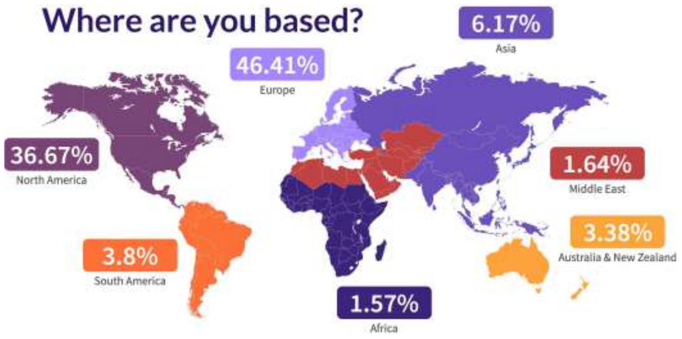
Bridging the divide between developers and management

The 2018 GitLab Global Developer Report (http://bit.ly/global_dev_report) highlighted several interesting findings about the state of adoption of DevOps. Specifically, it showcased the different perspectives between software delivery practitioners (developers) and their managers, where managers were often more optimistic than the developers. It is in these differences where we can infer insight about DevOps adoption to accelerate software delivery and improve business results. The reality is that developers and their managers often see the world differently because they face different challenges.

How the survey was run

In late 2017, we surveyed 5,296 software professionals from around the world to learn from IT organizations adopting continuous improvement practices and seamless automation across their software development lifecycle. We believed they would have happier, more collaborative, and well-functioning teams who are better positioned to meet their goals and objectives. The majority of respondents identify as a software developer or engineer and work for small- to medium-sized businesses (SMB) in the hardware, services, and SaaS industries.



Managers are more positive and optimistic

While managers and developers tend to agree on many topics, the managers are generally more optimistic about things. They agree about the importance of collaboration and they agree that DevOps saves time. But, when you look closely, 81 percent of managers expected DevOps to save time, yet only 65 percent of developers shared their view. And 39 percent of developers report deploying on demand, but 47 percent of managers more optimistically think they are able to deploy on demand.

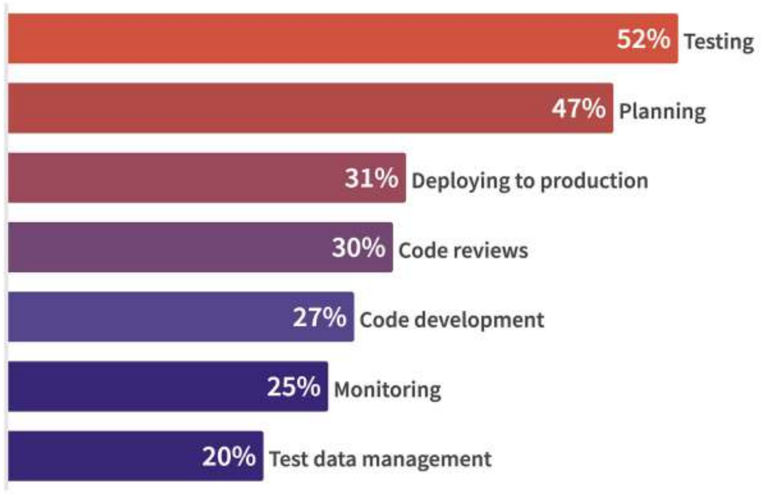
It's not surprising to see this disconnect between managers and the people they lead. Leaders and managers are often focused on the bigger picture, and don't face the day-to-day details their developers do. Managers believe that DevOps is saving them time, but the reality is that developers are the ones who have to address the challenges integrating disparate tools, resolving issues between siloed processes and teams.

DevOps saves time in the development process	**81%** Managers agree	**65%** Developers agree

Top delay: testing and planning

Developers reported that testing causes the most delays, while their managers highlighted challenges in the planning phase. When you think about complete DevOps, it is about the end-to-end flow of new ideas through the entire delivery value chain. The fact that developers and managers perceive different bottlenecks is no surprise. Typically, developers are not focused on the details of portfolio planning and governance, nor are managers and leaders in the weeds of the development process to sense the issues in testing and QA. It is critically important for teams to consider their full value chain and collectively focus on removing the most significant bottlenecks. DevOps is about continuously improving the flow and developers and managers need to work together. They are better together.

Where in the development process do you encounter the most delays?



Other 5%
We don't encounter delays 4%
Don't know 1%

n=3,281

Higher-Performing vs Lower-Performing

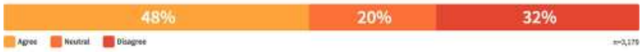
When we looked at the difference between what we called "Lower-Performing" and "Higher-Performing" organizations, there were a few key differences that emerged.

- Project goals and requirements more clear in higher-performing teams (68% to 48%)
- Higher-performing teams feel more set up to succeed (78% to 69%)
- Higher-performing teams have 'realistic deadlines' (70% to 57%)
- Higher-performing teams rarely need to sacrifice quality to meet deadlines (55% to 42%)

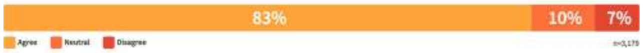
Here you can see the comparison between higher-performing teams and lower-performing teams where on every dimension, the higher-performing teams reported a better experience.

lower-performing vs higher-performing

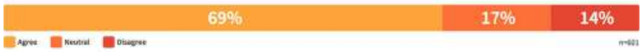
Project expectations and requirements are set up front.



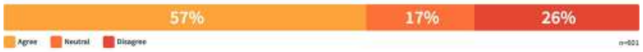
My ideas and opinions are valued.



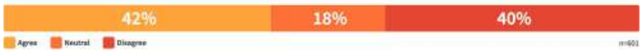
I feel set up to succeed.



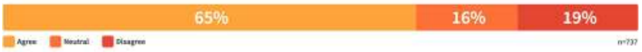
I'm given realistic deadlines.



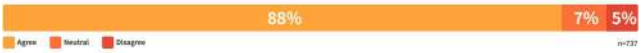
I rarely need to sacrifice quality to meet a deadline.



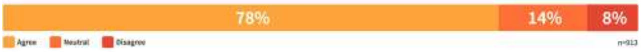
Project expectations and requirements are set up front.



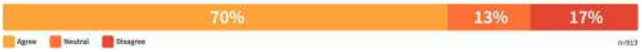
My ideas and opinions are valued.



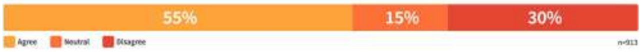
I feel set up to succeed.



I'm given realistic deadlines.



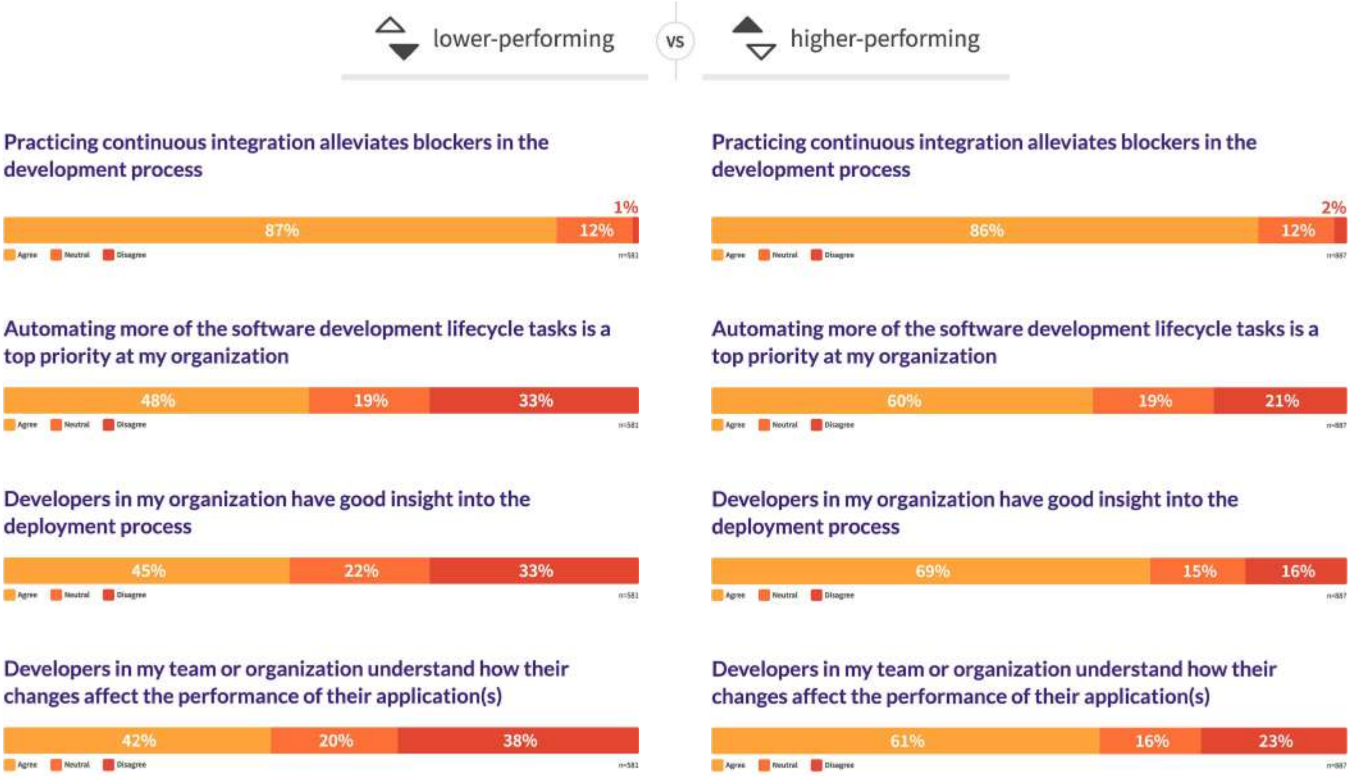
I rarely need to sacrifice quality to meet a deadline.



Teams that are able to release faster are also experiencing better results across their entire SDLC. Embracing DevOps means solving so many issues and sources of friction in delivery, one would expect the higher-performing teams to be more favorable across the board. The goal for everyone may not be to release multiple times per day, but rather to release at the speed of the business. The challenge all IT teams face is addressing the sources of friction that create bottlenecks in the delivery value chain.

Continuous integration is key

Another difference between higher-performing and lower-performing teams was their perception of continuous integration. Overall, our survey showed that 87 percent of the respondents agreed that continuous integration alleviates blockers in the development process (up from last year's survey - 77 percent). When you look at specific questions such as the importance of automating the software development lifecycle, the difference between higher-performing (60 percent) and lower-performing (48 percent) is striking. On one hand, there is consensus about the importance of continuous integration, but when you look deeper, the reality is that lower-performing teams are not getting the benefit of the automation and pipelines that continuous integration would expect.



Continuous integration is about automating as much of the toolchain as possible and here the difference between teams that are moving really fast and those that are not (yet) is telling. Faster teams clearly value better automation and also visibility into the pipeline.

Recommendations:

1. Managers and developers can make DevOps a reality, if they address their fragmented tool chains. The disconnects in the organization, tools and processes is a very real source of friction that leads to higher costs, lower quality and lower productivity.
2. Automation of the development tool chain enables teams to eliminate manual errors, automate the testing and validation of changes and give the team fast feedback into the quality of their work. Build and improve your continuous integration and continuous delivery pipelines, because this will help you to remove friction, reduce errors and accelerate delivery.
3. Focus on speeding up end-to-end cycle time. Strive to be able to release faster. When you look at the full value chain (from idea to production), and then address constraints, you are often addressing the issues in your organization, your process, and your tool chain that are cause of your delivery challenges.

About GitLab

PLAN » CREATE » VERIFY » PACKAGE » RELEASE » CONFIGURE » MONITOR

GitLab is the first single application for all stages of the DevOps lifecycle. Only GitLab enables Concurrent DevOps, unlocking organizations from the constraints of today's toolchain.

GitLab provides unmatched visibility, radical new levels of efficiency, and comprehensive governance to significantly compress the time between planning a change and monitoring its effect. This makes the software lifecycle 3 times faster, radically improving the speed of business.

GitLab and Concurrent DevOps collapses cycle times by driving higher efficiency across all stages of the software development lifecycle. For the first time, Product, Development, QA, Security, and Operations teams can work concurrently in a single application. There's no need to integrate and synchronize tools, or waste time waiting for handoffs. Everyone contributes to a single conversation, instead of managing multiple threads across disparate tools. And only GitLab gives teams complete visibility across the lifecycle with a single, trusted source of data to simplify troubleshooting and drive accountability. All activity is governed by consistent controls, making security and compliance first-class citizens instead of an afterthought.

Built on open source, GitLab leverages the community contributions of thousands of developers and millions of users to continuously deliver new DevOps innovations. More than 100,000 organizations, including Ticketmaster, ING, NASDAQ, Alibaba, Sony, and Intel trust GitLab to deliver great software at new speeds.

Learn how GitLab can help you unlock the full potential of DevOps (https://about.gitlab.com/)