

# Strategy for agile development teams: with or without dedicated QA?

By Poul Skjørring Sørensen. December 2017

This is a question that has been raised regularly since the agile development has gained popularity. Today, while I work with agile teams and practitioners, this keeps coming up as a subject of choice to be made rather than a result of a deliberate strategy.

This short article presents strategies and consequences in both cases. Opinions presented below were carefully chosen based on voices of others' and own experience.

NO DEDICATED QA	WITH A DEDICATED QA
<p>Without a dedicated QA, the team must think of ways to fulfill its purpose using the skills and resources available. In this case the team is thought as not even having access to an external QA team or resources. Developers must learn how to take responsibility; not only for the code but for the end-user product result and its impact. Having built the wrong product or even the right product of poor quality the team is not to blame anyone else but themselves. Taking responsibility means to act on bad results. It could mean to take precautions such as testing early with the end user and splitting the delivery in smaller chunks that are demonstrated regularly and often to get feedback (as in iterative and incremental agile development).</p> <p>What the team might do? For example, practice TDD (a way of development rather than testing) that would help the team to build the code right. The team could also work closer with the domain experts and the real users on acceptance testing and finding obvious holes and lacks in functionality. The team might have a tester role shifting within the team, or played by the test-minded team members (even Scrum Masters, PO and business analysts can come into play).</p>	<p>In many agile teams, dedicated testers play a vital role of leading and promoting the quality assurance. The role itself if called QA Engineer or similar. The idea behind their actions is to bake the quality in to the product rather than by reactive testing after development. Even writing tests before the code might result in a wrong product, unusable by the end users. QA is also about to make sure the team are building the right product. QA is the user's lawyer, having the real-world use examples in mind when engaging in the team work. Having spent some time in understanding end-users dedicated QA engage in exploratory testing using that perspective. The findings might go beyond what the team focused on to deliver, but this does still matter because better understanding is eventually a precondition to delivering something useful.</p> <p>Although QA can facilitate TDD, they are familiar with methods and tools for testing that aren't strictly related to that. A dedicated QA have typically spent a lot of time thinking how to break things and can see patterns and even attack recipes that form the base of experience-based testing.</p>

Finally, the team might find itself aiming to reduce risk of delivering the product. This can be done by minimizing impact of incremental changes produced by the team: release to a smaller group of beta testers or a small percentage of actual users before everybody else, using feature toggles and fast product delivery cycles that allow to fix things fast as they break in production

A dedicated QA also spends time on improving requirements and making sure that there is a common understanding of all situations the team is building the product for. Eliminating misunderstandings and assumptions by a very early testing activity to make sure less bugs will be introduced due to the lack of the right knowledge

## Team composition strategies

There are more options than simply choosing between having a dedicated QA person on the team or not. External QA managers, test coaches and specialized testers might complement QA-less teams during shorter or longer periods of time to establish baseline of testing mindset, principles and practices. Consultancy services are typically used to get access to their expertise and experience, especially if these roles cannot justify full-time employment. Unfortunately, that might lead to a risk of losing the knowledge brought by external parties. This happens since a team's culture is a dynamic entity, influenced by context and trends inside and outside of the organization. An example of such trends might be changes to the company's overall strategy, leadership expectations (or a lack of such), trends on social and experience-sharing media and communities and even options of prominent team members. Ultimately, the strategy for quality assurance is determined by the team but the influence might come from anywhere.

One thing to remember is that teams, no matter how self-directed they are, do exist in organisations. Quality minded organisations seek to achieve a growing QA mindset and a stack of good practices, principles and values on their teams which can be a great source of influence. Therefore, the choice is not about having a dedicated quality-person or not; it is about what organisations plan and do to produce quality products within their context and limits. Seeking to achieve and make the quality certain is a sign of long-term investment in the business.

## Sources

Content of this comparison is inspired by the ideas from the following sources:

"Dedicate Tester in an Agile Team" conversation:

<https://groups.yahoo.com/neo/groups/scrumdevelopment/conversations/topics/37701?threaded=1&m=e&var=1&tidx=1>

"Dedicated tester on an Agile team" article at <https://www.infoq.com/news/2009/04/dedicated-tester-agile-team>

"Specification by Example: How Successful Teams Deliver the Right Software 1st Edition" by Gojko Adzic, ISBN-13: 978-1617290084

"Quality is free – The Art of Making the Quality Certain" by Philip B. Crosby ISBN-13: 978-0451621290