

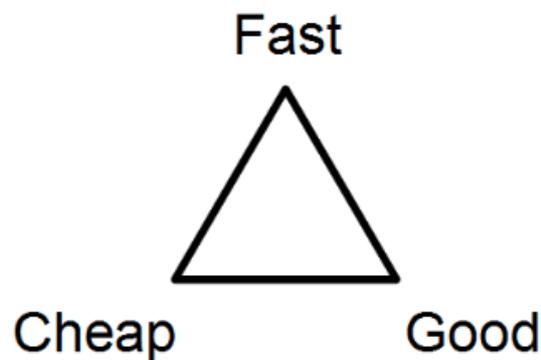
2019 Diffblue Developer Survey

What's wrong with software speed, quality, and cost?

The Cost, Quality, Speed Trade-Off

Development teams are simultaneously benefitted and plagued by high demand and low supply. On the one hand, they have great job security. On the other hand, they have far too much to do. **Gartner predicted** years ago that demand for application development would grow at least five times faster than the capacity to deliver it through 2021, and so far, this has played out accurately.

Even for companies with development teams, consistently and quickly producing high-quality code often costs more than the budget will allow. This dilemma is the perfect example of a classic project management problem: How do you choose between delivering at high speed, high quality, and low cost if you can only pick two?



An engineering/project management triangle. Pick 2 of 3!

[Image Source](#)

Companies today have to respond to consumer demands faster than the competition, so one of these priorities—speed—has been pre-selected. This leaves a choice between quality and cost, but the two can't be separated: A minimum level of quality is required to be acceptable to users, and straying below this level will inevitably end up costing companies more.

The Move Towards DevOps

For many companies that want to streamline the software development lifecycle (SDLC), regain control over their release cycles and increase deployment frequency, the answer is to

embrace DevOps, which encourages simplifying processes and automating as much as possible. But some areas of the SDLC have remained relatively untouched by automation—one of which is testing, the area most likely to impact quality. An excerpt from the 2019 book [Enterprise Continuous Testing](#) sums up the situation nicely:

“DevOps is all about removing the barriers to delivering innovative software faster. Yet, as other aspects of the delivery process are streamlined and accelerated, testing consistently emerges as the greatest limiting factor. A recent GitLab survey that targeted developers and engineers found that testing is responsible for more delays than any other part of the development process.... Even teams with palpable test automation wins face roadblocks: They can’t create and execute realistic tests fast enough or frequently enough.”

Our own team’s anecdotal experience has suggested that this is an area of stress for developers, and one that deserves a solution.

Inspiration for the Study

In conducting this study, we wanted to get the developer perspective to find out what the people who actually build the software think about their companies’ software quality problems, and how they would solve these challenges based on their first-hand experience with them. We put together a survey to ask 300 developers about their company’s code quality, their organizations’ expectations of them, their daily schedules, and their pain points.

The questions we sought to answer included:

1. What is the quality of the code currently produced in large organizations?
2. Where do you see quality being held back?
3. How do manual processes impact your work?
4. How do you think quality can be improved efficiently?
5. How do you think quality can be improved at a reasonable cost?

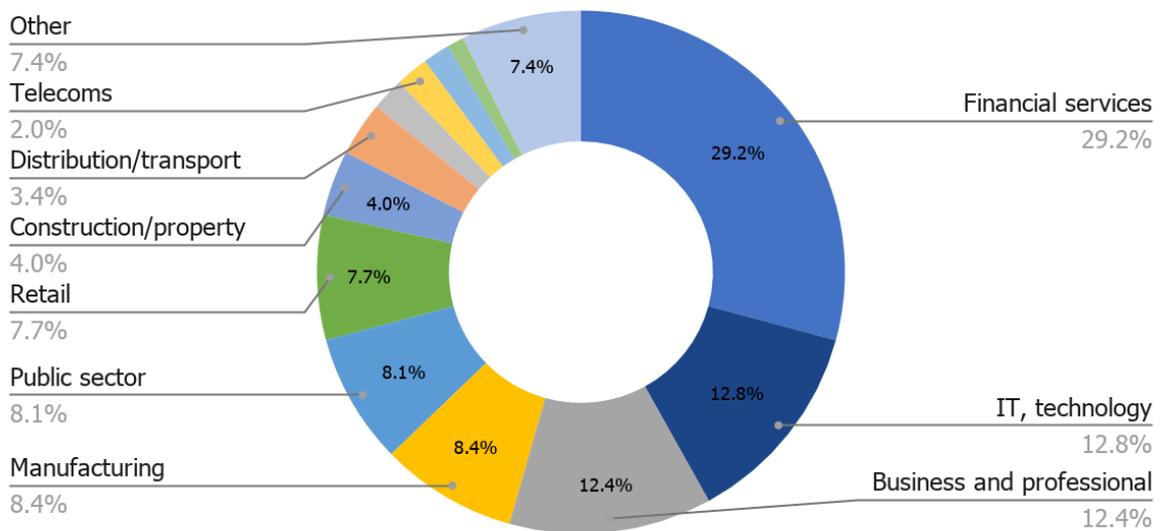
Methodology

Our 300 respondents were surveyed online by the research agency [Vanson Bourne](#): 200 in the US and 100 in the UK, working in software development, application development and DevOps, with job titles below the executive level. The survey included a mixture of Likert scales, multiple choice questions, and open-ended responses.

Demographics

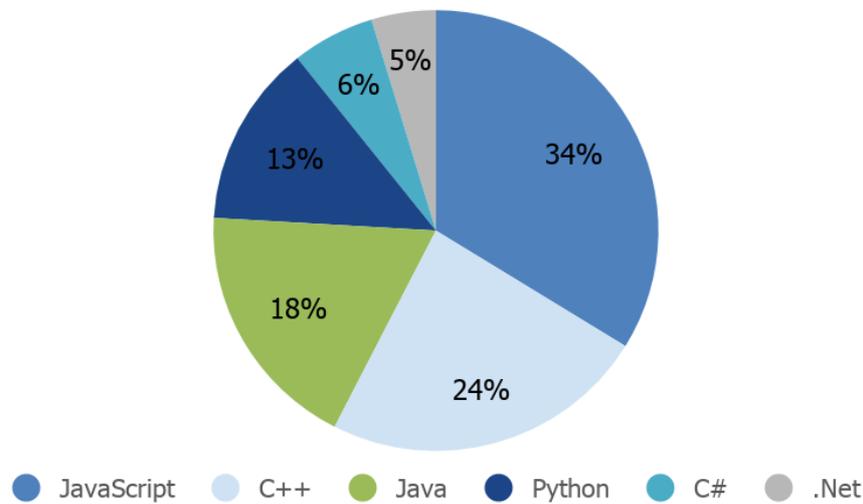
All participants work at companies with at least 500 employees. Sectors include financial services (29%), IT/Technology (13%), business and professional services (12%), and various others (see the graph for the full breakdown). The average annual salary of our respondents (before tax) is £67,624.

Participant Demographics: Sector Breakdown



Participants work an average of 43 hours/week and spend 50% of their time writing code (on average). The three most popular primary coding languages are JavaScript (34%), C++ (24%), and Java (18%).

Participant Demographics: Primary Programming Language



Key Findings and Presentation of Data

The findings from this study have been grouped into the following six themes based on the trends we identified in the responses:

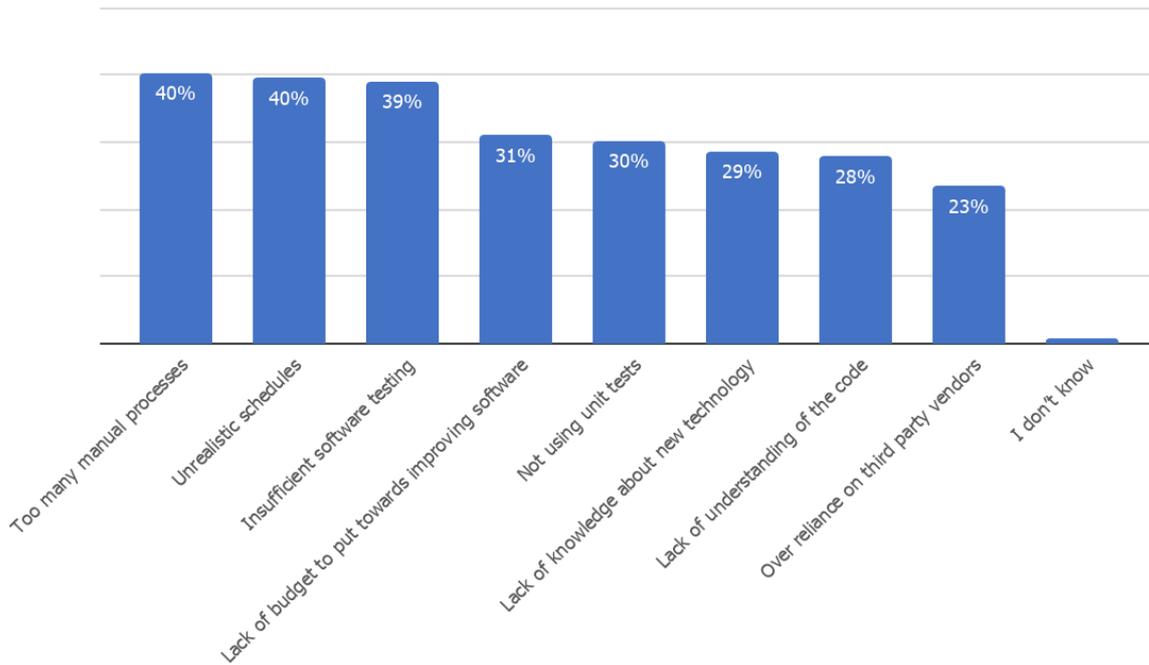
1. Developers think their companies' software quality could be improved
2. Current attempts at achieving code quality come at a high cost
3. Developers feel they cannot currently meet the quality targets their organizations impose on them
4. Developers agree about what is needed to produce better quality code
5. Developers agree that unit tests can facilitate DevOps
6. Targeted automation can improve developer job satisfaction

Theme 1: Developers think their companies' software quality could be improved

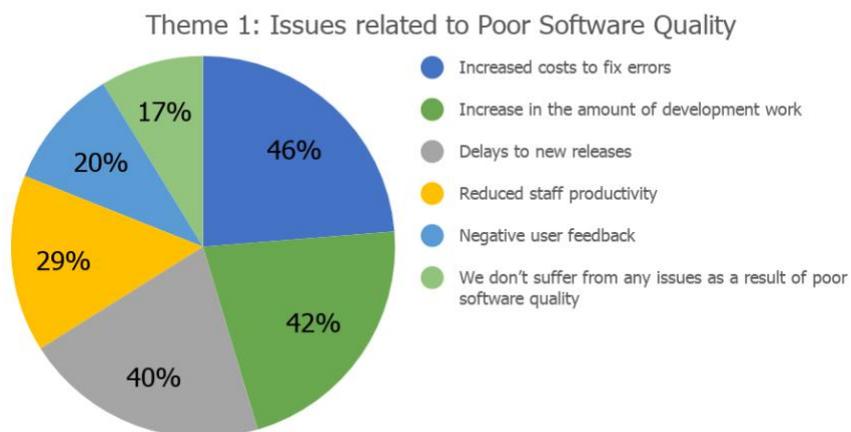
First, we wanted to get a feel for how developers feel about their companies' current code quality. The result: 83% of the software developers we surveyed agree that their company's software quality could be improved.

When asked which factors contribute to poor software quality, 40% of developers attributed it to manual processes, 40% cited unrealistic schedules, and 39% said it was due to insufficient software testing.

Theme 1: Factors contributing to poor software quality



When asked which issues their organizations suffers from as a result of poor software quality, the top responses were: Increased cost to fix errors (selected by 46%), increase in the amount of development work (42%), delays to new releases (40%) and reduced staff productivity (29%). See the graph for the full breakdown:



When a release is delayed due to poor quality, the delay takes an average of 3 weeks.

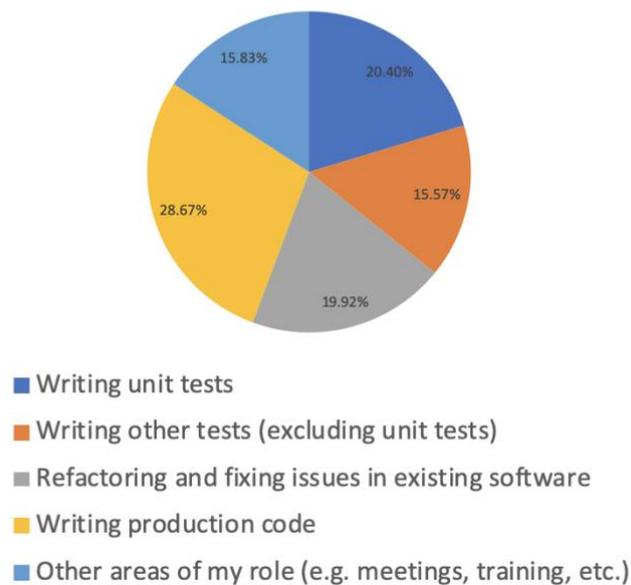
Theme 2: Current attempts at achieving code quality come at a high cost

Unit tests form the most basic and fundamental level of testing: these numerous, small tests cover individual units of code, to make sure each part functions as intended. Unit tests are important to code quality, but because they are more effective the more of them you have, writing enough of them to provide a high level of protection against regressions can take quite a lot of time.

We were interested in finding out exactly how much time is spent on unit testing, and the results from this study indicate that developers spend about 20% of their time writing unit tests, at an average cost of £14,287 per developer per year. With 45 developers employed (on average) at participants' companies, the average annual cost of unit testing for just one organization is approximately £643,000. Developers spend 15% of their time writing all other types of tests.

See the graph for a full breakdown of how developers spend their time:

Theme 2: What proportion of your time is spent on each of the following areas of your role?



Theme 3: Developers feel they cannot currently meet the quality targets their organizations impose on them

Organizations often set “code coverage” targets for developers to achieve, which requires writing unit tests to ensure software quality. The data revealed that the average code coverage target for the organizations the study participants work at is 63%.

In addition to spending 35% of their time just writing tests, developers are also under pressure to deliver on new product development, which takes 29% of their time. Perhaps as a result, almost half (48%) of developers reported that they have found it difficult to achieve the unit testing coverage set by their organizations, and 42% of developers have skipped writing unit tests in order to speed up new feature development.

Theme 4: Developers agree about what is needed to produce better quality code

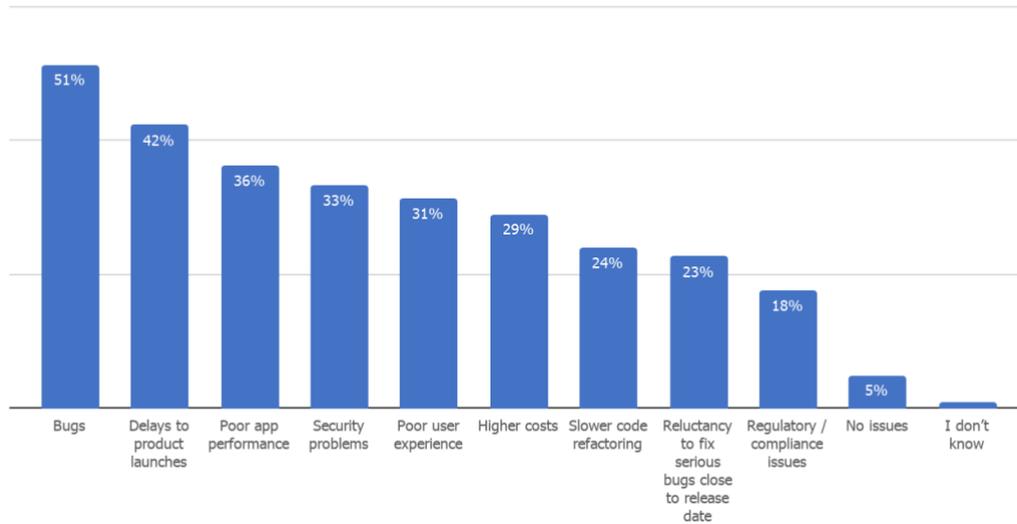
When it comes to the benefits of the unit tests that developers are instructed to write, however, they tend to be in agreement that unit tests improve software quality (90% agree) and speed up code maintenance (95% agree), making maintenance 40% faster on average.

Most developers also think unit tests make it easier to work with code that was created a long time ago or written by somebody else. 75% of respondents agreed with the statement “It’s faster to rewrite legacy code that has unit tests,” and 77% agreed with the statement “It’s easier to review peer’s code to identify errors when the code has unit tests.”

The issues developers would most expect to see from releasing software applications without unit tests are bugs (selected by 51%), delays to product launches (42%), poor application performance (36%) and security problems (33%).

See the graph below for the remaining responses:

Theme 4: Issues When Releasing Software without Unit Tests



Theme 5: Developers agree that unit tests can facilitate DevOps

According to our respondents, unit testing and DevOps go hand-in-hand. When asked to rank the top three benefits to their organization of using unit testing as part of software development, 55% of developers chose the response "It helps us to achieve our DevOps goals," and 50% chose "It makes the end-to-end development process faster"—the top responses behind "It improves the quality of code" (chosen by 63%).

See the graph for a full breakdown:

Theme 5: Benefits of Unit Testing

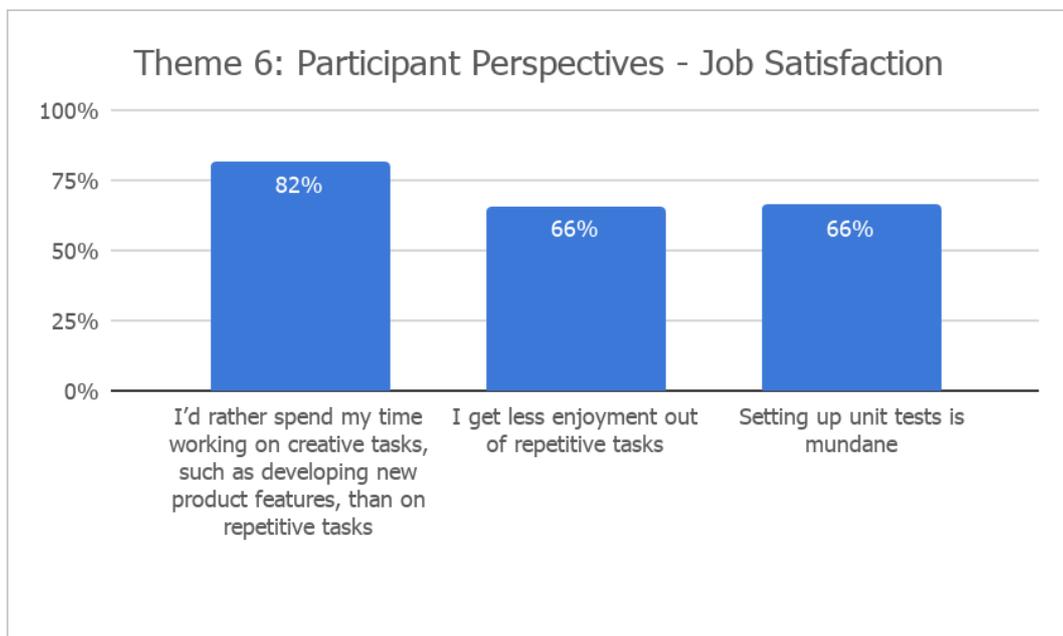


Theme 6: Targeted automation can improve developer job satisfaction

73% of developers in the UK and 53% in the US report that they could be more satisfied in their jobs, for (apparently) a variety of reasons. Of the factors that influence job satisfaction, those revealed in our study linked back to two key areas of improvement developers would most like to see: 1) having achievable targets 2) having automation tools to help them reach these targets, reduce the number of repetitive manual tasks they are expected to do, and increase the time they can spend on creative work.

In terms of having achievable targets, 87% agree that having realistic targets is important to job satisfaction, and 91% say it's important to have the right tools. When it comes to writing unit tests to meet internal coverage targets, 66% agree that unit test setup is mundane and 39% of developers wish they didn't have to write unit tests at all.

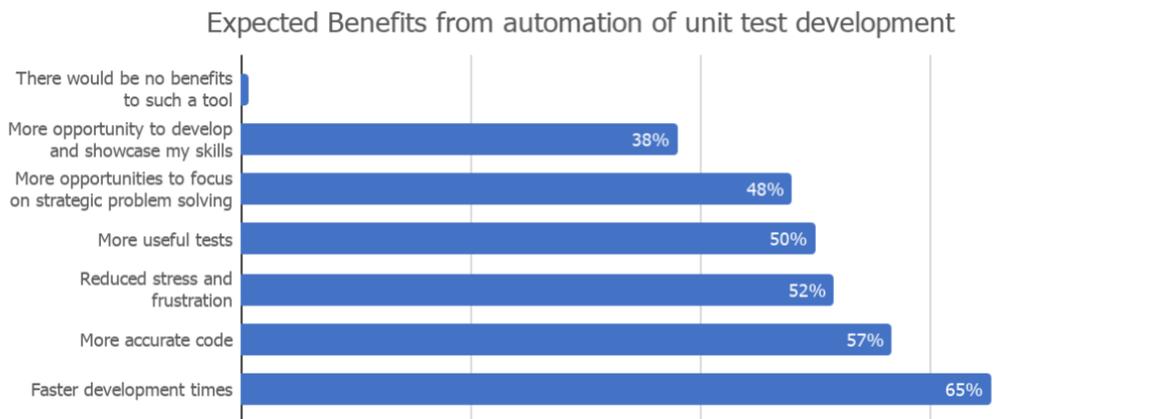
66% of developers agreed that they get less enjoyment out of repetitive tasks, and 86% say the availability of automation for repetitive tasks is a factor in their job satisfaction. 86% also say it's important to have time to work on creative problem-solving tasks, and 82% agreed that they would rather spend their time on creative tasks, such as developing new product features, than on repetitive tasks.



For 84% of respondents, their organization's willingness to adopt new technology, like AI, is important to job satisfaction. 68% of our respondents said their company has partially or fully adopted AI tools that assist with software development. When asked which tasks

developers would most like to see automated, findings bugs was ranked first by 33%, and software testing (including writing unit tests) was the first choice for 28%.

Developers would expect automating the development of unit tests to result in the following benefits: faster development times (selected by 65%), more accurate code (57%), and reduced stress and frustration (52%), among other benefits (see the graph for full breakdown).



Discussion and Final Takeaways

Circling back to the research questions we started with, the answers developers gave are consistent and not that surprising: Currently, code quality has room for improvement at most organizations, and is primarily being held back by a combination of unrealistic schedules, too many manual processes and insufficient testing. Manual processes are also contributing to high costs, and replacing some of these tasks with automation could improve quality efficiently and cost effectively.

Some of the details that came out in the results, however, are more interesting and unexpected. Perhaps the most salient is that 42% of developers say they have skipped unit testing to speed up new feature development, suggesting a disconnect between management expectations and what developers can actually deliver in a given timeframe with their existing tools.

While management and developers are in agreement about the importance of catching regressions with unit tests, developers are under a lot of pressure to produce software quickly; when they have to make a choice between writing unit tests and writing new features, many choose the latter. Even though most companies have code coverage targets, it would appear that unit tests are not prioritized when new feature development deadlines

come up, which could be in part due to developers' own preferences for working on creative tasks over repetitive ones.

One thing is clear: Software quality is too important to rely on the hope that overstretched developers will find time in their schedules for a time-intensive, repetitive task that 39% of them wish they didn't have to do. Bugs caught late in the software development lifecycle are **substantially more expensive to fix** than those that are prevented or caught in the earliest stages, and in many cases late-stage bugs are only identified after they've negatively impacted the customer.

Developers Need Help

So what can be done? Our respondents have given a clue. Developers need help from CIOs, CTOs and management to unit test efficiently, or costs and delivery timelines will suffer. 97% of the developers in this study said they would be interested in having the opportunity to use a tool that could help guide them to develop unit tests.

Many developers say eliminating perceived mundane and/or repetitive tasks would increase satisfaction at work. With skilled developers in shortage globally, attracting top talent is key to being able to compete, and developers are less likely to enjoy working at companies that do not modernize.

Notably, twice as many respondents in the US (compared to the UK) answered that their company has "fully adopted" an AI tool for software development, reflecting a trend among respondents from the US to voice greater demand for automation tools, to report higher adoption of automation tools in their companies—and to have higher overall workplace satisfaction than their counterparts in the UK.

Automate Everything

Going beyond workplace satisfaction, providing automation support for developers is a key aspect of DevOps that can't be done halfway. Companies that aim to fully embrace DevOps practices need to find a way to actually live up to the mantra "Automate Everything" and eliminate the gaps that require manual effort and cause development teams pain. Currently, the largest of these gaps is in testing—which happens to fall at the intersection of cost, speed, and quality—and unit testing is an area of particular pain for companies that want to achieve DevOps targets. After all, if you aren't continuously creating tests, then your CI isn't actually continuous.

Looking to the Future

Stay tuned for our upcoming survey on IT decision-makers' perspectives on the challenges of DevOps, the "2020 Diffblue DevOps Survey: What can be done to improve DevOps?" which will be conducted next spring. To stay up-to-date in the meantime, you can sign up for our newsletter and check out our [blog](#) for more regular articles on how to improve software development.

About Diffblue

Diffblue was founded in 2016 as a spin-out of the University of Oxford. The company's technology uses AI to create tests that improve the quality of code at the earliest stages of the software development lifecycle, making truly continuous integration possible for Java development. Headquartered in Oxford, Diffblue is funded by Goldman Sachs and Oxford Sciences Innovation.

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