The State of Continuous Delivery

Continuous delivery is all about optimizing the software delivery loop via a set of well-defined best practices that enable DevOps teams to make as many iterations as possible to a feature or service. Continuous integration, test automation, continuous deployments and automated security all need to be integrated within those processes.

Deployment Frequency: How frequently team successfully releases into production.

Lead Time for Changes: The median amount of time for a commit to be deployed into production.

Change Fail Rate: The number of failures per the number of deployments.

Time to Restore Services: The median amount of time between the deployment which caused the failure and restoration.

PLAY VIDEO

Continuous Delivery: A Closer Look

Despite widespread adoption, continuous delivery can be difficult to master, incorporating multiple elements that are anything but one-size-fits-all.

The Continuous Delivery Challenge

The global continuous delivery market size is expected to reach **$6.38 billion by 2028**, representing a **17.7% compound annual growth rate.**

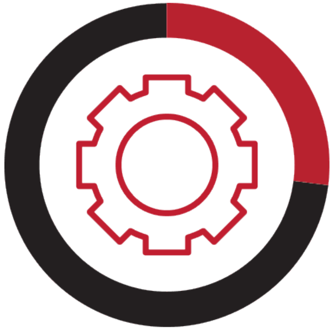
Only **1 in 10 developers are elite performers** in terms of deployment frequency, i.e. they release multiple deploys per day.

Nearly **two-thirds of developers take at least one week** to go from code committed to code successfully running in production.

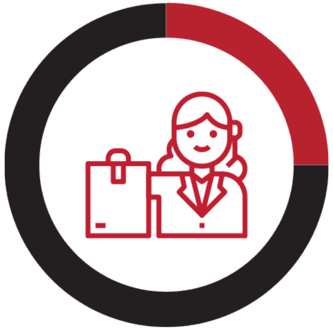
Only **half of developers** report that they restore service from an unplanned outage in less than a day.

Source

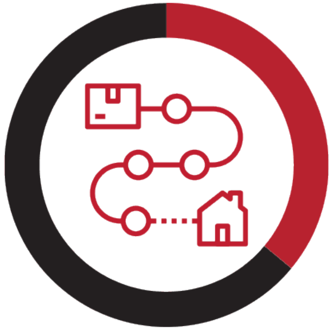
The Continuous Delivery Reality



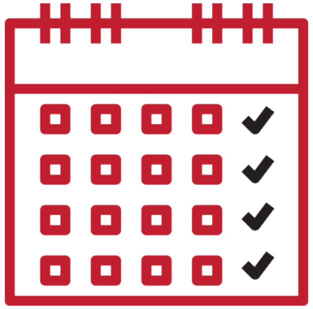
Only **27% of respondents** say they practice continuous deployment with automation, from commit through to testing, build / integration, package and final release.



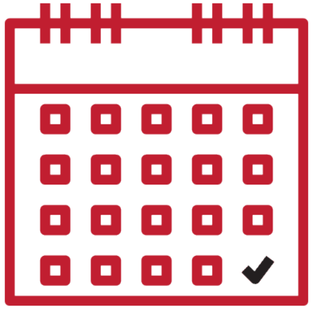
Just **25% of companies** rate their governance, risk and compliance capability for software delivery as being very mature.



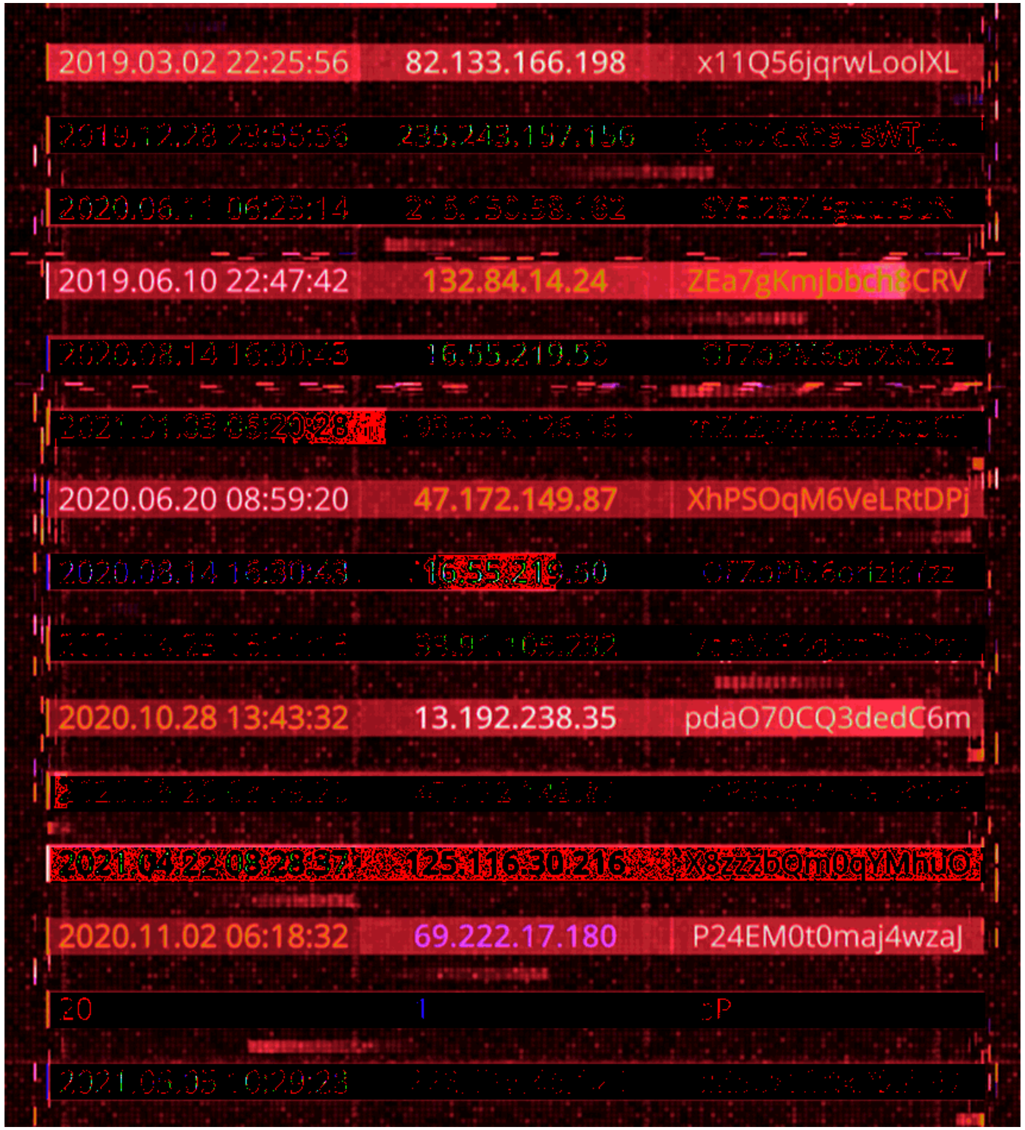
More than **a third of organizations (36%)** have been able to achieve continuous delivery, but only **just over a quarter (27%)** said they have also achieved continuous deployment.



Only **a third of organizations** deploy new code at least once per week.



Almost half still deploy less than once per month.


2019.03.02 22:25:56 82.133.166.198 x11Q56jqrwLoolXL
2019.12.28 22:55:56 235.243.157.156
2019.06.10 22:47:42 132.84.14.24 ZEa7gKmjbbch8CRV
2020.08.14 16:30:43 16.55.219.50
2020.06.20 08:59:20 47.172.149.87 XhPSOqM6VeLRtDPj
2020.08.14 16:30:43 16.55.219.50
2020.10.28 13:43:32 13.192.238.35 pdaO70CQ3dedC6m
2021.04.22 08:28:37 125.116.30.216 X8zzzbQm0qYMnuO
2020.11.02 06:18:32 69.222.17.180 P24EM0t0maj4wzaJ

The Promise of GitOps and Kubernetes

GitOps uses a Git repository as the single source of truth for declarative infrastructure and applications. Software agents identify any divergence between Git with code running on a Kubernetes cluster. If there's a difference, Kubernetes automatically updates or rolls back the cluster. That capability makes it easier for organizations to standardize continuous delivery across a common set of APIs. At the moment, however, **only 16% are running Kubernetes exclusively**, so challenges remain.

VIEW WEBINAR

Summary

Continuous delivery represents the ultimate in DevOps best practices. It is becoming more attainable for organizations that today are more dependent than ever on software. In fact, continuous delivery means the difference between simply deploying code and running an organization like a true software company.

Sponsored by:

PLUTORA

STACKHAWK

LEVELOPS

KEYSIGHT
TECHNOLOGIES

Thank you for reading

The State of Continuous Delivery