

# How to deliver quality assurance at speed

Changing quality assurance to quality assistance

BY LAURA DALY

## 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

It is difficult to adapt traditional testing methods to an agile culture; teams feel forced to trade off the quality of their product for the speed of shipping.

To combat these issues teams at Atlassian pioneered a different approach to agile testing known as Quality Assistance. Instead of creating a separate test team held responsible for quality, a small team of Quality Assistance engineers evangelizes and coaches sustainable testing methods across the development team. Learn more about this transformation and how to:

- Create a culture of quality
- Push responsibility for testing back to developers
- Prevent bugs, not detect them



## Q & A

Read the Q&As from this presentation to learn more about how a team of 65 engineers build and rapidly ship a high-quality product with only six QA engineers.

**Q1: How long does it take to get a developer up to speed on this type of thinking?**

**A1:** It's harder to change the culture of a whole team than it is to transform individuals. It's taken us five years to get the Jira Software team to the level of quality mindset it has today, but it doesn't take each new developer very long to get up to speed. They quickly pick up the mindset from their fellow existing devs, and they soon pick up the testing skills via pairing and workshops. The hardest part is picking up all the knowledge of risks and the product. This can take years, but we mitigate this through knowledge-sharing in QA kickoffs and QA demos.

**Q2: Is there still a need for test cases, are those for regression/automated testing only?**

**A2:** Scripted manual test cases don't come into our strategy at all. If a test is just a 'check' – that is, a set of predefined steps and a defined assertion – we find it is more efficient and less error-prone to have it executed by a computer instead of a human. If a test is genuinely a test – requiring critical thinking, freedom to investigate and risk assessment – we find it is better to execute it as part of exploratory testing in order to include that freedom and intelligence in the testing.

**Q3: Developers are typically more expensive than testers. If we use developers as testers, is that not an inefficient use of budget/manpower?**

**A3:** Absolutely, using developers as testers to execute a separate testing step is expensive and wasteful of developer time. But having a separate testing step at all – even one executed by testers – is expensive and wasteful of developer time. Every time a story or bug is pushed back from testers to developers, it's not just a testing cost, it's a developer cost. By dropping the rejection rate from 100% to

## RELATED TUTORIAL

Learn scrum with Jira Software

[Try this tutorial →](#)

## SUBSCRIBE

Sign up for more articles

Email

email@example.com

[Subscribe](#)

4%, we've saved a lot of development time that was being wasted on reworking stories and fixing stupid bugs before release. We've saved the time spent on investigating, reporting, triaging, assessing, reproducing, and fixing internally-found bugs. And the code is designed from the ground up in a more testable way because the developers know they're the ones who will have to do the testing. Our DoTing (developer-on-testing) stage was an intermediate step along the path of pushing quality upstream, so that we could remove the separate testing step entirely. It was a temporary investment that has more than paid off.

**Q4: We have developers and QA testers in different time zones. Would this model only work in the same time zone? How do you work with remote teams?**

**A4:** We've done remote quality assistance with teams in Poland and Vietnam, with the QA engineer based in Australia. It's not as effective as having skilled QA onsite, as a big part of being a good Quality Assistance engineer is building a personal relationship with your devs. A remote QA engineer is easily cut out of the loop, and it's much harder to gauge the overall culture of the team. However, we were able to successfully run remote QA demos, QA kickoffs, and pairing sessions via video calls – just calling directly from the dev's machine to the QA's and sharing the screen.

**Q5: Are the QA notes on a story-per-story basis or do you build a knowledge base of QA notes? How do you deal with recurring risks?**

**A5:** QA notes are on a story-per-story basis, so it's usually the QA engineers that spot the patterns of recurring risks. This has become harder over the years as our Jira Software QA team has grown, as each QA engineer doesn't necessarily know what the others know. Until now, we've mitigated this with weekly knowledge-sharing meetings, and wiki pages where we keep track of common or surprising risks. We're getting to the point where this no longer scales. Right now we're working on a more structured knowledge base with a database of rules that are run over each commit. So, for example, if it sees you're using the User object in your Jira Software code, it would add a comment to the issue saying, "The User object can be null if the current user is anonymous, please make sure you're handling that correctly". This will help us get the knowledge out of QA heads and, in the best case scenario, may even replace the need for QA demos and kickoffs. That would be helpful!

Overview

Developer

Dev managers vs scrum masters

Git

Branching

Git branching video

Code reviews

Release

Stress free release

SHARE THIS ARTICLE



LAURA DALY

Laura is former Product Marketing Manager with experience on various product teams including Jira Software, Portfolio for Jira, and Bitbucket. When she is not writing about agile best practices you can find her in the mountains chasing storms or looking for the perfect berm.

**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

email@example.com

Subscribe



Up Next  
Technical debt →