

If software matters, then the efficiency of software delivery matters.

The IBM Software Delivery Report 2023



Legal and logistics

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT.

IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

This report is intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. IBM shall not be responsible for any loss whatsoever sustained by any organization or person who relies on this publication.

The data used in this report may be derived from third-party sources and IBM does not independently verify, validate, or audit such data. The results from the use of such data are provided on an “as is” basis and IBM makes no representations or warranties, express or implied.

To learn more about IBM DevOps Automation and how it can help your organization to improve software delivery efficiency, we invite you to join the IBM DevOps Automation community.

<https://ibm.biz/DevOpsAutomation>



Contents

Introduction	4
Key recommendations	8
Current trends and perceptions on software delivery efficiency	10
▶ Developers want more automation	12
▶ Quality is an essential part of the software delivery experience	14
▶ Most DevOps toolchains are perceived as being inefficient	16
▶ Organizations can minimize the environmental impact of software delivery	18
▶ Teams want help identifying efficiency blockers	20
▶ Maturing software delivery is a journey that requires data	22
About the study	24
How to measure and improve efficiency	26
Thank you	28

Introduction

IBM has been at the forefront of innovation in the software development and delivery sector. A pioneer in software engineering, IBM DevOps Automation is the main advanced DevOps automation provider for enterprises.

Continually leveraging real world experiences and challenges, IBM provides automation solutions that transform DevOps data into actionable intelligence and increases the use of automation in the areas of development, testing, delivery, value stream management, release activity, and configuration management to improve the efficiency of software development and delivery.

From launching UML in 1995 to applying Artificial intelligence (AI) to boost the productivity of developers in 2023, IBM has a history of creating automation capabilities that have advanced the ability of software delivery teams to develop and deliver reliable software as fast as possible.

For over 30 years, we have been inventing and providing capabilities that help our clients develop and deliver software. This unique experience and leadership mean that IBM has leading automation capabilities to empower teams and partnerships to achieve the most efficient software delivery. With support for both business and technical stakeholders, IBM DevOps Automation is creating a better software delivery experience for everyone and bringing software delivery to the core of delivering business value.



“Our clients are innovators and IBM DevOps Automation is enabling them to deliver reliable and secure software as fast as possible. Efficient software delivery is essential for modernizing the applications that run our businesses, and that people and researchers depend on daily. Software matters, therefore, the efficiency of software delivery matters and that is why IBM DevOps Automation is so important for our clients.”

Sebastian Krause, Senior Vice President and Chief Revenue Officer, IBM



Welcome to the first edition of the IBM Software Delivery Report. This report provides a unique look at the trends and perceptions of software delivery from the perspective of the people who are developing and delivering software every day.

Our survey polled software development and delivery individuals for their perceptions about how efficient software delivery is in their organization. The people polled are a mix of IBM clients, professionals, and partners. Most have over ten years' experience in software development. We are grateful for a 97% completion rate. The people who completed the survey are the very people delivering critically important software applications and services which make or break an organization's ability to compete. It is a privilege to have had the opportunity to engage with these individuals, and now to provide a summary of their unique viewpoint into the true state of enterprise software delivery efficiency.

The mission of IBM DevOps Automation is to create a better software delivery experience. Our IBM DevOps Automation subject matter experts have analysed survey results and added best practice advice. Our goal is that we can all learn real world solutions to the real-world problems revealed by the software development professionals who took the survey.

This is not a report based on a technology-first perspective; it is one that looks at software delivery from the ground up. It is not about DevOps hype; it is about DevOps data. Delivering secure and reliable software as fast as possible is how software makes a difference to an organization, to our society and to our world. That is why this report is so important. The findings speak for the practitioners who are hands on delivering the software.

This report would not have been possible without the contributions of software delivery engineers from around the world. If you are one of the many hands-on practitioners who contributed, we thank you for your time and contribution to helping us and our readers get an insider perspective into the state of software delivery efficiency. You have voiced the potential for real change, and we want to help.



Madhu Kochar

Vice President, Product Development, IBM

The software delivery industry is facing an overwhelming demand for the fast delivery of secure and reliable software, but the IBM Software Delivery Survey 2023 shows, that for many teams, software delivery is inefficient.

With organizations now facing difficult economic headwinds, a focus on software delivery efficiency is needed now more than ever. Improving software delivery efficiency is not about responding to market DevOps hype, it is about improving the developers' experiences and productivity based on your DevOps data. The increasing number of tools in the delivery toolchain, gaps in collaboration between stakeholders, frustration on the state of quality, and constant digital disruption are driving teams to look for more automation to become more efficient.

The IBM Software Delivery Survey 2023 reveals that there are two prevailing gaps in a software delivery team's ability to become efficient:

The software delivery experience gap

Teams are frustrated by the slow speed of adoption of DevOps and Agile principles. Nearly all developers say manual processes are expensive and unproductive whereas intelligent automation can reduce cost and lower risk. But less than one-tenth feel they have enough insight to make changes happen to their own software delivery process.

The software delivery automation gap

Developers say they could deliver better software faster if they could automate more of the non-creative parts of the software delivery process. Over two-thirds of developers say their software delivery is inefficient. And more than half of developers say the quality of their software is impacted by parts of the software delivery lifecycle outside their control.

With these gaps in mind, one thing is abundantly clear: to drive sustainable business success, organizations need to give their software delivery teams the automation tools they need. Hence the theme of this year's report, **"If software matters, then the efficiency of software delivery matters."**



James Hunter

Program Director, DevOps Automation, IBM

Key Recommendations

Help developers become more efficient by utilising more automation

- ▶ Make software efficiency an integral part of the business strategy.
- ▶ Remove manual steps from code integration and software deployment.
- ▶ Initiate test automation whenever possible.
- ▶ Assess the delivery experience of the whole delivery team.

Ensure DevOps automation adds value for both business and technical stakeholders

- ▶ Start measuring the efficiency of your DevOps lifecycle.
- ▶ Treat test automation and service virtualization as essential practices.
- ▶ Add intelligent gates to the delivery process.
- ▶ Work with process SMEs and a platform engineering team.

Measure the flow of value through your DevOps lifecycle

- ▶ Introduce automation across your hybrid DevOps toolchains and track value.
- ▶ Take a data inventory and tag the data sources that will be relevant.
- ▶ Build on what tools you have then optimize over time.
- ▶ Use a DevOps data lake to get insight from across all your tools.



Reduce the carbon footprint of software delivery

- ▶ Increase awareness and usage visibility.
- ▶ Avoid 'energy drift' and minimize the energy costs of software delivery.
- ▶ Cut carbon emissions by giving software delivery teams access to efficient environments.
- ▶ Optimize high impact activities, such as performance testing.

Test efficiently

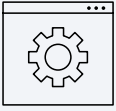
- ▶ A product's quality assurance metrics can be an indicator of the efficiency of the software delivery process.
- ▶ Combine different types of tests to assess end to end technology and business scenarios.
- ▶ Use leading indicators, such as test diversity to predict problems, support tools, and process changes to test more efficiently.
- ▶ Use lagging indicators such as DORA metrics to review the efficiency of testing activities and fixes.

Move to Data Driven DevOps

- ▶ Analyze your data from across the entire DevOps pipeline.
- ▶ Make changes to reduce bottlenecks quickly.
- ▶ Automate the production of reports and audit requirements.
- ▶ Add intelligent gates to improve efficiency and manage risk.



Current trends in software delivery efficiency



Developers want more automation



Quality is an essential part of the software delivery experience



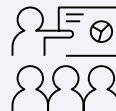
Most DevOps toolchains are perceived as being inefficient



Organizations are ready to minimize the environmental impact of software delivery



Teams want help identifying efficiency blockers



Maturing software delivery is a journey that requires DevOps



Developers want more automation

Businesses want enhanced customer experiences and competitive advantages delivered as fast as possible.

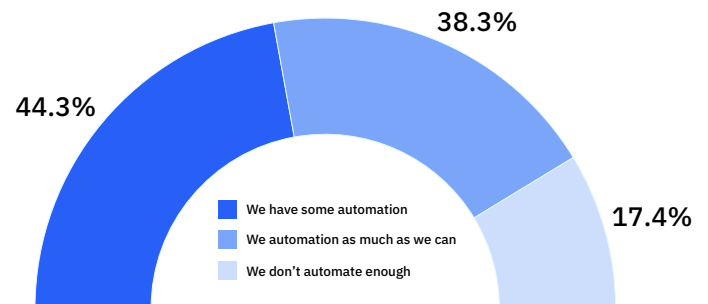
Young digital native companies want to scale their software delivery as fast as possible so they can compete with traditional enterprises. To compete with the digital native companies, traditional enterprises want digital transformation as fast as possible.

Achieving both business goals depend on the company's speed of delivering reliable and secure software. To increase their speed, software delivery teams must maximize automation and utilize emerging technologies such as artificial intelligence (AI) and AI-powered automation to deliver improved efficiency in their software delivery.

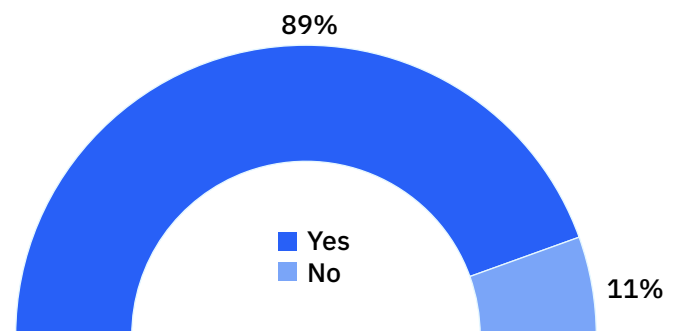
In this first section, we look at straightforward questions about automation's value and the team's level of automation.

The vast majority of those surveyed see the value in automation. With 38% of the responders indicating that they maximize that advantage. This indicates that there is an apparent gap between the perceived value of automation and the extent to which it is applied. Developers are clearly saying that they need to be able to adopt more automation to deliver better software faster.

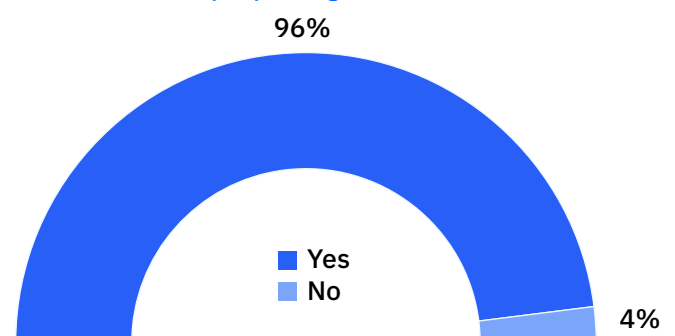
How would you rate the level of automation in your software delivery process?



Do you think your team could deliver better software faster with more automation of the non-creative tasks in your software delivery process?



Do you agree that automation can reduce cost and lower risk of software delivery in your organization?



Best practice advice

To deliver better software faster, software delivery needs to become leaner and more automated. This means we must make software delivery integral to the business roadmap. Digital-first organizations approach adopting software automation as a central part of the business strategy itself, not as an enabler of a strategy.

Remove manual steps from code integration and software deployment .

- ▶ While Continuous Integration and Continuous Deployment (CI/CD) are table stakes for most development teams, our experience is that most CI/CD involves scripting and manual work, especially for complex applications or applications that run on multiple platforms. An enterprise level automation tool for deployments is a foundation for efficiency in nearly all organizations.

Initiate test automation whenever possible.

- ▶ Testing should happen automatically whenever software is moved through the delivery lifecycle. Scan every code commit for security vulnerabilities and the end-to-end quality with functional, integration, and performance test automation. This is often the most complex set of automations to implement but has the critical benefit of lowering the risk of poor performance and quality in production, which have become intolerable to most end users.

Fix problems early.

- ▶ Start testing as early as possible so fixes happen early when they are less expensive and less disruptive to the delivery team. Fixing problems early also helps keep a release on schedule. Service virtualization can help with early testing (shifting left) as it emulates services that may not yet be fully developed and/or available to test.

Assess the delivery experience of the whole team.

- ▶ Use Value Stream Management to locate further inefficiencies and bottlenecks. Assess the developer and delivery team experience then use your baseline data to increase the level of automation to make delivery more efficient and create a better delivery experience for the whole team.

Quality is an essential part of the software delivery experience

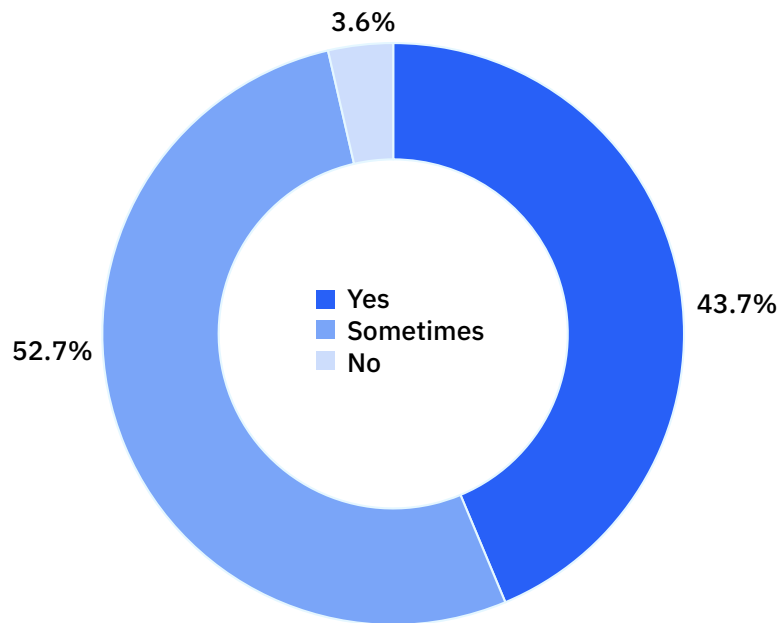
Developer job satisfaction is an important consideration for many executives as organizations compete for expert programmers.

Software delivery teams want to be acknowledged for their innovations and the quality of their deliverables. They want end users to be happy and they hope that their software will be used and will make a difference. Talented developers are frustrated by poor quality products. Not only is it a matter of pride, but it's also a practical problem as customer support case escalations will come to them.

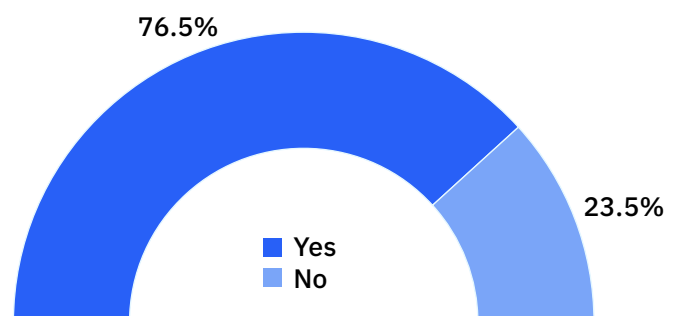
Less than half of those surveyed felt the quality of their work is represented by the quality of the overall solution that is finally delivered. Almost 8 in 10 told us that quality was affected by software delivery activities outside of their control.

There are several factors that can impact a team's perception of quality including the number of defects and issues raised by end users, the time it takes to release software after it has been developed, or a lack of visibility into how the whole delivery process fits together to produce good software.

Do you think the quality of the work you do within the delivery process is represented by the quality of the overall solution that is finally delivered?



Do you think the quality of the overall solution is marginalized by software delivery activities that are outside your control?



Best practice advice

The level of product quality can be representative of the efficiency of your software delivery.

Identify leading indicators to predict areas for improvement.

- ▶ Test diversity, the ability to reuse assets and testing early are good leading indicators of the efficiency of your testing.
- ▶ Ensure your test coverage includes unit, functional, performance, integration, and security testing. Then combine different types of tests to assess the end-to-end technology. Test data should be reused whenever possible, and automation can be used to convert one type of test to another. This level of reuse is often overlooked but is an essential part of assessing quality.
- ▶ Test as early as possible in the DevOps lifecycle by shifting testing left. For instance, waiting to perform security and performance tests until just before production is a common issue. Vulnerabilities detected late in the lifecycle disrupt the flow to production and require extensive retesting. Ideally these tests should be conducted as part of development tests to ensure changes have only positive impacts on the application.

Use lagging indicators to review the efficiency of testing activities and fixes.

- ▶ Quality as perceived by end users is a lagging indicator of your software delivery efficiency. Defects must be found before you can calculate the change failure rate or the mean time to detect and resolve issues. How your team performs against two key DORA metrics will show how efficiently problems are being fixed. Deployment frequency shows how often an organization deploys code to production or releases to end-users. Mean Lead Time for Changes measures how long it takes to push out code changes. These key metrics should be visualized in the value flow to highlight both quality and speed so efficiency improvements can be made to minimize the impact of testing activities and quality problems on the overall software release.

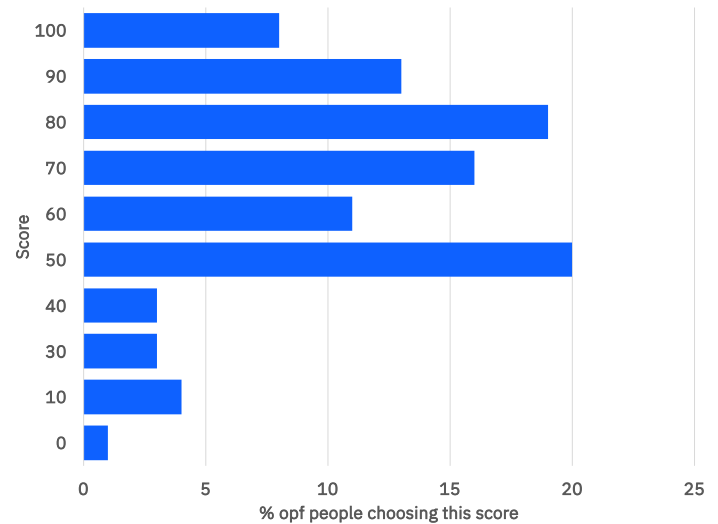
Most DevOps toolchains are perceived as being inefficient

Developers are creative people who want to deliver great software that gets used and makes a difference. It should be no surprise that they demand faster change to the adoption of automation so they can deliver software more frequently.

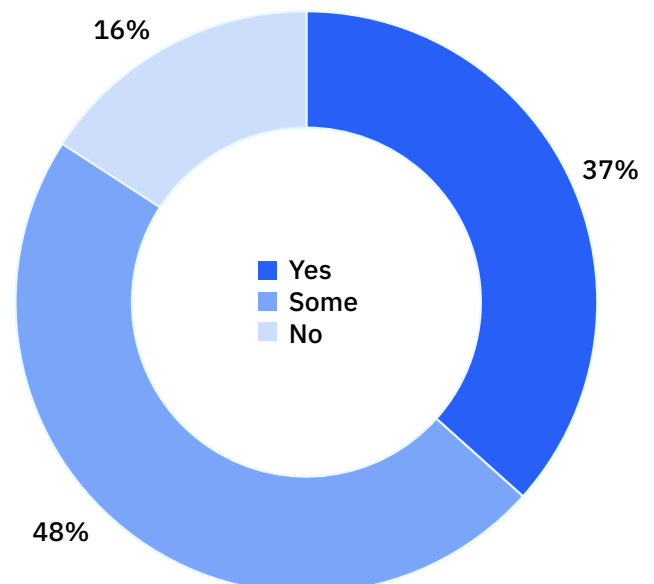
While the overall adoption of Agile and DevOps principles across our industry is encouraging, developers still think their organizations can do better. While 4 in every 10 developers are happy with their organization's rate of automation tool adoption to support DevOps and Agile principles, more than 90% of respondents think their software delivery process could add more value to the ultimate end-user or customer. Perception is often based on an individual's visibility into the state of the delivery process and the outcome of the deliverable.

Organizations that model the flow of value through their software deliver process can provide greater visibility to the state of their efficiency. For example, simply showing cycle times and visualising the flow of work from idea to business value can provide meaningful insight. DevOps should add value to all stakeholders, and value can flow through all aspects of the business, not just software development. This is the visibility that both the software delivery team and the business stakeholders need to reduce frustration and build confidence of the value being delivered by all stakeholders.

What percentage of your team's software delivery process do you feel adds actual value to the ultimate end-user or customer?



Do you think there is any frustration in your organization about the extent and rate of adopting tool automation to support DevOps and Agile principles?



Best practice advice

Eliminating inefficiencies starts with identifying the constraints and ends with remediating or eliminating those constraints.

Introduce automation across your hybrid DevOps toolchains and track value.

- ▶ Most successful DevOps toolchains are hybrid, not singular. Instead of trying to force teams to use a single DevOps tool or platform, teams can improve efficiency by introducing automation across their hybrid DevOps toolchains and tracking the flow of value through the lifecycle.

Build with the tools you have and optimize over time.

- ▶ Many organizations struggle with the cost-benefit of purchasing and implementing new software applications with speed or improving the quality of software delivery. Data gathered from the tools involved in the actual software delivery process help provide visibility into where the delays reside or where testing or quality is short-changed. Visualization of these problems in software delivery will help quantify and justify investment in missing automation.

Use a DevOps data lake to get insight from all your tools.

- ▶ Most DevOps pipelines involve planning, development, deployment, testing, release orchestration, and management. Each stage is supported by one or more tools to help with executing that task and the data for that activity or task is usually stored in that tool's repository. Identifying opportunities to become more efficient requires real time value stream data from across your software delivery lifecycle. Whenever possible, include business stakeholder tools as well as delivery tools to derive insights from across the business. Measure the flow of value then analyse trends over time as your bottlenecks improve so you can continually assess and prioritize the next focus areas to further improve your software delivery efficiency.

Organizations are ready to minimize their environmental impact

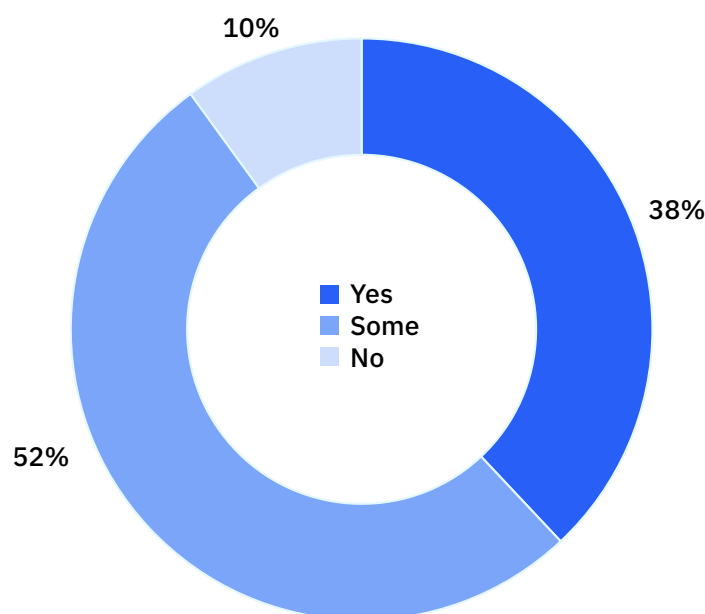
Modern businesses are trying to balance what they can do to reduce their impact on the planet with their need to continuously develop, deliver and run software solutions.

While most organizations have set a net-zero carbon emissions goal, only 38% of those surveyed said it is common practice in their organization to assess and optimize the environmental and sustainability impact of their software delivery process and/or software releases.

Recent Institute for Business Value consumer research revealed that 53% of surveyed organizations view environmental sustainability as one of their top priorities within the next three years, which is an increase from the 39% that consider it a top priority today. However, while 86% of those companies have a sustainability strategy, only 35% had acted on that strategy.

Cloud infrastructures are becoming more efficient so carefully identifying where to run high intensity activities like test execution and performance testing are leading activities that can make a difference. Truly sustainable software development and delivery is difficult to achieve. While it is positive that 52% of those surveyed agreed that their organization did not assess or optimize the environmental and sustainability impact of their software delivery process and/or software releases, the software delivery team think they should.

Is it common practice in your organization to assess and optimize the environmental and sustainability impact of your software delivery process and/or software releases?



Best practice advice

Improving the efficiency of software delivery contributes to achieving net-zero carbon emission objective and developers play an important role.

Minimize energy costs.

- ▶ Reduce energy costs and avoid energy ‘drift’ that often goes unnoticed, such as using ephemeral test environments that are decommissioned when not in use.

Cut carbon emissions.

- ▶ Give software delivery teams access to hybrid cloud and multi cloud environments. Then allow teams to assess and consider where to run workloads based on optimization of performance, cost and Greenhouse Gas emissions of the required infrastructure or service.

Optimize high impact activities.

- ▶ Maximum test coverage with the minimum number of tests. Then run tests at times and locations that generate less carbon.

Increase awareness and usage visibility.

- ▶ Combine energy usage data from the portfolio level with delivery activity reports.

Teams want help identifying efficiency blockers

Most teams are being asked to turn ideas into business value as fast as possible.

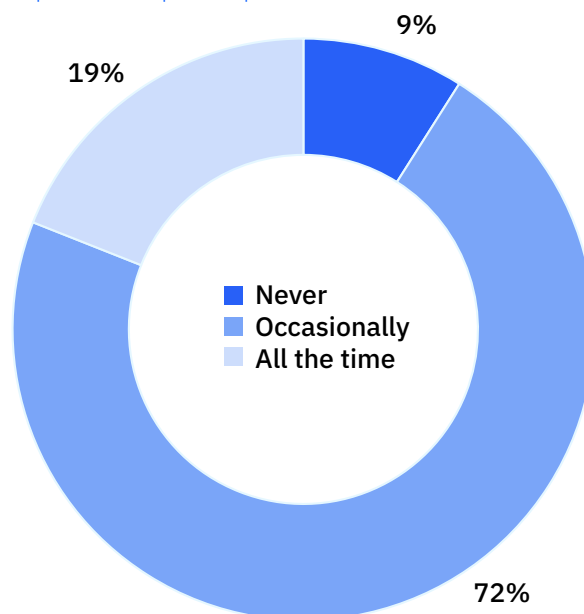
As one might expect, how to achieve this viewpoint most efficiently greatly varies. A person's role in software delivery will often dictate their view about the causes of inefficiency. In most organization's roles still tend to be siloed, even on a highly collaborative software delivery team. Work is complex and requires deep focus on the tasks at hand. However, some trends do emerge. The people who responded to this year's survey say manual steps impact more than any other detractor on efficiency, followed by the decision-making process within their organization. Both require automation and insights that, as we have seen in other areas of this survey, are lacking in most teams.

Teams that have modernized their software delivery automation are more likely to release products more frequently and have fewer delays in their product releases due to a more efficient delivery. DevOps teams that are more agile and using automation tools, are also more likely to identify further opportunities for improving efficiency and speed of their delivery of reliable software. However, tool experts were identified as the role most often missing in software delivery teams.

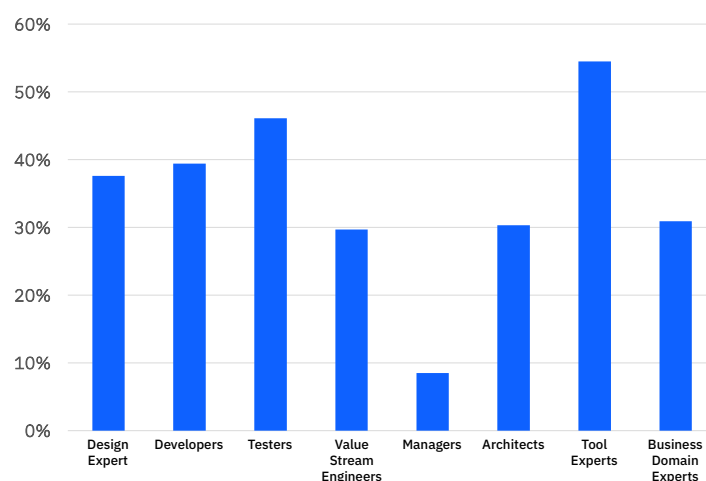
Tool experts are needed to keep the software delivery tools running efficiently, especially when you consider that the majority of teams occasionally or frequently add customized tools to their toolchain. The onboarding of new tools and low investment into tool experts can significantly slow down the software delivery process.

Efficiency blockers in DevOps are impacted by processes, skills, manual intervention, change requests, environment setup, tool usage and decisions. Efficiency is about closing the loop as the application progresses through each phase of the lifecycle. To close the loop, visibility, collaboration, and observability of the flow of assets is essential and the reason why there is a significant demand from teams for more automation.

How often do you fill gaps in your software delivery toolchain with point or bespoke capabilities?



Do you think there are any skills missing in your team(s) that could help improve the efficiency of software delivery in your organization?



What do you think are the biggest causes of inefficiency on software delivery in your organization ?

Item	Overall Rank	Rank Distribution	Score	No. of Rankings
Manual steps in the delivery process	1		601	127
Decision making	2		567	136
Lack of skills in the group	3		514	129
Managing tool and technology updates	4		503	116
Changing requirements	5		494	127
Lack of testing environment or data	6		491	142
The number of tools and integrations in the delivery process	7		491	123

Best practice advice

Teams need help in identifying where to start on their journey in improving their software delivery efficiency. DevOps Automation should not be seen as an inconvenience. It should complement culture change and be part of the overall strategy to deliver products to the market.

- ▶ **Before adding new tools, measure the efficiency of your DevOps lifecycle by understanding each phase and how it impacts the downstream workflow.**

Optimizing efficiency requires data to accurately prioritize investment. Start by creating a DevOps data lake and integrating existing tools. Then trace the flow of value through the software delivery lifecycle to derive actionable insights from the collated data to help identify where to prioritize. When possible, deploy new tools on new, smaller projects to learn about them and to cultivate experts before introducing new tools into existing pipelines. Use data to determine if the new tool will really add value.

- ▶ **Speed up the onboarding process for new teams by considering automation of the toolchain and a self-service portal.**

Approach platform support as you would a development project.
Leverage well-established CI/CD principles to automate your toolchain and maximize reuse.
Integrate tool data into your DevOps data lake for future analysis.

- ▶ **Track work items automatically in swim lanes.**

Highlight who needs help, identify which items are blocked and automate the production of reports and audit reports.

- ▶ **Add intelligent gates to the delivery process to improve efficiency and manage risk.**

Ensure there's representation from each phase of the DevOps Lifecycle including tool experts or a platform engineering team, then improve collaboration. Train teams on tools that eliminate additional manual process, then use observability tools that will provide feedback in each DevOps phase.

Maturing software delivery is a journey that requires data

Most enterprises are on a journey to transform their software delivery. They want to speed up their digital transformations and improve the return on investment (ROI) for their spend on DevOps automation tools. The challenge for any organization on this journey is to understand their current level of DevOps automation maturity and their potential. Only then can teams plan their journey to high performing continuous delivery of reliable and secure software.

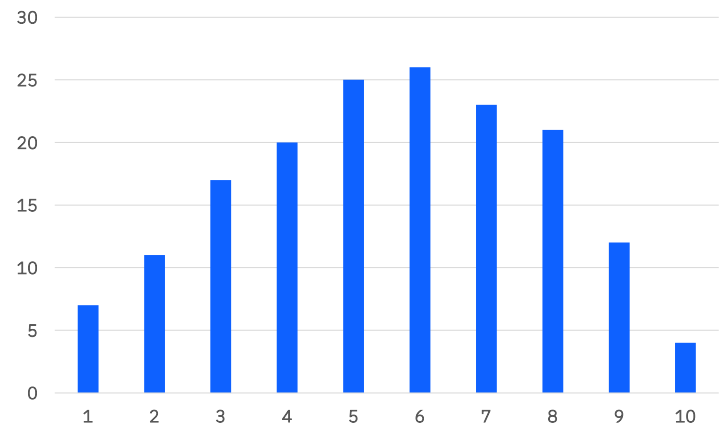
Initially, the team's perception of their maturity of automation is important simply because it gives an indication of how costly it is to be an organization that develops and delivers software. Most organizations want to cut costs and increase the speed of software delivery. The focus then must be on improving efficiency.

The people involved in delivering software told us how mature they feel their use of automation is today by rating it from 10 (indicating it is very automated and efficient) to 0 (indicating a significant lack of automation and very inefficient delivery). Perceptions of current maturity resulted in a classic bell curve and more teams are still in the early stage of their automation adoption journey.

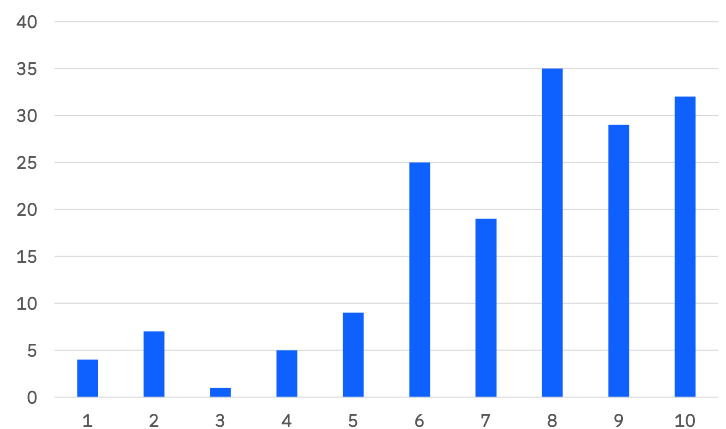
Although most teams aspire to improve, less than 20% felt their organization is likely to achieve the highest levels of DevOps Automation. Prioritizing what to automate requires collating data from across the software delivery toolchain, but only 10% of developers felt their organization had the visibility into their toolchain to identify and influence efficiency improvements.

Most organizations want to mature their use of automation but struggle to know how to maximize potential because there is not enough insight into where to optimize.

How would you rate the current maturity of DevOps automation in your group?



How would you rate the potential of DevOps automation in your group?



Best practice advice

Maturing software delivery efficiency requires data and intelligent automation to identify and address blockers but efficiency measurements mean nothing unless you act.

DevOps automation should add value to all stakeholders. This means a change to become more efficient depends as much on automation technology as it does on the culture within the group and the level of executive support to invest in improving efficiency. All relevant stakeholders should have access to the software delivery value flow data.

Analyze the data from across the entire DevOps pipeline and make informed decisions to boost your software delivery efficiency.

- ▶ Where teams have made large investments over time into disparate tools with separate datasets, teams should look for automation to collate data from disparate sources into a single, comprehensive view to identify bottlenecks, measure efficiency, and prioritize changes for maximum impact.

Make changes quickly to reduce bottlenecks and improve your software delivery.

- ▶ Use a single source of DevOps data to identify bottlenecks and derive insights on where to prioritize change to improve the efficiency of your software delivery quickly.
- ▶ Track work items automatically in swimlanes to highlight who needs help or which items are stuck.

Maximize the return on your existing investment into DevOps toolchains and data sources.

- ▶ Enhance efficiency and manage risk by incorporating smart gates into the delivery process.
- ▶ Streamline the generation of reports and meet audit requirements through automation.

About this study

To understand how software delivery professionals perceive efficiency, we surveyed members of the IBM Community who are from a variety of industries and geographic locations.

Region

Americas	36.20%
Asia	15.90%
Europe	39.10%
Middle East	4.30%
Africa	2.90%
Oceania	1.40%

Role description

I lead or manage software delivery teams or activities	30.40%
I am a designer or software architect	26.10%
I am a software engineer/developer	49.30%
I mostly focus on assessing quality and security	10.10%
I manage releases	13.00%
I am responsible for processes and fixing problems	29.00%
Other (please state).....	13.00%

Time in software delivery

Under 2 years	14.70%
3-5 years	10.30%
6-10 years	11.80%
More than 10 years	63.20%



How to measure and improve efficiency

Automation has been shown to have positive effects on the efficiency of software delivery for many organizations, and the IBM Software Delivery Survey 2023 showed that more than two-thirds of developers see opportunities to improve the efficiency of software delivery in their organization.

Many organizations understand the need to deliver software efficiently yet struggle to measure and realize the return on their investments into disparate automation tools and processes. It can take many different complex processes to develop and deliver software — from idea capture and design, to development, testing and controlled release to production. These processes are often hindered by bottlenecks or rife with inefficiencies that can slow down delivery, increase the risk of poor-quality software and ultimately jeopardize customer satisfaction and business objectives.

Intelligent automation, underpinned by a DevOps data lake, value stream management and AI, provides decision-makers with visibility into the software delivery process, and guidance to reduce efficiency blockers.

Visibility into the software delivery process afforded by intelligent automation can have many beneficial effects: it simplifies manual and automated processes, frees up resources, improves operational efficiencies, and accelerates the returns on investment. For organizations looking to implement intelligent automation, Value Stream Management (VSM) is an ideal place to start. VSM data aggregated from the existing software delivery toolchain provides insights for improving the efficiency of software delivery. With VSM, the business benefits from a detailed view into how well an application is progressing through the software delivery pipeline, detects where there are inefficiencies and where optimization and automation can have the greatest impact. That's why understanding your value stream is an ideal starting point.

At IBM, we've seen the benefits of improving the efficiency of software delivery with intelligent automation and how it can quickly make a positive impact for our business.

97%
reduction in the cost
of releases



75%
faster deployment
times



35%
increase in test
coverage



IBM DevOps Automation makes it easy to get a view of software delivery bottlenecks and the flow of value so teams can identify the key areas to prioritize for improved efficiency. The end goal is for businesses to realize value in the form of reliable and secure software developed, delivered, and released as fast as possible.

IBM DevOps Automation has individual capabilities that add automation to an existing software delivery toolchain or can be used as a complete delivery platform for any software delivery project. And it is unique in the market as the only DevOps Automation solution by IBM that supports all IBM strategic technologies including Instana, Turbonomic, API Connect, Liberty, Robotic Process Automation, Ansible, Watson, Engineering Lifecycle Management, IBM CloudPaks and more.

To learn more about IBM DevOps Automation and how it can help your organization to improve software delivery efficiency, we invite you to join the IBM DevOps Automation community.

<https://ibm.biz/DevOpsAutomation>



Thank You

IBM would like to thank all those who took part in this Software Delivery survey. This survey was conducted following professional ethics and practices and collected opinions from the people who are hands on in the process of software delivery.

The results provide a unique look into the real state of software delivery from a ground up perspective and highlights the practical problems and concerns that need to be addressed if businesses want to develop and deliver software efficiently.

The survey was voluntary, and all responses are completely confidential. The results have only been made available in aggregate and not shared with any reference to any individual or organization who contributed. Being trusted to manage the personal data of the people who took part in the survey, means we get a true and honest representation of the real-world challenges. This insight from these collated results together with the analysis and advice by DevOps subject matter experts will ultimately help all software development and delivery teams to become more efficient.

Report Author

James Hunter

Marketing & Comms

Cassidy Rimer

Neil Patterson

Steve Fitchett

Creative Lead

Connor Allen

IBM DevOps Automation

Madhu Kochar

Vice President,
Product Development
kochar@us.ibm.com

James Hunter

Program Director
DevOps Automation
james.hunter@uk.ibm.com

Osman Burucu

Capability Lead,
Value Stream Management
osman.Burucu@at.ibm.com

Lettie Ar-Rahmaan

Capability Lead,
Test Automation
willetta@us.ibm.com

Cassidy Rimer

DevOps Automation
Product Marketing Lead
crheath@us.ibm.com

To learn more about IBM DevOps Automation and how it can help your organization to improve software delivery efficiency, we invite you to join the IBM DevOps Automation community.

<https://ibm.biz/DevOpsAutomation>



<https://ibm.biz/GetInTouch>

© Copyright IBM Corporation 2023

IBM Corporation
New Orchard Road
Armonk, NY 10504

Produced in the United Kingdom | January 2023

IBM, the IBM logo and [ibm.com](https://www.ibm.com) are trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of IBM trademarks is available on the web at “Copyright and trademark information” at: [ibm.com/legal/copytrade.shtml](https://www.ibm.com/legal/copytrade.shtml).

This document is current as of the initial date of publication and may be changed by IBM at any time. Not all offerings are available in every country in which IBM operates.

THE INFORMATION IN THIS DOCUMENT IS PROVIDED “AS IS” WITHOUT ANY WARRANTY, EXPRESS OR IMPLIED, INCLUDING WITHOUT ANY WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND ANY WARRANTY OR CONDITION OF NON-INFRINGEMENT. IBM products are warranted according to the terms and conditions of the agreements under which they are provided.

This report is intended for general guidance only. It is not intended to be a substitute for detailed research or the exercise of professional judgment. IBM shall not be responsible for any loss whatsoever sustained by any organization or person who relies on this publication.

The data used in this report may be derived from third-party sources and IBM does not independently verify, validate, or audit such data. The results from the use of such data are provided on an “as is” basis and IBM makes no representations or warranties, express or implied.

