BEYOND SOFTWARE:
Applying Agile Principles to Connected Product Development

accenture
By 2020, there could be more than 75 billion devices connected to the internet. ¹ What’s driving their proliferation? A combination of pervasive wireless connectivity and low-cost hardware platforms. The good news? Plenty of opportunities for product developers with great device ideas.

The bad news? Many product developers’ projects will fail to meet expected financial goals. Declining R&D ROI over the past five years provides ample evidence.² Our experience suggests that products frequently fail for the simple reason that they don’t impress customers. The principal reason for that is incorrect assumptions about customer requirements are often discovered far too late in the product development project cycle.

Agile development methods can help with this. But many product developers dismiss the incremental approach Agile requires. They assume, incorrectly, that Agile is only useful for software teams.

Product developers do face challenges when they try to adopt Agile methods. But with the right help and by following some key steps, it is possible to introduce Agile methods to product development and gain some significant rewards.

"Properly wielded, Agile can be used to handle even the largest programs with complete visibility and a sense of “what are we going to make happen today.”

Chris Eidler
VP Solutions Engineering, HPE
A typical product development project employing a phase-gated model will make early and extensive assumptions about customer needs and technical feasibility. These assumptions are made during expensive, upfront definition and design phases. They represent significant project risks as they are effectively only hypotheses about what will delight the customer and the feasibility of the enabling technologies.

And all too often, those assumptions are only discovered to be incorrect when it’s too late to change course. That leaves project leaders with a dilemma: do they add a new feature and accept a major delay, hold the schedule but release a poor product or cancel the project completely?

Software developers, in contrast, know a better way to develop the right product, faster: Agile techniques. By constantly prioritizing the most important features and limiting the amount of work under development, Agile software teams quickly learn what their customers really care about and can easily change direction knowing that they have not wasted significant time or money.

Product development teams typically comprise software engineers working closely with various types of hardware engineers. Leaders of these inter-disciplinary teams are keen to emulate the Agile approach and capture its benefits. But they’re often discouraged by misunderstandings about how they might apply Agile to product development.

Everyone has yet another PowerPoint deck but decks don’t get you funded. To stand out and win funding, you need the ability to show something quickly, real hardware or the application."

Christopher Marsh, VP Engineering, Comcast TPX Advanced Products Group
FOUR STEPS TO AGILE

To help product developers overcome those misunderstandings and move towards Agile approaches to delivery, we believe there are four key dimensions that they need to consider.

01 Empower the user voice

Product development has traditionally been ‘requirement driven’. An extensive document such as the Product Requirement Specification includes definitive statements outlining what the product ‘must do’. Requirements Engineers are there to understand the product intent and describe functionality to groups of developers who may never meet nor understand the needs of the product’s end user. A lengthy and expensive process, requirements solicitation and analysis consumes a significant amount of total project time, particularly as given the separation between developer and user, the specification must be complete and exacting.

An electronics manufacturer dedicated a significant percentage of total project time to product definition and planning. Given that both of these tasks are non-value add (but admittedly essential) the impact on Process Cycle efficiency was profound. Accenture’s value stream analysis of the product development process suggested that Process Cycle efficiency for this manufacturer was well below 10%. In simpler terms, viewed through a Lean lens, less than a tenth of all development work was actually considered value-add.
So how can the product specification process be improved? A critical step is to replace the document-heavy approach of traditional requirements management with lightweight statements voiced from the perspective of the end-user. These are then verified with valuable feedback captured from frequent product demos. This approach has the following benefits:

- The definition phase is significantly reduced, enabling the team to start on valuable development tasks much sooner.
- Development teams have the space to interpret the feature request and apply their own local context making it easier to understand the cost-benefit of tradeoffs.
- Teams can make changes more readily because they’re not subject to significant ‘sunk’ costs.
- Developers and users are encouraged to engage regularly, which further enhances product relevance and development efficiency.

This approach is best exemplified by the ‘user story’ concept Agile software teams employ. But product development teams have largely not adopted it. That’s mostly because they have concerns about requirements traceability or the perceived incomplete nature of specifications. But these objections are easy to address. Careful process and collaboration tool design can help maintain traceability. What’s more, the benefits of a shorter specification phase far outweigh the lack of fidelity some product developers fear.
Prioritize the **minimal** set of differentiating product features and **demonstrate** early and often

What makes a great product experience? It’s usually the result of just a few critical features perfectly executed. Those features are the very essence of a product. So it follows that when they’re inadequately realized they can jeopardize the success of an entire project. An Agile mindset dictates that modern product development teams should start work on just those key features and nothing more. Prioritizing a small subset of product functionality means teams can quickly demonstrate functionality, seek feedback from users and validate early assumptions of user preferences and technical feasibility.

This approach is the foundation for software development methodologies such as SCRUM. But product development teams are typically reluctant to work on a small subset of features, often citing interdependence concerns and the risks of over-prioritizing. And while those concerns are understandable, our experience suggests that if the risks associated with critical features are not explored and retired early, the consequences simply accumulate over the project cycle. The results of that are potentially disastrous.

“Building and testing the most important product features first sounds easy, but in reality is hard. The most important aspects of a product are often at odds with what an engineering or executive team is most excited about building. If you take a hard, honest look at what actually matters to end users and what differentiates your product, you can get much earlier feedback that you’re on the right track or not—even with hardware—to get a successful product out faster.”

**Jesse Dorogusker,**
Hardware Lead, Square

For example, a recent Mindtribe³ client sought help developing a stylus for use with a tablet computing device. Initial consultations with potential product users revealed that the drawing experience for users was the single most important feature.
If that could not be adequately realized, then there was no value in investing in other product features such as small form factor or long battery life. The Mindtribe team quickly prototyped several early models and thanks to frequent feedback from users, were able to rapidly evolve their original ideas on customer preference and technical feasibility. Once the core feature of a great drawing experience was confirmed, the team then moved to the next most important feature of the design and the project progressed in this way through to completion.

At Mindtribe, the tool we use to establish product priorities is the Product Nucleus.

To create a Product Nucleus, you start by establishing the user needs your product should address, in order of importance. Prioritization is key—without it, slight differences in the envisioned product embodiment are uncovered deep into development when change is costly.

The product definition then becomes the minimal set of features required to meet each need and the engineering team starts developing the most important features first.”

Steve Myers
CEO & Founder

Adopt fixed development cadences to align planning and integration across all teams

One of the best known aspects of Agile methods is the fixed development timebox or ‘sprint’. Software teams work in fixed-duration increments ranging from one to four weeks. During that time, developers will plan, develop, test and deliver a complete software feature (or story). Short increment working facilitates rapid feedback because Agile teams will always seek to deliver something demonstrable to end users rather than partially complete software components.
Product development teams, typically comprising both software and hardware engineers, find this approach challenging owing to the high cost of change for hardware. A software engineer can change a product with just a few source code edits. Hardware engineers might need to wait for lengthy updates to mechanical tooling and slow printed circuit board (PCB) fabrication cycles.

Rapid prototyping technologies can significantly reduce the cost and time of hardware updates. However, in our view a longer, fixed development timebox should be put in place in order to accommodate hardware teams. A field-proven solution for multi-disciplinary engineering teams is the dual timebox approach advocated by the Scaled Agile Framework (SAFe™). SAFe describes a short sprint ideal for software teams but aligns planning and integration with hardware teams through a longer timebox known as the Program Increment. The duration of the Program Increment is constant and a fixed integer multiple of the sprint duration, typically ranging from eight to twelve weeks. The benefits of such a synchronized, dual-speed approach are:

- Significantly lower project management overhead, as the classic project constraints of time and cost are made constant, meaning only scope needs to be actively managed.
- The longer time-box is ideal for hardware teams to develop prototypes for customer demonstration and feedback.
- The fixed relationship between sprint and Program Increment durations provides natural alignment points between teams running at different speeds.

Accenture helped an electronics manufacturer introduce the Scaled Agile Framework to their product development teams. Initially the engagement was directed at the software teams but as the benefits became more apparent, the method was rolled out to all engineering disciplines. Software teams sprinted on a two-week cadence while acoustics, electrical hardware and mechanical teams worked at the slower rate afforded by the program increment. Full system integrations were made no less frequently than every twelve weeks, so system issues were surfaced early. A year after the engagement was complete, client leadership reported that the number of issues found during the pre-launch test phase had reduced by over two thirds with a corresponding reduction in project duration and cost.
Invest in complex product models to **accelerate issue discovery** and decouple teams

To take the next step and become truly Agile, product developers need to decouple hardware and software teams so that they can execute concurrently but be confident that their integrated solutions will meet system-level specifications. There are many possible solutions to this challenge, but all feature simulation of hardware that, in effect, transforms rigid hardware into easy-to-change software. Four examples illustrate this:

01 At a global data storage technology company, highly specialized simulation engineers write C code to emulate the functionality of hard drive controller ASICS. The simulation code allows the firmware team to write and test code realizing the majority of the drive’s functionality long before the ASIC is delivered.

02 A common practice at many of our semiconductor clients is to make an extensive investment to emulate semiconductor functionality before tape-out using Field Programmable Gate Arrays devices. The resulting hardware emulators are used to confirm design assumptions and as an early test bed for device software.

03 As part of Accenture’s Industry X.0 practice, Pillar Technology has had over twenty years developing software for a wide range of industries. Pillar’s work in the automotive industry has resulted in an emulation solution called LOOP. LOOP allows software developers to replace, or mock, hardware with software versions that are consistent and manageable. This replacement allows for test scenarios specific to the client’s exact requirements to be developed in a controlled environment, allowing the software developer to progress without the need of a physical product—decreasing development time and increasing value.

04 The emergence of inexpensive development platforms such as Raspberry PI, enables product developers to quickly demonstrate system behaviors without investment in custom hardware.

Each of these examples shows a long-term view of product development efficiency as well as a significant investment in technology that enables faster, more flexible execution. While building the emulated environment will require significant investment, the rewards from early system issue discovery, high system confidence and reduced project management overhead will justify the expense.
Of course, to benefit from Agile principles, Product Development teams need to go beyond theory and put the principles into action. Below we show how we’ve helped clients make the move to Agile.

**MAKING AGILE WORK IN PRACTICE**

Consider a Systems Architect as backlog owner for an Agile Product Development Team.

Servant Leaders are the process authority of the team. Senior Engineers or Project Coordinators can fill this role.

Scaled Agile Framework and Accenture’s AutoScrum are both great examples of multi-speed development models.

Agile Product Development teams should be cross-functional including SW, HW, Mechanical where necessary.

Quality Gate criteria can be repurposed to create a Definition of Done for Product Development Stories.

Kanban works well for managing the work of HW teams.

Fix the HW Cycle to a multiple of the SW cycle to simplify planning and integration.

Continually invest in rapid prototyping and virtualization methods to shorten HW cycle.

Repeat cycle indefinitely.

Kanban works well for managing the work of HW teams.

Demonstrate the system or system models every system development cycle.

Re-order the backlog after each system demo.

Invite all stakeholders to regular demos.

Invite all stakeholders to regular demos.

Manage product features in a prioritized, itemized backlog rather than a requirements document.

SYSTEM DEVELOPMENT CYCLE

**SYSTEM DEMO**

+ System Demo

- SW SPRINT

- SW SPRINT

- SW SPRINT

- SW SPRINT

SYSTEM DEVELOPMENT CYCLE

**SYSTEM DEMO**

+ System Demo

- SW SPRINT

- SW SPRINT

- SW SPRINT

- SW SPRINT

System Demo

- SW SPRINT

- SW SPRINT

- SW SPRINT

- SW SPRINT

INVITE ALL STAKEHOLDERS TO REGULAR DEMOS

QUICK HIT TESTING OCCURS IN EVERY SPRINT

SYSTEM DEMO
With careful process and organizational design, Agile methods can work for hardware product development teams.

Agile methods are well established for software teams but application is much rarer at the hardware product level. In many cases, the overall product development project will run with a waterfall process, limiting Agile to the software developers. But this hybrid approach is a missed opportunity. Worse still, mixing waterfall and Agile methods burdens the software team with big upfront planning and requirements phases. These completely negate the primary benefits of the Agile method.

There is a better way. By following the four guiding principles we’ve set out, organizations can maximize the return from their development investments through improved product relevance, reduced project risk, shorter feedback loops and reduced time to issue discovery.
Accenture is a leading global professional services company, providing a broad range of services and solutions in strategy, consulting, digital, technology and operations. Combining unmatched experience and specialized skills across more than 40 industries and all business functions—underpinned by the world’s largest delivery network—Accenture works at the intersection of business and technology to help clients improve their performance and create sustainable value for their stakeholders. With 477,000 people serving clients in more than 120 countries, Accenture drives innovation to improve the way the world works and lives. Visit us at www.accenture.com.

References


3 Mindtribe, part of Accenture Industry X.0, is an engineering team that helps companies build innovative, connected hardware faster. https://mindtribe.com/


Copyright © 2019 Accenture.
All rights reserved.
Accenture and its logo are trademarks of Accenture.