

A survival guide

Do we still need a QA manager's role in agile organizations?

QA managers in agile environments face a conundrum: How do you understand your role when, in a strictly agile organization, it's not defined? QA management professionals know how to work in a traditional waterfall environment. Understanding exactly what's required as they move into agile projects is key to a QA manager's success. Here's a road map.

In pure agile organizations, quality is managed by development team leads and development testers within the team, not by someone outside of it. There's a good reason for that: when the team is responsible for and is measured by the quality of its code, delivering high-quality code (as opposed to just completing a user story) becomes a priority. Quality becomes an integral part of the team's planning, development, status meetings, and the definition of "done." Code quality established by measurable results becomes a target for the whole team. As a result, your team delivers better code.

But there's also a downside. [Transitioning to agile](#) is a process that takes considerable time and effort before quality can become a core competency. Development testers and developers are usually highly technical and therefore tend to focus on the technical aspects of quality assurance, such as test automation and tools. They can become disconnected from the more earthy aspects, such as the business drivers behind the code, integration, and working with customers.

QA is a distinct profession and not just something that you do along the way as you develop software. QA professionals need to have the proper background and to keep up to date with the latest tools, trends, technologies, and methodologies—something that somehow never makes it into a developer's schedule.

Lay the ground rules

While managers should give agile teams a great deal of independence and responsibility to manage themselves, the organization as a whole must still function as a coherent unit. You must set a few basic ground rules, such as what testing tools to use, which methodologies to apply, how to define defect statuses, how to classify the state of a build, and so on.

As a QA manager, I work with the R&D managers to define and agree on what standards to use, since those determine how we introduce methods to the teams, how we train them, and how we monitor progress and quality throughout the development process. These become the guidelines the R&D managers use to implement QA processes within their development teams.

Build the team

Each development team has its own area of expertise. One team may work on the user interface (UI), while another focuses on back-end services. Different fields of development require specialized knowledge, as do manual and automated testing. Test automation may be very different, depending on whether you're working on the user interface, REST API, or cascading style sheets implementation.

Your role as QA manager is to build the team wisely, with the right number of development testers and end-to-end QA architects, as well as to [define their roles](#) and responsibilities and how those roles work together. You should also ensure that development testers are properly equipped to test their team's particular development work.

Choose quality metrics

Before you can evaluate the quality of deliveries, you must establish the metrics you'll use to measure it. As QA manager, you should help the development teams define quality metrics. For example, how are defects classified? What constitutes a

"show stopper"? How is efficiency defined? What's the regression ratio? What caused the regression? What was the initial quality delivered?

As QA manager, I also take a macroscopic look at quality by examining the trends in metrics across all teams in my organization. This not only gives me a view of the quality of the organization's code as a whole but also helps me identify teams that might not be working effectively.

There are many reasons why teams underperform. They may not be following testing methodologies correctly, and initial code quality may not be good enough, resulting in a high number of defects. Additionally, insufficient code coverage in automated tests may cause regression defects. Whatever the reason, it's your job as QA manager to identify the issue, delicately drill down into the team's work practices, and bring them up to par with the rest of the organization.

Look at the big picture

In my organization, the development teams develop, test, and deliver user stories within their own domains, while it's up to the QA architects to perform cross-functional testing to ensure that everything works well together. They test interactions between different modules in a variety of environments and configurations to ensure the system works from end to end, without any integration, security, performance, or other issues.

Your role as QA manager is to reflect on overall system quality. When you find problems at the system level, you must act as a bridge between the [development testers, QA architects, and development teams](#). Because you receive input from all these teams and cross-system activities, you're uniquely positioned to see the big picture, reflect on the overall status of application quality, and identify bottlenecks.

Balance the load

Manage efforts to balance testing demands on your team internally. If there's suddenly a need for extra test cycles, the development team lead may allocate additional testers from within the team. As an agile team, all members should have some capability to run tests. In reality, however, this isn't always possible.

In these cases, the QA manager may move developers or development testers between teams to answer an immediate need for extra testing. This technique is known as "[swarming](#)." In my group, this often happens when we reach the feature freeze phase of programming, during which we conduct intensive regression testing to stabilize the product. At that point, I may involve more developers in testing and move development testers between groups.

Above all, your role as a QA manager in an agile organization should be to have a vision and set strategy. You should stay abreast of new developments in the QA domain, evaluate new tools, and learn new methods. You must implement QA processes within the group and constantly improve on those by defining activities, setting priorities, and balancing resources at group level. In this way, all teams in your group can work together to consistently produce the highest-quality code.