Step-by-Step MAHD

Modified Agile for Hardware Development

The Smart Coffee Maker Project

Part 9 of a 9-part series to walk through an agile development project from concept to launch

Step 9:

Project Complete!

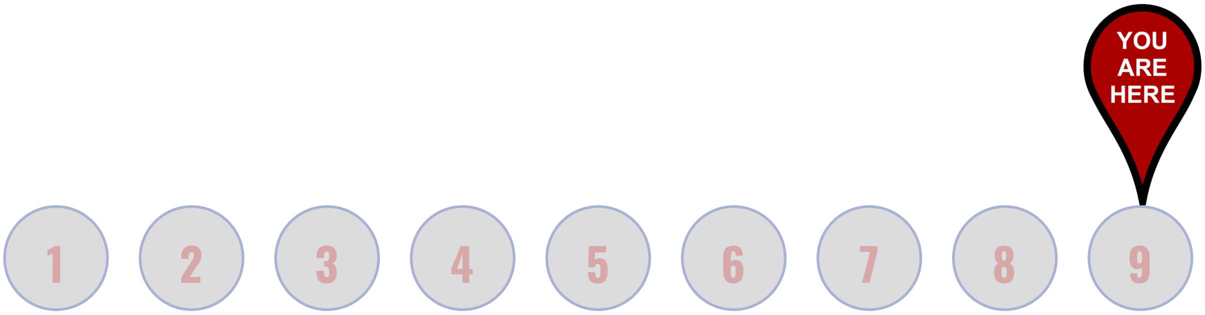
By Dorian Simpson and Gary Hinkle

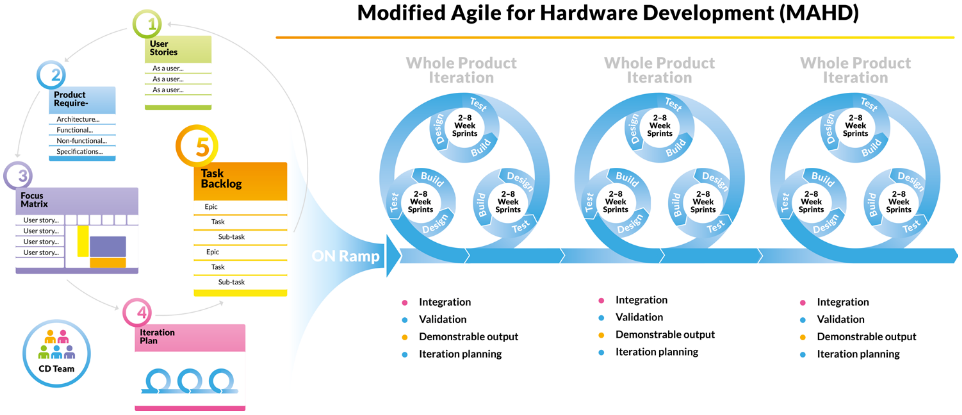
A Quick Intro to MAHD

Agile methods have proven superior over traditional product developement processes to quickly adapt to customer needs, reduce waste and accelerate development. However, the application of agile requires significant changes to support the needs of hardware products. This led to the development of the Modified Agile for Hardware Development (MAHD) Framework — an open-source initiative to embrace the principles of agile while recognizing hardware's unique needs.

THE COFFEE MAKER PROJECT: STEP-BY-STEP AGILE IN NINE STEPS

To help practitioners visualize the MAHD Framework, we have developed a series of nine articles to explain how agile methods and tools can be used for physical products, who should be involved, the deliverables for each step and tips for how to overcome challenges. We hope you'll join us on this journey as JavaBrew uses the MAHD Framework to develop an innovative new coffee maker.





The MAHD Framework: Similar to Agile for Software, but with Important Differences

Learn More

To learn more about the MAHD Framework, download related ebooks and whitepapers, or sign up for e-learning opportunities, visit **[www.agileforhardware.org.](www.agileforhardware.org)**

Step 9: Project Complete!

STEP 9: THE SITUATION

It has been over a year since the JavaBrew team started their smart coffee maker project. Since this was their first agile project, you can imagine it has not been perfect. Of course there were doubters and the old traditional ways of product development crept into both mindsets and behavior. This was all to be expected. But since the team was committed to changing and realizing the benefits of agile, they diligently worked through the transition challenges.

With each MAHD Iteration and Sprint, the team got better at planning, estimating tasks and keeping each other accountable. The hardest part of their transition was managing to the rough schedule they started with during early MAHD Onramp planning. The team (and management) was not accustomed to working without a detailed schedule and it seemed that for every task they finished from the backlog, another task was added. Initial estimates of major tasks seemed to explode when the tasks were broken down each sprint. While some major tasks were much simpler than expected, this was the exception. And even though they had a set of high priority user stories and product goals, the team had to constantly question whether the product they were designing would achieve the market goals. This required them to learn, adapt and reprioritize tasks with each sprint and iteration review. In other words, it forced them to become agile.

If you look back to the original iteration plan as developed in Step 3 and shown in Exhibit 1, the JavaBrew team was able to follow the overall plan closely, especially the critical prototype plan. However, as you should expect with agile, many details changed throughout the project. After each iteration, the team had to account for dependencies, mitigate risk, question the schedule, make tough design decisions, manage executive expectations and fight off other project priorities.

Let's look at what happened from the first iteration until the last and then we'll look at what worked well during JavaBrew's transition to agile and what didn't work. Finally, we'll identify the key lessons all can take away based on their journey. Note that several of these Iterations and associated challenges have been discussed in our previous steps, but of course there were many project details we could not include.

> **The MAHD On-ramp:** In the first five papers in this series, the team worked through each step of the MAHD On-ramp. They developed product goals, wrote user stories (Step 1), identified major product attributes, clarified areas of focus (Step 2), developed the initial iteration plan as shown in Exhibit 1 (Step 3), built their product backlog (Step 4) and prepared for their first sprint (Step 5). The team was learning, excited and focused. Of course, reality would set in soon as execution began.
>
> **Iteration 1: Validating the Value Proposition**
>
> The first iteration was initially planned to consist of three, two-week sprints (Step 6), but this didn't work. They team quickly fell behind, missed tasks and had to rethink their plan for getting early customer feedback. This was the period that Lynda (the new coffee maker's product manager) was on the hook to find customers and build a feedback mechanism (Step 7).

Iteration 2: Refining the Product Concept

In the second iteration, questions loomed as to which features should be included or removed. Resources were still an issue and costs were unknown, but early indications were that costs were out of control. Management also created demands and had expectations that may or may not be consistent with market needs, such as the inclusion of voice control features. The schedule was also a big concern since undefined tasks started to balloon as they learned more about the project details.

Iteration 3: Nailing Down the Product Definition

Through early customer engagement and prototyping the team was able to finalize the desired feature set. This was good and bad. It created focus, but the “final” product definition created significant resource, cost and schedule concerns. The team had to continue to prioritize tasks and create contingency plans, especially if the early prototypes showed the product would need significant changes, such as the need to add a multiple-bean option as shared in Step 8. Customers loved the ability to select beans as well as the retro-looking industrial design of their early concept, but getting to the final design wasn't easy.

Iteration 4: Design Lockdown and Early Manufacturing Readiness

In Iteration 4, the team was able to test with nearly working prototypes. The product worked but needed major refinement. They were supposed to have the product design completed at 100% but this was not possible since the product was still being tested. Management was nervous about the cost and manufacturing was concerned about the Bill of Materials (BOM). This forced the team to rethink how they approached supplier management in an agile environment. While the team continued their innovation efforts in order to reduce cost and meet customer needs, managing the uncertainty across the company became a critical challenge for the agile project team.

Iteration 5: Product Complete and Production Ready

Everything started coming together in Iteration 5, but with the problem they identified implementing the product’s “multiple bean” feature, the whole project was in serious jeopardy (see Step 8 for details). This forced the team to make the typical project tradeoffs of whether to add resources, extend the schedule or reduce scope. They decided to add external resources, but this highlighted that traditional project challenges never go away in an agile environment. However, this critical risk was identified early so it could be managed.

Iteration 6: Market Launch

In the final iteration the focus changed from product readiness to market readiness. While production challenges still consumed time and resources, other elements such as sales, marketing and channel factors came critically into focus. The agile team expanded with new team members which create new challenges for Jordan (the Agile Project Manager.) The added team members were often new to agile and it took time to explain the importance of planning tasks and managing to sprints. Major new tasks such as international regulatory compliance, privacy policies and even patent conformance also created many “fires” that were not anticipated. While these items were in the backlog, they didn't rise to high importance fast enough to manage easily. All normal for teams transitioning to agile.

STEP 9: PROJECT RETROSPECTIVE: LESSONS LEARNED

Many teams follow a similar path to JavaBrew when starting a transition to agile and adopting the MAHD Framework. As we shared in earlier steps, while the above iterations might look a lot like a traditional waterfall project plan, the way the team worked together, validated work-in-progress, updated plans and made decisions was far different. The major iteration milestones gave the team structure and clear interim goals but did not represent any form of "gate" that had to be passed before moving forward. Learning and feedback was built into every step and any time there was a roadblock or information needed, the task was always prioritized and work continued on other important items.

After Iteration 6, the final iteration, was complete, Jordan gathered the JavaBrew team to lead a project retrospective discussion. The team came to the following conclusions:

What worked well

1. Overall, the team was happy with their transition to agile. They were able to kick off the project fast with minimal product definition and project planning.
2. Developing user stories up front provided a clear customer focus but the process for updating them consistently and tracking progress to ensure the top user stories were satisfied was not obvious. After several iterations, this became easier.
3. Team collaboration worked well, but it took some time to optimize. As the range of tasks grew, it was difficult to bring in new team members that were only partially committed to the agile team.
4. Having a solid, updated MAHD Iteration Plan was critical to success. Without this, the team realized that working only from the backlog of tasks without this big-picture structure would have been frustrating. In addition, the uncertainty of milestones, schedules and dependencies would likely have caused the team to fall back to traditional project management techniques.
5. Identifying risks and planning contingencies early worked well. If Jim had not found an outside firm that could help the team with the mechanical design, the outcome may have been far less successful.

What Didn't Work

1. Generally, all areas of agile had a learning curve. Nothing was insurmountable, but more time could have been allocated to learn. The team may have started using agile with a less mission-critical project.
2. Jordan was a bit overloaded. His role as the Agile Project Manager was essential, but he had to prepare for every meeting and hold pre-meetings with each functional lead to refine their tasks, prioritize all work items, manage the releases, etc. This role could have been more distributed with functional leads taking more responsibility for preparing for sprint meetings or taking some of the scrum master role.
3. Management did not completely buy into the role of customers in the development process — at least not at first. Executives had their own vision, which was great, but they needed to trust that the designers and development team were experts and were using real customer feedback to guide decisions.
4. Velocity tracking was difficult! It took almost four iterations to fully define the project and really understand if the team was on track. With experience, the team believes this can be improved.

What Could Be Improved

Some areas the team agreed they should work on for their next project include:

1. Spend more time in Iteration 1 to define the project, set up the customer feedback mechanisms and identify the right resources.
2. Train new team members on agile techniques and mindset before they start participating in sprints and refinement meetings.
3. Ensure all stakeholders – production managers, executives and even partners – are familiar with the agile development approach so they know what to expect when they see a prototype, iteration plan or other agile artifact that may not look like what JavaBrew delivered in the past.
4. Share the scrum master role with others. Let functional leads help coach, prepare for meetings and share the responsibility. This will not only help them grow, but allow the Agile Project Manager to focus more on critical project tasks.

PROJECT OUTCOME

While the JavaBrew project is based on a variety of real-world projects, it focuses on a fictional product. The authors could obviously give it any ending they wanted. Some of the possible outcomes were discussed in the previous step included; 1) delaying the project to fix the “coffee bean option” feature, 2) using an external vendor to accelerate delivery, or 3) de-featuring the product to hit their release target date. These would all be good outcomes, but we all know projects can have a variety of far worse outcomes. For example, management might have been frustrated with the project problems and decided to just kill the product. or the outside vendor Jim selected may have created a disaster and designed a coffee bean selection mechanism that had more problems than the original design.

In JavaBrew’s case, the team had a successful outcome. They were able to launch the product successfully, but of course it was not perfect. Since the team was new to creating apps and smart technology, they needed to manage unique customer inquiries such as “I can’t download your app!” and “Why is it making coffee at 1 in the morning?!” With the right agile mindset and customer focus these are details that can always be worked through.

For JavaBrew’s next project, they will learn from their previous experiences and continue to refine their MAHD process. As with every business improvement, the transition is always a journey more than any specific event. It will take time, continuous learning and a concentrated effort to reap both short and long-term benefits.

NEXT STEPS AND THANK YOU

Agile methods, including the Modified Agile for Hardware Development (MAHD) Framework, is not a panacea for perfect development. However, the authors know from experience that agile methods provide significant benefits over traditional waterfall development projects, including:

1. Agile ensures a fast, successful start to a project, including large projects with many unknowns.
2. Agile provides the mechanism for continuous customer engagement, learning and adjustments.
3. Agile provides early awareness of major technical and market problems — enabling you to remove risk before it becomes critical.
4. Agile allows you to see real-time progress for both teams and individuals.
5. Agile creates a collaborative environment where communication is central to success.

If you made it this far into our agile journey, we are both thankful and impressed. Hopefully by now you also see the benefits agile can provide, but also that agile as defined for software must be modifed for physical products.

If you'd like to learn more about the MAHD Framework, train your team on how to implement MAHD, or just want to share your own journey, we'd love to hear from you.

Best of luck to you!

Gary and Dorian

EXHIBIT 1

Exhibit 1: Initial Iteration Plan

Preparing the big picture in readiness for sprint planning (From Step 3)

	Iteration 1	Iteration 2	Iteration 3	Iteration 4	Iteration 5	Iteration 6
Major Deliverables	• Preliminary concepts • Preliminary features scoped	• Cost estimates, initial industrial design, electronics design • Smart features defined – app/ voice	• Coffee making mechanism design • Integrated device with smart control for testing	• Integrated device with ~80% target feature set • Smart app 90% complete	• Mechanical and electronics designs complete • Packaging design compete • BOM complete	• Production drawings, user guide, website, materials, packaging • Forecasts, marketing plan
Prototypes	Preliminary brochure	3D video highlighting smart features and ease of use	Rough working prototype – smart features + physical design	Fully working prototype	Production-ready prototype	Production starts
Key Questions	• Is value proposition right? • Which features drive value? Physical interface needed?	• Value of voice control? • App features? • Can cost meet targets?	• Do customers love it? • Any design show-stoppers? • Can we hit the market window?	• Can we really lock down features? • Are retailers on board? • Can we create demand?	• Is it ready for production? • Is the quality “good enough?” • Are costs under control?	• Are retailers ready to take orders? • Can we hit the forecast? • Is sales engaged?
Milestones/ Approvals?	Concept and plan approved	Cost estimates approved	Major feature lockdown	Design/feature lockdown, prototype tooling approved/ ordered early in iteration	BOM complete, launch plan, final tooling, compliance & certifications	Pricing, sales and channel ready
Risk Mitigation	• Team momentum – need committed resources. • Early customer engagement.	• Schedule risk – team velocity on target? • FCC compliance.	Design approvals	• Prototype tooling approvals. • Final feature set approval.	Production tooling approvals.	• Sales and marketing risks. • Engage with channel partners.
Target Date	3/15/19	5/24/19	8/2/19	12/6/19	4/10/20	7/3/20
Number of Sprints	3	5	5	9	9	6 (2 sprint buffer for 7/31 launch)

ABOUT US

To Be Continued...

GET THE SERIES

To see the previous steps and receive each new step of this project as it is published, visit www.AgileForHardware.org. Each step will be available for download and sent directly to your email.

ABOUT THE AUTHORS

The MAHD framework is an open-source process, available for all to use, build on and improve. We look forward to hearing from you and your experiences with agile, waterfall and other processes. The MAHD framework was developed by Gary Hinkle and Dorian Simpson to address the needs of hardware development.

To learn more, get involved, or just join our community for discussion, visit:

www.AgileforHardware.org

About Gary Hinkle

Electronics, mechanical and software engineering are all part of Gary Hinkle's background, working in design, management and executive leadership of communication, industrial, telemetry, audio, avionics, computers, test & measurement, among other industries. Today, he's principal consultant at Auxilium, a company he founded to help engineering-oriented businesses increase productivity.

Contact Gary

W: www.Auxilium-inc.com
P: 971-222-6234
E: gary@auxilium-inc.com

About Dorian Simpson

Dorian Simpson is an innovation and product development consultant, trainer, speaker and author of *The Savvy Corporate Innovator.* Companies he's worked with include ABB, Tyco, Owens Corning, Technicolor, FEI, VTech and Freightliner. Before consulting, Dorian held positions at Motorola and AT&T in product management, sales, marketing, business development, and engineering.

Contact Dorian

W: www.KingsleyInst.com
P: 971-235-4905
E: dorian@kingsleyinst.com

Kingsley Institute
for Strategy and Innovation

Auxilium