

How to be an awesome agile developer

Turning habits into culture



BY DAN RADIGAN

BROWSE TOPICS

- Agile manifesto
- > Scrum
- > Kanban
- > Agile project management
- > Product Management
- > Agile at scale
- > **Software development**
 - Overview
 - **Developer**
 - Dev managers vs scrum
 - Technical debt
 - Testing
 - Incident response
 - Continuous integration
- > Design
- > The agile advantage
- DevOps
- > Agile Teams
- > Agile tutorials
- > About the Agile Coach
- All articles

Agile development isn't just about developers

Nobody wants to ship software with lots of bugs, performance issues, and low customer satisfaction. [Continuous integration](#) and [code reviews](#) help prevent this... but who has the time, right? Well, agile teams maketime.

Agile developers focus on sustainable development—not heroics. Sustainability is about good estimation, effective branching strategies for managing code, automated testing to protect quality, and continuous deployment to get fast feedback from users. Adopting sustainable development practices requires a discipline most of us aspire to—but often struggle to realize—as individuals. That's because nobody can go agile in a vacuum. The culture of the entire organization has to rally behind it. That means getting project leaders to buy into the notion that quality is more important than scope or schedule, which is often the hardest part of adopting agile.

But it's worth it! Developers get the freedom and accountability to develop software sustainably, while still maintaining a great relationship with the business. And the business gets to take a higher-quality product to market, which further reinforces that great relationship with engineering. What's more (and here's the best part), agile developers rarely face "death marches." When development falls behind schedule because maintaining high quality takes more effort than anticipated, the scope side of the iron triangle can flex to accommodate that reality—and nobody's weekend gets hijacked.

All software developers know the "iron triangle" of project management: scope, schedule, and quality. And most of us have been on projects where the scope was inflexible, the schedule got mangled, and development was overwhelmed by mounting [technical debt](#). Sometimes—just to add insult to injury—the end product wasn't even what the market wanted. That's a frustrating, and tragically familiar, place to be.

But fear not: there is good news.

With agile development, scope becomes the dynamic variable so teams can protect quality, build a vibrant development culture, and stay tightly coupled with the business. At Atlassian, agile is at the core of every development team (as well as many non-dev teams!). And for good reason.



Agile isn't just a set of ceremonies. It's a cultural and technical philosophy.

It empowers individuals with practices that build a solid technical foundation into their product and a culture of collaboration into their team. Developers on agile teams are more engaged, write better code, and have more fun.

Strong connections means a stronger product

Agile is about teamwork, which is no surprise since most software today is built by teams. Developers build strong relationships with product management, design, QA, and operations because writing sustainable code means staying connected to all facets of the project. Atlassian has seen huge benefits in code quality and developer satisfaction by

RELATED TUTORIAL

Learn scrum with Jira Software

[Try this tutorial →](#)

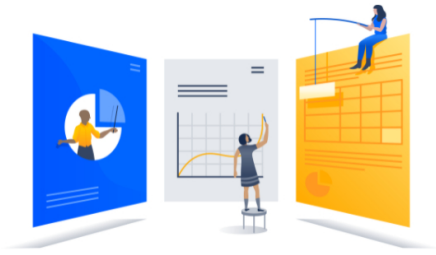
SUBSCRIBE

Sign up for more articles

Email

[Subscribe](#)

empowering developers to work directly with other parts of the business. Better code, less "thrash" (i.e. duplication of effort and/or conflicting streams of work), and more effective cross-functionalism are just a few of the benefits.



Mentoring is big, too. [Agile teams](#) cross-train each other to ensure knowledge of the code base is spread throughout the team. One way this is done is through code reviews, which not only protect quality, but cross-pollinate familiarity with the code across the team. Regardless of how the knowledge spreads, agile teams don't end up with critical-path developers who can't go on vacation because they're the only ones who understand a particular area of code. Because nobody wants to be that developer.

Agile developers can also work up and down their product's technology stack with greater ease than their waterfall counterparts because agile teams self-organize, giving members chances to pick up new skills. It's a fact that developers who deliver entire features—from UI to DB—take greater ownership of their code. At Atlassian, we cultivate full-stack developers because we believe in sharing knowledge across the team, and throughout the company.

Coding, culture, and making agile development awesome

Going agile is all about building a great development culture in your organization. Keep reading to learn more about effective [branching strategies](#), [automated test techniques](#), [continuous integration](#), and creating effective relationships with other parts of the business. The next articles will dig deeper into the specific changes thousands of developers have made during their transition to agile, and prospered because of it.

Agile development is a journey. And we're behind you every step of the way.

SHARE THIS ARTICLE



DAN RADIGAN

Agile has had a huge impact on me both professionally and personally as I've learned the best experiences are agile, both in code and in life. You'll often find me at the intersection of technology, photography, and motorcycling.




TUTORIAL

Learn scrum with Jira Software

A step-by-step guide on how to drive a scrum project, prioritize and organize your backlog into sprints, run the scrum ceremonies and more, all in Jira.

[Try this tutorial →](#)



ARTICLE

Development Managers vs. Scrum Masters

Learn how to be an effective agile manager as a Scrum Master or Development Manager. See how these two roles differ and work together effectively.

[Read this article →](#)

Agile Topics

Agile project management
Scrum
Kanban
Design

Software development
Product management
Teams
Agile at scale
DevOps

Sign up for more agile articles and tutorials.

Email

Subscribe



Up Next

Dev managers vs scrum masters →