

HOW TO MEASURE THE VALUE OF YOUR DEVOPS ORGANIZATION



many ways, moving from development to production is like driving along a road trying to reach your destination as efficiently and smoothly as possible. Taking a road with bumps, blind spots, and heavy traffic can make it difficult to get there faster and safer - not to mention the aggravation that occurs when you're backed up in traffic and delayed.

When it comes to [DevOps](#), the road to application delivery can also be bumpy and challenging to navigate. There's a lot of pressure to deliver apps rapidly to get your services to market and keep up with your customers' changing needs. Fortunately, you can accelerate application delivery — and do this easier, more efficiently, cost-effectively, and accurately — with best practice processes and technology.

DORA findings describe how to measure the ROI of DevOps

Begin by analyzing where your organization stands in terms of application delivery and what you can do to improve it. The DORA Report, [Forecasting the Value of DevOps Transformations – Measuring the ROI of DevOps](#), provides a comprehensive overview of how DevOps teams can measure and enhance productivity. It explains how to estimate the potential ROI associated with delivering more capabilities and improving application performance.

Here are some key takeaways from the DORA Report¹ to help determine how well your organization is performing. The report describes how developers can measure agility and how IT operations can measure reliability. The list below includes some capabilities to consider:

1. **Deployment frequency** — Did you know that the highest-performing IT organizations deploy more applications more frequently, such as **multiple deployments per day** versus between once a month and once every six months for lower performing IT organizations? More frequent deployments reduce the chance of disruption caused by more complex, less frequent deployments.
2. **Mean-time-to-restore (MTTR)** — Higher performing organizations restore service in **less than one hour**, while the lower-performing ones report restoring service in one day. Since the average cost of critical application failure is between \$500K to \$1M per hour, these disruptions can be extremely expensive.
3. **Lead time for changes** — While it takes high performing organizations **less than one hour** to go from commit to code that's successfully running in production, it can take between **one and six months** for low IT performing organizations.
4. **Value and percentage gained from avoiding unnecessary rework** — High IT performers reduced rework and dramatically increased revenue by using that extra time to develop more value-added services. In fact, high performers spend 50% less time on unnecessary rework than low and medium performers.
5. **Potential value from reinvesting in new features** — When your DevOps teams are more efficient, they can free up time to create new features customers want and drive more revenue. High-performing large organizations, which deploy significantly more apps than medium or low-performing organizations, have the potential to add about **50x more revenue** than the low-performing large organizations and **about 5x more revenue** than medium-performing large organizations.

Take a Jobs-as Code Approach to DevOps

Since delivering new services to market rapidly drives revenue, look at what your organization can do to improve the process of application delivery. That's where [Control-M Automation API](#) fits in. It simplifies scheduling and enables developers to create more reliable applications quickly and easily, allowing DevOps to adopt a **Jobs-as-Code** approach to minimize the pressure, avoid rework, and accelerate the speed of application delivery by 30%.

So, instead of using a cobbled together group of solutions for job scheduling and batch automation, Control-M Automation API provides a 360-degree view with operational instrumentation baked into the coding phase. This takes the pain out of defining jobs because developers can use a simple code-like notation combined with application business logic. There's no scripting required to build, test, and debug application automation. All of these capabilities lead to **more frequent deployments and shorter lead time required for changes**.

Enjoy the ride

Businesses can meet the demands of digital business when Dev and Ops aren't slowed down by process-driven speed bumps and conflicting priorities. Developers can easily move quickly to embed workflow automation while applications are being developed. Errors are reduced by testing early. This reduces rework in operations and improves application quality, so costs go down.

As more tasks are automated, and developers are no longer working on the operational "plumbing" when they're coding applications, they have more time to dedicate to developing new, innovative features that drive business value.

Learn more about how taking a Jobs-as-Code approach with Control-M lets you drive at the speed your business requires on the road from application development to service delivery. [View infographic](#).

¹ DORA Report, *Forecasting the Value of DevOps Transformations – Measuring the ROI of DevOps*, March 2017.