

Step-by-Step MAHD

Modified Agile for Hardware Development

The Smart Coffee Maker Project

Part 5 of a 9-part series to walk through an agile development project from concept to launch

Step 5:

Planning Your First Sprint



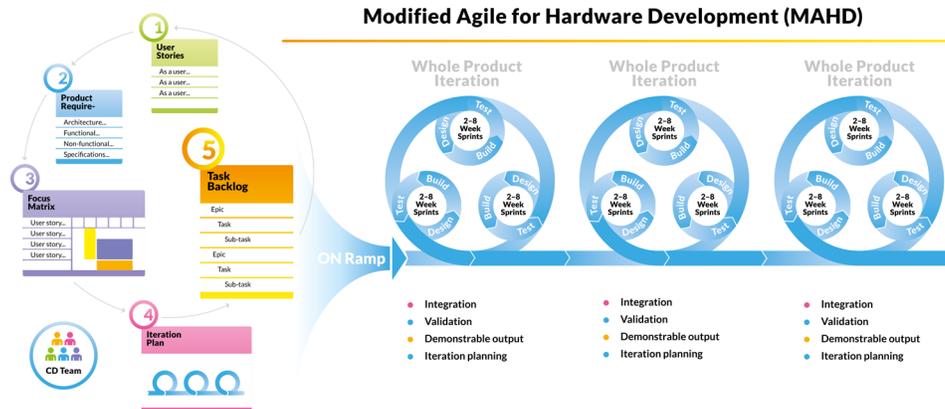
By Dorian Simpson and Gary Hinkle

A Quick Intro to MAHD

Agile methods have proven superior over traditional product development 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.

Step 5: Planning Your First Sprint

THE SITUATION

In the previous step, Jordan, the Agile Project Manager (APM), led the team to develop their initial backlog — the prioritized list of high-level work to be done. The team categorized the work very roughly and identified major buckets such as mechanical design work, electronics/software/interfaces, and other elements as temporary placeholders to organize tasks. Jordan captured the work items during their meeting using Excel.

In this step 5, the team will further refine their backlog and identify the tasks that team members will **commit to in their first sprint**. The team agreed that each sprint duration will be two weeks, starting on Monday and ending 10 business days later on Friday. The high-level work buckets identified in step 4 will now need to be broken down to tasks each team member can be fully execute during the sprint., especially those tasks where the team needs to make immediate progress.

It's Thursday morning and the team has gathered in their project room ready for action.

AGILE ACTIVITIES

Before the sprint planning session, Jordan made sure that every team member had access to the project backlog so that each development lead could work through the major tasks and break down the priority 1 items into smaller work items. This effort won't be done until the team meets since only the task owner can actually commit to completing it.

Jordan is now displaying the project backlog so that all team members can participate in planning their sprint. As the meeting gets started, Jordan explains that the meeting has three goals:

1. All team members will have enough work defined for the sprint, with clear ownership and commitment to get it all done in the two-week period.
2. Agree on acceptance criteria for every task in the sprint so it's clear how the tasks will be defined as completed.
3. Some tasks for the next sprint will be defined so that team members can get more done if they finish early, or if they get stuck they can keep progress going on other priorities.

After tasks are broken down as needed, task owners are identified. The team estimates these tasks together when possible or lets the subject matter expert estimate the task. E.g. only the electrical engineers could really estimate how long it would take to define the circuitry needed for the physical interface. Once the task is defined, they then agree on acceptance criteria.

One major area of discussion is whether to use person-hours for task estimates, or “points” – a relative scale commonly used by agile teams. They decide to use hours because it seemed difficult to determine what a "point" meant for the wide range of tasks from concept design to app development.

Jordan knows that it's good practice to have the next two sprints also broken down to task level so that if team members are unable to complete tasks planned for the current sprint (if they are blocked by external dependency or other factor), they have other high-priority tasks that can be completed. This also gives team members who finish all their tasks in the sprint clear direction for additional work that can be completed. Another reason is that it makes the each sprint planning meeting faster by always looking a couple sprints ahead.

They agree that every *planned* work item in the backlog must eventually be estimated roughly in terms of effort, but only the near-term work would be estimated with granularity and accuracy. The estimate for the entire project will become more accurate when the product is fully defined and no major changes are planned. Of course there will be many items in the backlog that will never get done for this project since all feature requests and ideas go into the backlog as a low priority.

For now, the team only estimates tasks for the first sprint (plus a few for the next sprint) so they can start delivering value right away. They are ready to get going!

STEP 5: CONFLICT AND RESOLUTION

In the first four steps (a real time effort of just several weeks) the team was in early planning mode. As in most companies, once a project has been approved, there is not a lot of team conflict since the tough decisions don't need to be made for quite awhile. In traditional projects, this may be months away since the product manager will define the product, write a product requirements document, pass it over to the development team for review and everyone ramps up slowly to meet the requirements. This is sometimes referred to as the project honeymoon period.

With agile, this honeymoon period is much shorter, but with good communication and rapid learning, should provide a much better environment for long-term success. Even now, as the team transitions into project execution with the first sprint, a great deal of debate has started on the major coffee maker features. Should the new maker have voice control? Does it need a physical interface? How will a hardware company manage consumer apps? All good questions that must be answered quickly and accurately. Does this team have the agile skills for success?

STEP 5: OUTCOMES

The exhibit on the following page shows the team's early backlog using a simple spreadsheet format.

Exhibit 1: Backlog and Sprint Plan in Table Format

While the JavaBrew team will eventually use an agile project management tool such as Atlassian's Jira, Excel spreadsheets are fine to get started. The table format shown illustrates that special tools are not needed to manage agile backlog and sprints. Many teams use readily available tools for agile planning, management and communication such as Excel, whiteboards, sticky notes, etc. The number on the right of each task is the estimate in hours. Large tasks requiring more than one sprint of effort, often considered epics (such as the task "Create conceptual drawings"), is broken down into smaller tasks. The tasks highlighted in green are committed for the first sprint and given acceptance criteria. Those not highlighted are planned for later sprints.

As progress is made, task owners will advance their tasks to the appropriate status. For every task completed the team will get credit for completing the hours estimated. However, tasks that don't get completed do not get partial completion – they carry over to the next sprint. This provides motivation to complete a task (as determined by the acceptance criteria) and drive commitment. E.g. completing 90% of a task is very different than completing a task!

This concept is important for determining the team's velocity, which will be established after completing several sprints. Their velocity will be 383 person hours per sprint solely based on the estimate for this sprint, but that will change as they learn what they can actually complete, and as team members move in and out of the project. When their average velocity is established and all planned work is roughly estimated, they'll know if they are on track to complete iterations and the entire project on schedule.

NEXT STEP

The team will meet again in two weeks to conduct a review of sprint one and to plan for their next sprint. With each sprint, they will also evaluate their progress and determine if they are on track to achieve their iteration goals as well as their overall project objectives. Join us for Step 6 and let's see how they are doing. Will the JavaBrew team embrace agile methods and learn to manage a complex project without the perceived certainty of a detailed schedule or will they quickly fall back to their previous project management practices?

Exhibit 1: First Sprint Tasks

Using a simple tool to manage a backlog and sprints.

Each team member had 80 hours of capacity for the sprint. Any task requiring more than 80 hours needed to be broken into sub-tasks to ensure they would fit within a sprint.

Pty.	Task Description	Est. (Hrs)	Owner	Acceptance
Mechanical Design				
1	Create conceptual drawings	80	Frank	
1	1 Develop 3 unique designs	40	Frank	Team review
1	1 Finish conceptual drawings	40	Frank	
1	Innovative filter design	48	Chenghao	Report
1	Investigate options for materials	32	Chenghao	Report
1	Refine initial production cost estimates	8	Jordan	Team review
1	High level block diagram	16	Jason	Design review
1	Decide about grinding mechanism	8	Frank	
1	Define product architecture	300	Jason	
2	Determine water reservoir size	16	Chenghao	
2	Prototype tooling	200	Frank	
2	Finalize carafe and housing materials	70	Chenghao	
3	Determine color options	8	Lynda	
3	Production tooling	400	Frank	
3	Other design tasks	TBD	TBD	
Interface, Electronics, Smart Features, Software				
1	Define SW interfaces – embedded & app	36	Jason	Design review
1	Electrical interface initial plan	28	David	Design review
2	Research voice controlled interface	16	Jason	Report
2	Research iOS & Android control methods	12	Jason	Report
2	Control and power electronics	80	David	
2	Physical interface electronics (if needed)	TBD	David	
2	Physical interface firmware (if needed)	TBD	Jason	
3	Develop control apps (iOS and Android)	320	Alec	
Other Elements				
1	Develop prototype brochure for feedback	20	Lynda	Team review
1	Get full resource commitments	6	Jordan	Updated plan
1	Customer engagement plan	60	Lynda	Team review
1	Setup configuration/change management	8	Jordan	Admin review
1	Define roles and responsibilities	4	Jordan	Team review
1	External resource plan	8	Jordan	Management Review
1	Add risk mitigation tasks to Jira	1	Jordan	Team review
1	Get customer input about voice control	20	Lynda	
1	Get input about need for physical interface	20	Lynda	
2	Complete 3D video	80	Frank	
2	Setup bug/defect tracking	8	Jim	
2	Determine first release MVP	24	Lynda	
2	Capital expense approval	24	Jordan	
2	Patents filing	40	Jason	
3	Retail package design	120	Frank	
3	QA plan	80	Jim	
3	User documentation & disclaimers	160	TBD	
3	Compliance testing	80	Jim	
3	Final product verification & validation	TBD	Jordan	
3	Manufacturing ramp-up plan	80	Jim	
3	Release/launch plan	120	Lynda	
3	Compliance testing	80	Jim/Ops	
3	Manufacturing ramp-up plan	80	Jim/Ops	
3	Release/launch plan	120	Lynda	

EXHIBIT 1

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

