MAHD Framework emphasizes agile's adaptability, team-driven iterative milestones and customer collaboration principles. Understanding your goals and the desired elements of the ideal product development process that achieves these goals are a good start to any process transformation. You can then develop an overall framework that allows for the inclusion of methods, tactics and tools that can help teams make informed decisions and choose the methodology that best aligns with their project requirements and desired way-of-working.

To address the question, "Are these two approaches complimentary or competing?", the answer is both. Teams can relatively easily incorporate Concurrent Engineering principles and tools into the MAHD Framework, while it is also possible, with a little extra effort, to add agile principles to Concurrent Engineering.

Ultimately, whether you opt for Concurrent Engineering or the MAHD Framework, the aim of any process should be to develop efficient, streamlined and flexible processes that drive innovation, reduce time-to-market, and deliver exceptional products to customers.

How Can We Help?

The MAHD framework, developed by Gary Hinkle and Dorian Simpson, is built on agile principles with input from dozens of companies ranging from electronics to medical equipment to address the needs of hardware development. Having both been involved with product development for decades and experience with development processes ranging from adhoc to Six Sigma, we have seen the challenges posed by many NPD processes and how Agile can help. However, working with teams trying to implement Agile processes designed for SW development, we were determined to find a better way to get the benefits of Agile while solving the needs of physical development.

Below are three recommendations to get started. To learn more, visit:

www.MAHDframework.com

1

GET AN OVERVIEW

Visit www.mahdframework.com to set up a 45-minute consultation

- Review your situation and goals
- Overview of the MAHD Framework
- Determine if MAHD is right for you
- How to best get started

2

TRAIN AND PILOT

The fastest way to start is to train a focused team and pilot a project

- Identify a pilot team and project
- Get hands-on MAHD training
- Execute MAHD w/facilitation
- Learn, improve and repeat

3

ASSESS AND PLAN

This is a great option if you know MAHD is right for your whole organization

- Establish clear NPD goals
- Identify areas for improvement
- Develop a roadmap w/milestones
- Pilot, learn, expand

Contact: info@mahdframework.com to get started or visit www.mahdframework.com to set up your complimentary consultation.

About MAHD Framework LLC

MAHD Framework LLC was established to help product development teams take their NPD capabilities to a new level of performance with Agile principles and methods. We provide a wide range of materials, training and certifications to ensure your team has the skills and tools necessary to succeed.

Challenges we address include:

- Slow or inefficient NPD processes
- The need to adapt quickly to market needs
- Improving ROI for new projects
- Achieving predictable NPD results
- Aligning hardware & software processes

Contact us to discuss how we might help you accelerate your NPD efforts.