

How to Build a Culture of Quality for Faster, More Confident Development

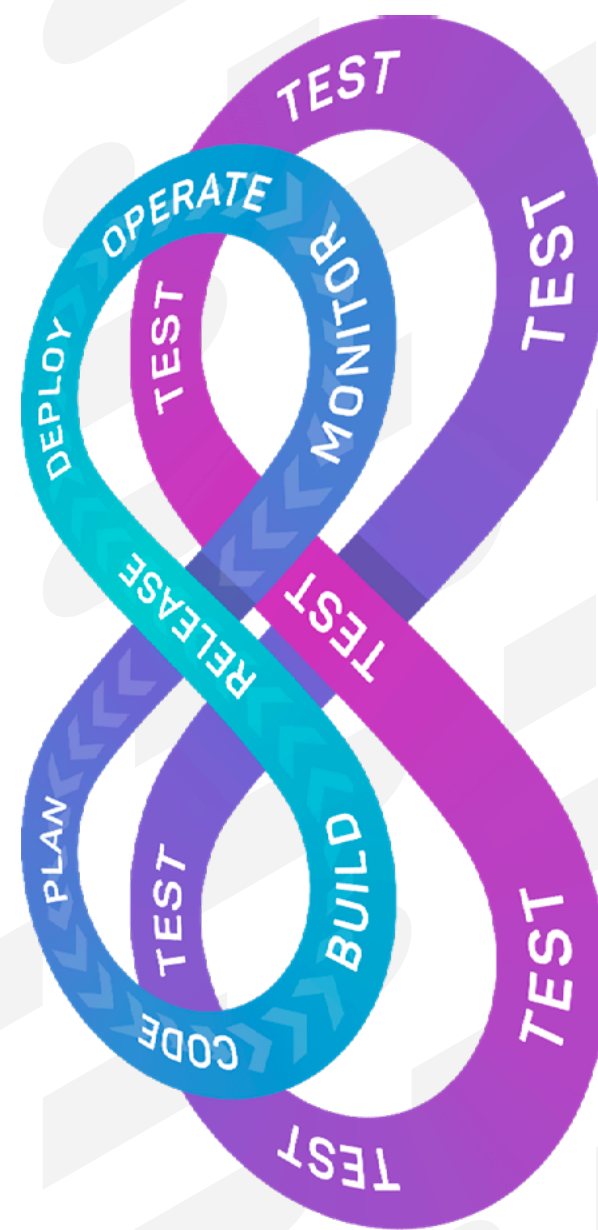


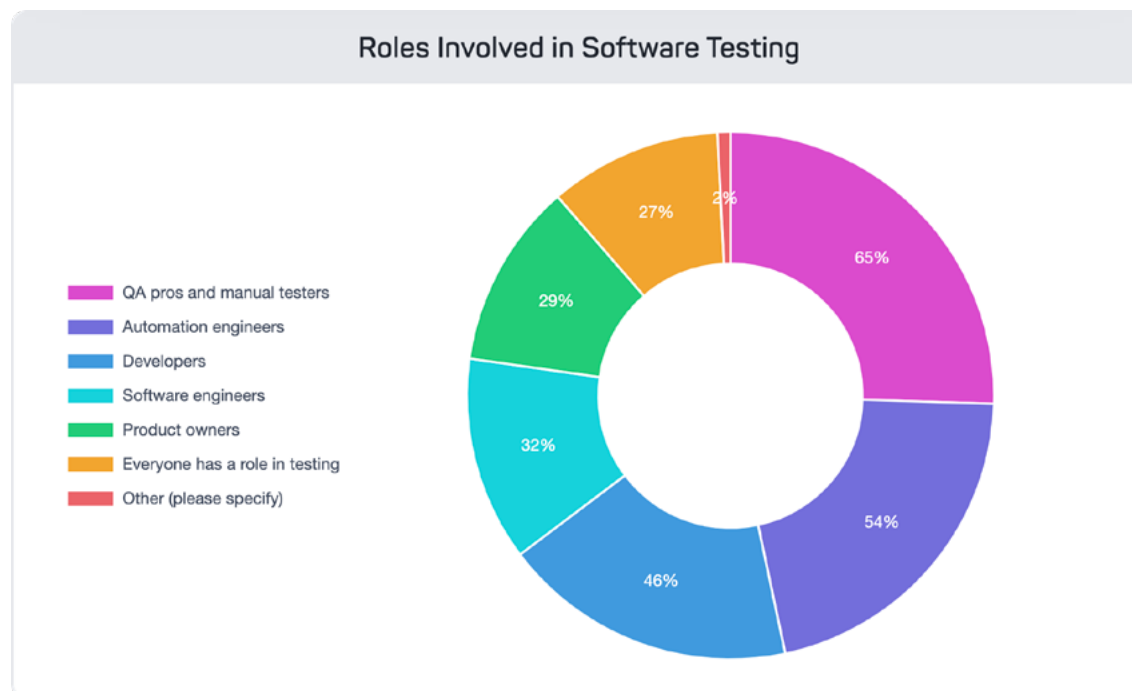
01	The Three Critical Elements of a Culture of Quality	07
02	People	09
03	Processes	13
04	Technology	17
05	Adopting and Scaling a Culture of Quality Transformation	20

Software development organizations are facing a seemingly impossible mandate: accelerate deployment frequency and continuously improve the customer experience.

Traditionally, faster development has been associated with a ‘fail fast’ mentality. Companies essentially bet that consumers would tolerate buggy or slow user experiences as long as it meant they had the most innovative product possible. But that mentality no longer withstands scrutiny, as consumers now value convenience and customer service above all else. Deloitte [found](#) that 78% of consumers place a higher premium on convenience than they did before 2020, making it one of the most important factors when making a purchase decision. Combined with the fact that [73% of all consumers](#) say that the CX plays an important role in their purchasing decisions, the pressure to deliver seamless experiences at record speeds is at an all-time high.

Enter quality engineering, a discipline that integrates software testing throughout the development lifecycle to deliver positive user experiences that help organizations satisfy, retain, and acquire new customers. Like DevOps, quality engineering isn’t a final destination. Instead, both strategies capture a state where software development organizations can continuously innovate and improve with fast, iterative deployments. DevOps ensures that teams can deliver new features or products faster, and quality engineering ensures they can do so without risking defects or product performance. Since most teams are still building their DevOps practice, quality assurance is well-positioned to build a complementary quality engineering practice that enforces best practices through a culture of quality.

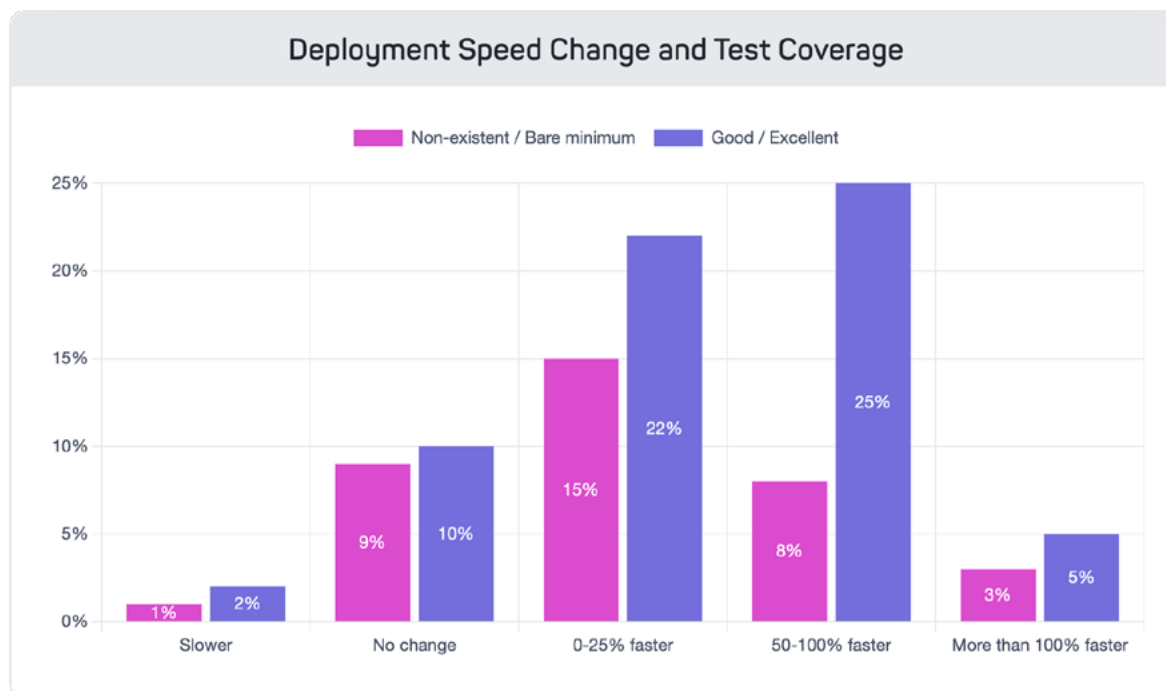




SOFTWARE TESTING: A SHARED RESPONSIBILITY

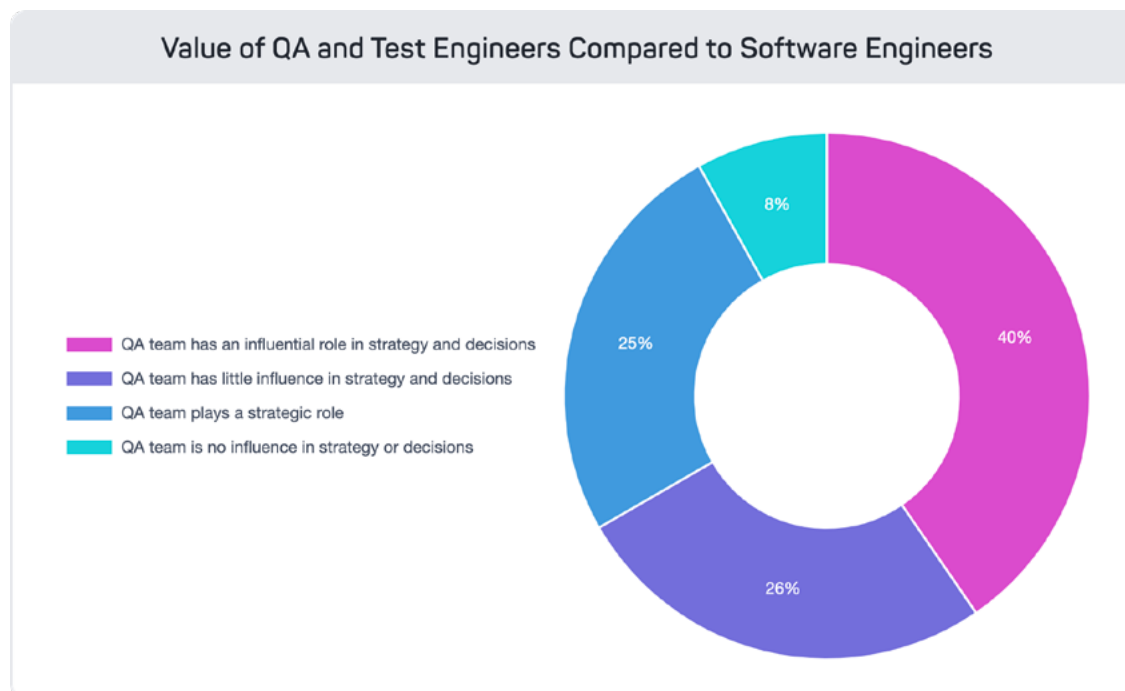
A key part of both DevOps and quality engineering: collaboration. The few teams that consider themselves full DevOps were most likely to have a strong culture of quality across their entire organization, making it easier to test, fix, and ship new products. According to mabl's 2022 Testing in DevOps Report, more roles are engaging in software testing across the SDLC. As teams establish a culture of quality, they're seeing developers take on more unit testing and business stakeholders create tests for newly-identified user journeys in the application.

Quality engineering teams can move into a leadership role that involves testing in staging and production, high-impact manual testing, managing overall testing strategy, and overseeing testing done by other team members. When entire teams collaborate on product quality, software testing is no longer a bottleneck in the development process and teams can ask bigger questions about quality, including non-functional requirements that improve the customer experience.



This theory is being proven in real-time by leading teams evolving their DevOps practices and CI/CD pipelines holistically within a culture of quality. Looking at the differentiators for teams that made the biggest gains in deployment frequency this year, quality comes to the forefront.

As seen in the above chart, teams that accelerated deployment frequency by 50-100% were 2.5 times more likely to have good or excellent test coverage, a key indicator of a culture of quality. Teams can move faster when they have a strong quality engineering practice that imbues the entire company with confidence in their product. That impact explains why more QA teams are taking on an influential role that encompasses strategic decision-making with the software development organization.



But as much as an organization may recognize the value of their quality and software testing team, rebuilding their development pipelines to maximize their impact is another task altogether. This guide will share the people, processes, and technologies necessary for building a sustainable culture of quality that unlocks DevOps success and higher customer satisfaction. You'll learn how to break down testing silos, make your case to leadership, and establish effective feedback loops that fix bugs fast through our expert tips and real-world examples.



01

The Three Critical Elements of a Culture of Quality

We define a strong culture of quality as one where team members, including developers, QA, and business stakeholders contribute to the creation, execution, and maintenance of tests across the development pipeline. This is a culture where the team understands the importance of quality, and actively works to contribute to processes and activities that optimize application quality.

The impact of a great culture of quality?

More

- ↑ Team efficiency
- ↑ Job satisfaction
- ↑ Product velocity
- ↑ Data-driven decisions
- ↑ Collaboration

Less

- ↓ Time spent on re-work
- ↓ Blame shifting
- ↓ Delayed releases
- ↓ Ambiguity
- ↓ Silos

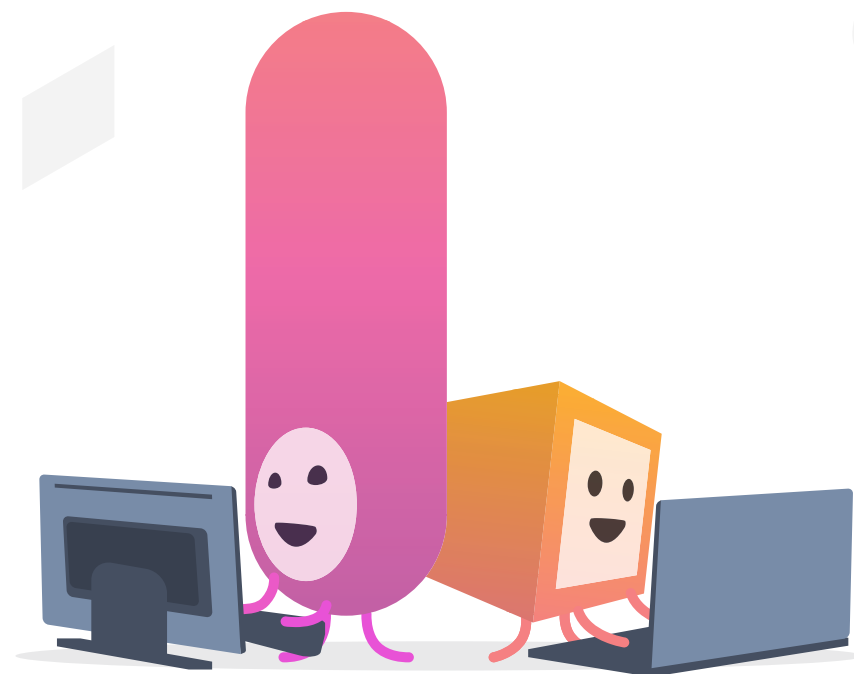
The foundations of a strong culture of quality are people, processes, and technology.

People: Consider the role each team member (in and outside the QA organization) plays in software quality, and how they collaborate with each other.

Processes: Assess the processes you have in place for testing activities, oversight, and collaboration.

Technology: What systems or software does your organization have that enable good testing practices?

Quality engineering teams should explore each of these culture pillars to understand where their teams are mature, or where there's an opportunity for improvement. The following sections will help you assess your current organization and build a plan to optimize your QA practices.





02

People

How to Break Down Silos and Build Careers to Support Your Culture

Team members who are collaborative, understand their role in testing activities, and feel they have learning opportunities will have the greatest impact on quality.

Breaking Down Silos

To truly embed quality into the development pipeline and culture, QA teams need to make their case to leadership. Many top-level stakeholders have goals that are closely tied to quality, such as time-to-delivery for new features, the number of customer complaints, and even company revenue growth. With the right metrics, QA can make a powerful case directly to these leaders and give quality a bigger seat in development planning, even if their current practices are heavily siloed.

Consider how your team can gather data that demonstrates the impact of testing on overall organization goals, then start identifying ways to make your case. Many QA teams find success in running small pilot projects specifically designed to help establish best practices for testing and quality. For example, if an organization is consistently delivering new features behind schedule, QA can partner with an engineering team to shift testing to the left for a single feature, then compare how quickly the new product was delivered, how much time was spent on reworking defective code, and how many bugs were shipped to production. By piloting new ways of working on a small scale, QA teams can build a more comprehensive (and more compelling) understanding of what testing and quality engineering practices that make an impact on high-priority issues that matter to leadership.

Everyone benefits from increased quality activities:

- Less re-work
- Faster feedback
- Improved coding skills
- More time on feature work
- Expanded test coverage
- Faster product velocity
- Fewer support requests
- Higher CX scores

When working with individuals on the development and product teams, it's essential to speak their language. Building an understanding of the pain each role might feel from a lack of attention to quality will help you identify opportunities to expand testing and quality impact.

Enabling Team Members to Grow Professionally

Participating in testing activities provide opportunities for folks across software functions to grow in their careers.

Let's look at opportunities across software development roles:

- With the availability of low-code test automation solutions, manual testers, who have a unique understanding of how testing impacts the user experience, can start automating manual test cases that augment manual testing efforts and participate in more exploratory testing.
- **Automation engineers** have the opportunity to advise developers or product owners on how and when to automate tests in the pipeline or start contributing to automating testing for new features.
- To build a culture of quality, **developers** can improve their coding skills by getting fast feedback on their code before merging code. Low-code solutions also make it faster to build end-to-end tests, which bring developers closer to the user experience and make it easier for them to participate in testing.
- As issues are identified, or new features ship, **product owners** or **business stakeholders** can start to contribute to building tests and monitoring quality metrics. By having more team members involved in testing, everyone will feel the responsibility of building quality software.

These activities can help you build a culture of quality across your team.

- Set goals and KPIs regularly
- Monitor goals in shared dashboard
- Paired work between QA and developers
- Exploratory testing sessions
- Encourage code and test reviews
- Highlight wins publicly, big and small

The goal of building a culture of quality is to maximize the impact of QA by elevating testing as an integral part of building good software and establishing better development practices. QA can serve as a center of excellence for quality while building a culture that everyone contributes to.

Case Study: Aligning with Company KPIs

SmugMug wanted to deploy any time, all the time. But doing so meant adopting a new testing strategy that involved their entire engineering organization.

Challenge: Manual testing limited the impact of testing across SmugMug's highly dynamic customer experiences, hindering efforts to expand testing and potentially introducing user-facing bugs as deployments accelerated.

Solution: Connecting testing to reduced customer support tickets enabled the QA team to showcase their value to leadership. SmugMug understands that they are fundamental to the livelihoods of their customers, especially professional photographers who have built virtual showrooms on the platform. By connecting testing to user experience metrics, QA was able to build buy-in around a culture of quality.

The SmugMug logo, featuring the word "SmugMug" in a bold, black, sans-serif font, followed by a green smiley face icon.



03

Processes

Set Your Team Up for Success

The slow pace of change was ranked as the top DevOps obstacle for the second year in a row in the [2022 Testing in DevOps Report](#). Other cultural concerns, such as a lack of leadership or internal politics, dominate the most common issues companies face when improving their development practices. Effective processes that promote communication, optimize workflows, and boost confidence are critical to sustainably and scalably overcoming these challenges and creating a culture of quality.

Processes are the nuts and bolts of a larger cultural change

To ensure that their quality transformation sticks, QA leaders need to clearly define testing processes on their teams. These processes should support your broader goals, such as enabling product velocity or improving the user experience. Some questions that effective processes answer:

- What tests are required when merging or shipping code?
- Who will build those tests, and who will execute and monitor the results of the tests?
- Are different people responsible for testing at different stages of the pipeline? In the testing pyramid?
- Who will manage the regression suite and monitor results?
- What happens when a bug is identified?
- What are the SLAs for resolving bugs at each severity level?
- What is being tested? What is the best way to document test cases?

Creating a shared understanding of quality

QA should evolve into quality coaches as your team's quality engineering practices mature. To do so, the team needs to be able to communicate process changes effectively - and see that change through to ensure that it's adopted by the wider organization.

First, many QA leaders have had success testing process changes with 1-2 team members, or a single team. Gaining buy-in from an engaged developer or business stakeholder can help you understand how process changes affect their workflows and give you time to gather feedback before rolling out changes to the broader team or teams.

Second, you'll need to serve as the leader of your new processes and roll them out to your team. People learn in different ways. Consider documenting the process (literally, write it down), but take time to demo it in a team setting, create a video or checklist, or host a lunch and learn. Try teaching team members in different ways to see what best resonates.

Create a feedback loop that fosters collaboration and communication

As you roll out changes and get more people involved in testing activities, the next step is building a continuous feedback loop. Though every team is different, real-world QA teams from startups to global enterprises have found successes with these tactics:

- **Prioritize testing activities in sprints:** When testing is visible from the earliest stages of development, everyone has a better understanding of how testing can accelerate delivery cycles and improve product quality.
- **Create internal training programs:** This equips team members with the knowledge and motivation to pursue common goals that prioritize the CX throughout the SDLC.
- **Offer QA office hours:** Giving developers and product owners dedicated time to improve their testing knowledge builds buy-in for quality engineering. Allow team members to seek out collaborative time for learning and knowledge sharing.
- **Create shared Slack/Teams channels:** Shared internal communication channels make it easy for team members to stay informed on test results, current testing activities, and any questions that arise.

- **Host friendly competitions:** Take gamification to the next level by hosting competitions that foster collaboration and introduce new ways to learn testing tools. Who can create the best tests? Or, who can create X number of tests in X amount of time? Contests are a great way to get team members engaged without the pressure of catching bugs mid-sprint.
- **Bug hunting sessions:** Similarly, bug hunts help team members explore how different issues can be caught by different testing strategies. The goal is to find as many bugs as you can - what tests will reign supreme? Your team will see how they can make their own QA practice more effective and build stronger relationships with this fun, informative activity.

A culture of quality spans collaboration, testing, product improvements, and reporting. To help team members across the SDLC confidently contribute to product quality, QA needs to lead activities that allow for learning and communication. The process of change may be slow, but shared activities can help teams sustain momentum as they grow their quality engineering practice.

Case Study: Building Processes

Chewy transformed their software testing strategy as part of their customer-centric ethos. Along the way, they enabled their development organization to embrace a new quality engineering practice.

Challenge: One quality engineer was dividing her time across four teams and manually testing functionality across three different sites for five different browsers on desktop and mobile. To accelerate development cycles, Chewy needed more developers contributing to quality.

Solution: Low-code test automation reduced the effort needed to create and run tests, making it easier for developers to participate in testing across the organization. To start, Chewy began nurturing a culture of quality within each individual team before scaling to another product team. Director of Engineering Russell Schwager shared:

"One of the things we've learned is, interest is high, but adoption is inherently slower. Teams are set in their ways, and there's an emotional impact to change. Some of the engineers were a little bit resistant to getting up and running on this. It's important to think about how you roll things out emotionally. People's roles do change, and changing the way things have always operated takes some effort. Continue to think big, and take a long-term view on it."

Quality processes that were proven and refined by their team, Chewy was able to successfully scale their culture of quality.

The Chewy logo, featuring the word "chewy" in a bold, blue, rounded, lowercase font.



04

Technology

Enabling Your People and Processes

The Testing in DevOps Report found that 72% of development professionals aren't hindered by technological capabilities on their DevOps journey. But your toolstack is critical to building effective processes and enabling your team. Once your team and processes are determined, you should audit your existing toolset or techstack to understand if it enables or hinders your teams' ability to collaborate on quality.

To determine whether you have the right quality engineering tools, here are some important questions to ask:

- Are team members working with disparate tools?
- Is there a unified platform that everyone can leverage?
- Are current tools easily accessible, yet secure?
- Do these tools integrate or speak to one another?
- Can they automate routine tasks?
- Can data be extracted from these tools to make informed, data-driven decisions?

Ultimately, your techstack should provide access to everyone on the team to prevent silos, be easy-to-use for team members with varying degrees of experience, and enable communication and collaboration among team members.

A traditional QA techstack includes software for:

- User behavior tracking
- Issue tracking
- CI/CD
- Test automation
- Communication
- Business intelligence

If you find that your current technology hinders the team's ability to build a culture of quality, start building a requirements document. You'll want to document the capabilities, integrations, and licensing needs of a new tool before you start researching vendors. Once you've narrowed your search down to a handful of vendors that meet your requirements, run POCs (proof of concepts) with each candidate. Gather feedback from your team, and ensure the vendor can support your long-term quality goals.

As you adopt new technologies, take a similar approach to rolling out new processes. Test implementations with one team member or a single team, build a feedback loop, and slowly roll out the technology to the broader team.

Case Study: Adopting New Test Automation Software

Stack Overflow improved quality collaboration across QA and development teams with test automation software that made sharing information easier.

Challenge: As a small team responsible for maintaining a 13-year-old codebase and building out a new product, Stack Overflow needed a testing solution that enabled their developers to continuously improve quality without wasting valuable engineering hours.

Solution: A test automation platform that supported more efficient collaboration processes. Software Engineer Meredith Hatzenbeuhler explained:

"I've been surprised by how much value we've gotten out of things like screenshots, test steps, network information, the DOM, et cetera. That's helped us collaborate with other teams. For example, if an SRE needs to step in, I can simply give them mabl credentials so they can log in, and look at the test. And they'll have exactly what they need. Whereas before, there would probably be back and forth discussions since we wouldn't always have such a detailed record with other types of automated testing. That loop of solving problems has definitely become much more efficient."

A tool designed for collaboration enhanced how Stack Overflow can respond to failed tests, enhancing their culture of quality and accelerating development cycles.



Adopting and Scaling a Culture of Quality Transformation

Once the core pillars of people, process, and technology are in place, it's time to start adopting and planning the journey to a culture of quality. Remember, a true transformation won't happen in a day, and many people will need time to adapt to the new ways of working. Janet Bracewell, Senior Engineering Manager, QA at SmugMug shared some practical advice for maintaining momentum on the quality journey:

"It's going to be bumpy. It just will be. Start small, pick one thing, like a section of code or an area of your platform, and begin building your quality strategy around that. Let the developers see that success so they can come to you already invested in quality transformation. If you gain some of that buy-in, then you can lead them by the hand and provide extra support. Always take your whole team along with you."

In order to build internal investment in a culture of quality, it helps to establish quick, simple ways for those outside QA to get involved. Many teams begin with automated end-to-end tests that run in the background every time developers hit "control + s." This makes headless UI testing an automatic function whenever a developer saves their work, making it easy to start shifting testing to the left and building a culture of quality.

Establishing a baseline for quality through expanded automated testing also enables quality teams to measure and share progress along the quality journey. Even small gains, such as increasing test coverage by a few percentage points, can make a significant impact as more bugs are caught in pre-production. Using technology to enable and document change helps keep people invested and processes standardized. Though your team may not be ready for fully integrated testing across the SLDC yet, begin establishing processes and solutions that can scale with your goals.

Attached is a workbook designed to help you evaluate your current quality tools and processes as you set the stage for transformation. Use it as a practical guide to help establish goals and timelines that will enable your team to innovate faster and with confidence.

WORKBOOK

How to Build a Culture of Quality



Development teams need to prioritize quality. Quality coaches can help.

This workbook is designed to help you evaluate your current culture around quality and identify necessary steps for improvement. The stages we describe may seem simplistic, but they are based on our experiences working with real quality teams at varying stages of DevOps adoption and automation maturity.

Though every team and product is different, we hope that this workbook shares useful ideas to help you build a culture of quality in your organization!

Additional resources can be found at mabl.com

About mabl:

mabl is the enterprise SaaS leader of intelligent, low-code test automation that empowers high-velocity software teams to embed end-to-end tests into the entire development lifecycle.

[Learn more at mabl.com](https://mabl.com)

Let's set the stage.

Describe your current team and processes.

Describe your product team's structure.

How are priorities set on a regular basis? Who decides what to test?

**Are people's talents and expertise fully utilized
(i.e. are people spending more time on rote work or problem solving)?**

Are processes documented in a shared space?

Does the team have a shared understanding of quality?

Are processes documented in a shared space?

Evaluate team collaboration.

Describe your current pains.

Signals that team collaboration could be improved

Obstacles preventing collaboration at a higher level

Challenges solved by improved collaboration

Evaluate yourself on the scale

1

QA team is responsible for building, executing, and maintaining all tests.
KPIs around quality may be tracked by leadership but not shared with the the broader team.

2

Developers test inconsistently early in the pipeline, but testing activities are siloed from QA.
QA still creates and maintains a majority of tests.
Quality KPIs are tracked and understood by the QA team.

3

Developers are testing early in the pipeline with some input from QA.
QA is still the sole owner of testing late in the pipeline.
Leadership shares and tracks quality KPIs with broader team, but testing impact is not widely understood.

4

Broader scope of team members create tests throughout the pipeline, and QA leans in to review or iterate.
QA may still own some test creation or maintenance (E2E, exploratory testing).
Testing activities and conversations happen in team settings.
Quantifiable goals are set around quality and the impact of testing is broadly understood.

5

Quality is so ingrained into culture and roles that QA acts as quality advisors, rather than gatekeepers.
The entire product team participates in all testing activities, including E2E or exploratory testing.
QA contributes to overall product and quality strategy.
QA and product teams are completely aligned on shared, measurable goals.

My team is currently:

What could I or my team do in the near to mid-future to make incremental steps to the culture we want to build?

Evaluate processes.

Describe your current pains.

Does the team have a shared understanding of quality?

Are processes documented in a shared space?

Is the team following set processes for testing each release or regressions properly? Why?

Is the team spending more time on process than progress?

Evaluate yourself on the scale

1

If testing processes exist, they are not well-documented or followed by the team.

The team has a difficult time adapting to changes in product or business strategy.

Processes are a roadblock to work.

2

Processes may be built for some test cases, or phases of the pipeline, but is not consistently followed by the team.

QA makes an effort to optimize processes, but product team has a difficult time adapting.

3

Testing processes for new code and regression suites is fully defined by QA.

Product team is adopting processes, but feedback is missing.

QA is connected to business strategy and user behavior but has difficulty translating it into processes.

4

Testing processes are built in collaboration with the broader team and QA incorporates regularly feedback.

QA drives the conversation around user behavior changes and can translate it into new testing requirements.

5

Processes are agile, driven by feedback from the team, and are optimized to make the most of everyone's skills.

Team can understand and adopt new processes with minimal interruption as strategy and user behavior change.

Team values improvement and has a shared understanding of quality.

My team is currently:

What could I or my team do in the near to mid-future to make incremental steps to the culture we want to build?

Evaluate tools and technology.

Describe your current pains.

Is the team working from disparate tools?

Do your tools integrate or speak to each other?

Do your tools help automate routine tasks?

Can everyone access (and use) the tools you have today?

Can you extract actionable data from your tools?

Evaluate yourself on the scale

1

Requirements for tech stack are not clearly defined. Tech may be used inconsistently, or is only useful to some of the team.

Output and data from tools is not actionable or not used.

2

Requirements for technology have been defined, but current tech stack is not aligned to the requirements.

Effort to make tools more accessible are blocked by usability or excessive technical requirements.

Current tech is still disparate or used inconsistently.

3

Centralized technology stack that fits requirements defined by broader team is in place.

Technology is accessible to the entire team, and they are trained on proper use.

Tech stack is not properly integrated with each other. Some inefficiencies still exist.

4

Point solutions are integrated with each other in a way that enables your processes and collaboration efforts.

Tech stack is reviewed regularly to validate value.

Actionable data is extracted from the tools that drive decision-making at leadership level.

5

Fully integrated and automated pipeline with a quality strategy at the core.

Tech is enabling more efficient work, and can trigger actions or communication as prescribed by QA.

The team can leverage actionable data and insights from your tech, and provide input on usage.

My team is currently:

What could I or my team do in the near to mid-future to make incremental steps to the culture we want to build?

Putting it all together.

Use this sheet to prioritize the themes and opportunities most important to you by customizing the themes and opportunities and the timeframe.

Review your scoring:

Collaboration Score:

Processes Score:

Tech Score:

Theme	Opportunities for Change	Month 1	Month 2	Month 3	Month 4	Month 5	Month 6



About mabl

mabl is the leading intelligent, low-code test automation solution built for teams who are building a culture of quality. It's the only SaaS solution that makes it easy for anyone to contribute to test automation, and tightly integrate those tests into the development lifecycle. mabl's unified platform makes creating, executing, and maintaining reliable browser, API, accessibility, and mobile web tests easier than ever.

By combining human and machine intelligence, users are more confident in test execution and can identify and resolve bugs long before they reach production. That's why customer-centric brands like Charles Schwab, jetBlue, Dollar Shave Club, Stack Overflow, and many others rely on mabl to create the digital experiences their customers demand.

[Try mabl free for 14 days.](#)