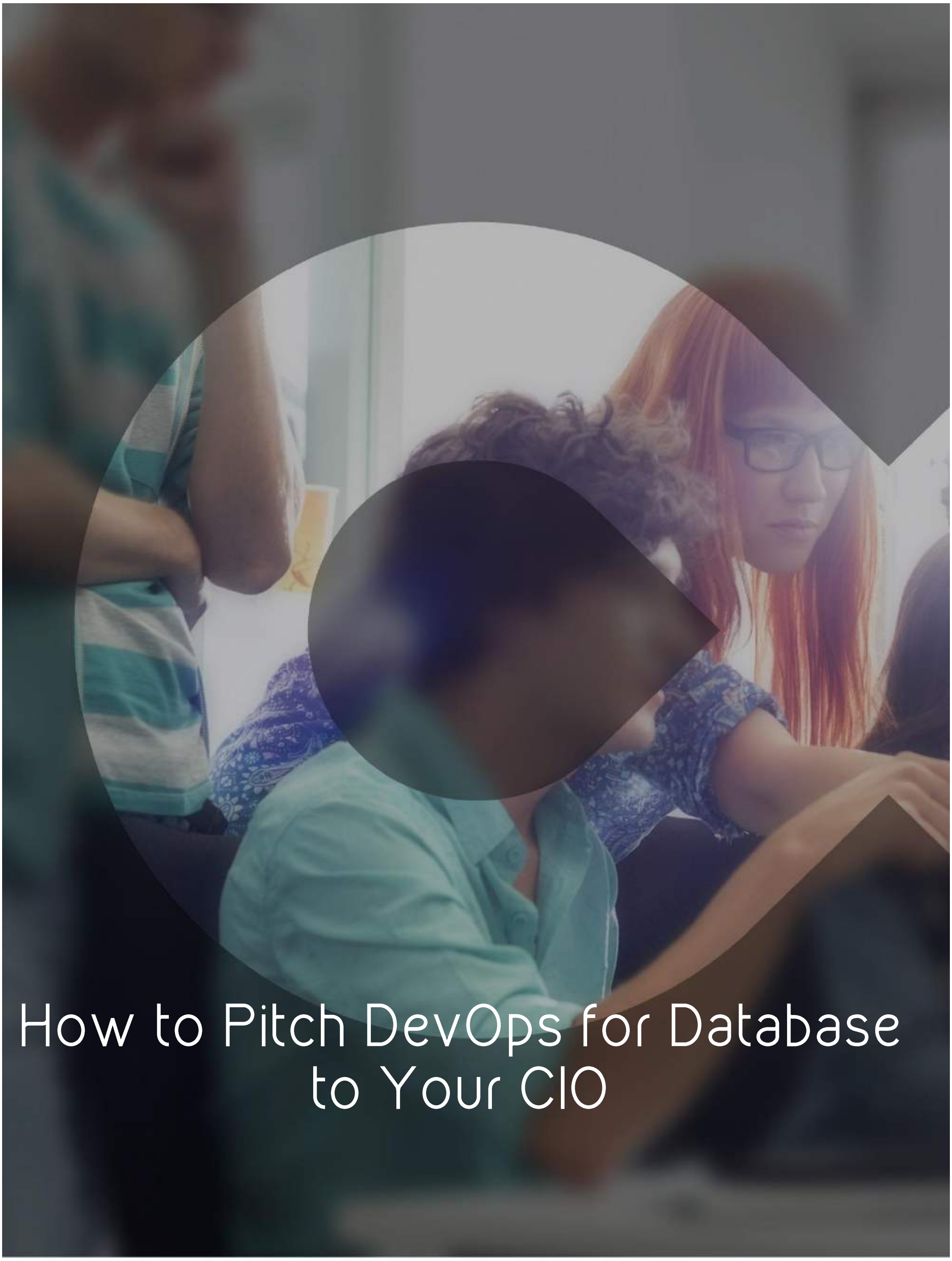


How to Pitch DevOps for Database to Your CIO

DBmaestro
DevOps for Database

Executive Summary

The modern chief information officer (CIO) plays a growing role in connection with product development. This includes all or any of technical elements of building the product, delivery, aggregation and monetization, the quality and effectiveness of which has a substantial impact on the company's financial success.

As more companies find themselves developing and maintaining applications – either proprietary applications for employee use or consumer-facing applications as a component of their larger business model – the need for greater agility to keep up with business needs and consumer demands becomes obvious. A common solution, whose use is rapidly spreading across various industries, is continuous delivery.

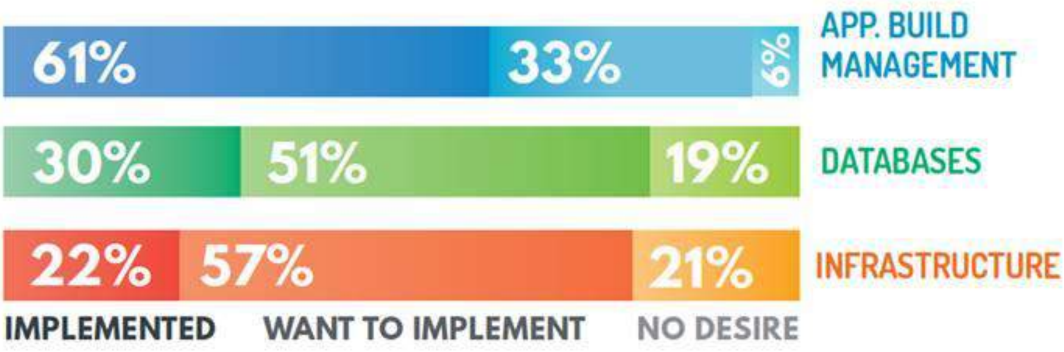
While continuous delivery practices are being used in application development, they are generally not being used for maintaining relevant databases. Instead, many resources are being devoted to inefficient, manual processes to keep the database updated; however, this risks catastrophic failure, possibly even after the "go live" stage.

A DevOps for the database solution can help organizations overcome the common challenges development teams face in keeping their databases up to speed in a continuous delivery environment. Such an improved database administration solution increases productivity, reduces errors, and generates substantial savings of both time and resources.

Those development benefits are well in line with CIO priorities, which opens the door to confidently pitching the value of DevOps for the database to your company's CIO.

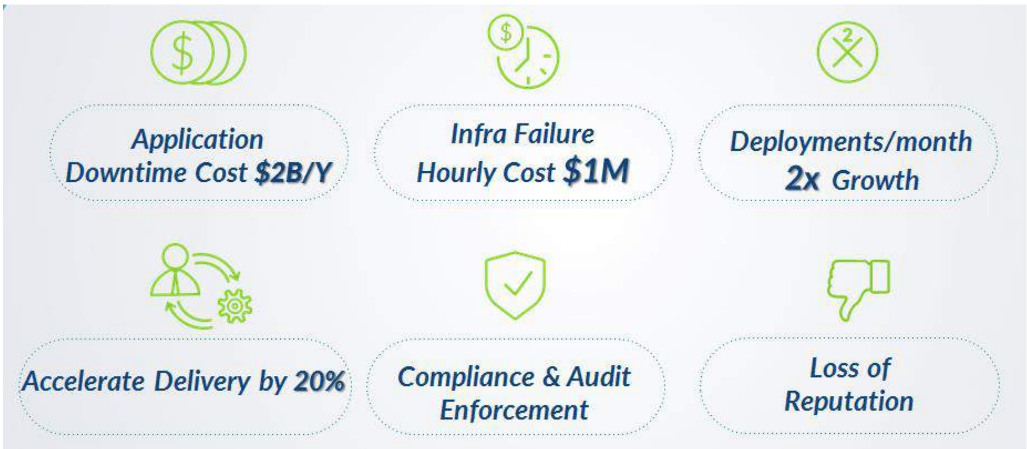
Continuous Delivery Continues to Grow

According to Dzone's 2015 Guide to Continuous Delivery, adoption of continuous delivery continues to grow for different software lifecycle environments. The most popular continuous delivery environment is application build management, with 61 percent of survey respondents reporting that they use CD for this purpose and another 33 percent stating that they want to implement it.



Dzone also found a nine percent (9%) growth in adoption of continuous delivery practices for the database. Dzone points that this growth is a substantial gain within less than one year, although the finding is not unexpected given the growing adoption of continuous delivery for database lifecycle management. Still, just 30 percent of survey respondents report that they have implemented continuous delivery for the database, with more than half (51%) stating that they'd like to implement CD for the database.

According to IDC's DevOps and the Cost of Downtime: Fortune 1000 Best Practice Metrics Quantified report from December 2014, the average number of deployments per month is expected to double in the next two years. In that same time period, DevOps is expected to adopt security, compliance, and audit responsibilities, with DevOps-led practices expected to accelerate the delivery of capabilities to the consumer by 15 to 20 percent.


Application Downtime Cost $2B/Y
Infra Failure Hourly Cost $1M
Deployments/month 2x Growth
Accelerate Delivery by 20%
Compliance & Audit Enforcement
Loss of Reputation

With the demands on development growing rapidly and showing no signs of slowing, there's an ever-increasing demand for development to become more agile. Mistakes in the development lifecycle, of course, can be costly. IDC's report finds that the average total cost of unplanned application downtime is $1.25 to $2.5 billion – per year. An infrastructure failure can have catastrophic financial consequences at an estimated cost of $100,000 to $1 million *hourly*.

Your Database is Being Left Behind

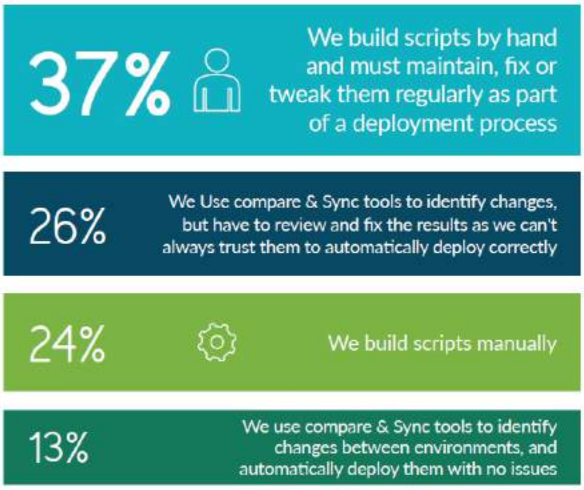
More than twice as many organizations have implemented continuous delivery practices for application development compared to those which have implemented continuous delivery practices for database development (61% vs. 30%). These findings are consistent with DBmaestro's DevOps for the Database 2015 survey of more than 450 IT professionals, which found that 66 percent of respondents were using continuous integration or build automation for their applications, but just 34 percent reporting that they use continuous delivery for the database.

That said, just 13 percent of respondents who said they use continuous delivery for the database actually have automated processes in place, which is a prerequisite for continuous delivery. The result, for both organizations that haven't yet implemented continuous delivery for the database and those attempting to implement continuous delivery for the database without automated processes, is that the database often gets left behind.

Understanding the Application-Database Development Disconnect

Why does this disconnect exist? While 81 percent of respondents in a previous survey indicated that they believed it possible to practice continuous delivery for the database, automation is still largely considered too risky for database deployment. To avoid these risks, most organizations plug the continuous delivery process with time-consuming manual steps that, while minimizing some of the perceived risks of automation, introduce the likelihood of human error.

More than one-third (37%) of organizations build scripts by hand and are thus faced with maintaining, fixing, and tweaking these scripts regularly as part of the deployment process, and more than one-fourth (26%) use standard compare-and-sync tools to identify changes; yet, even compare-and-sync tools require manual review and intervention, because there's no way to guarantee where changes originated or which version is safe for deployment. Just 13% say that they use compare-and-sync tools to identify changes between environments and are able to automatically deploy those changes without experiencing issues. Nearly one-fourth of respondents (24%) build scripts manually.



37% We build scripts by hand and must maintain, fix or tweak them regularly as part of a deployment process

26% We Use compare & Sync tools to identify changes, but have to review and fix the results as we can't always trust them to automatically deploy correctly

24% We build scripts manually

13% We use compare & Sync tools to identify changes between environments, and automatically deploy them with no issues

Consider the fact that most organizations are making frequent changes to the database (32% are making database changes daily, while 37% report making database changes weekly) and it's not difficult to realize the vast resources devoted to inefficient, manual processes that create the need for frequent re-tracing of steps and implementing quick fixes, if not complete crashes or failures. Even if you're among the lucky 13% of organizations who utilize compare-and-sync tools and automatically deploy changes without issues, odds are you won't be that lucky with every deployment.

Implementing DevOps for the Database: Why Baseline-Aware Analysis is Key

Deployment automation is generally less trusted among developers and DBAs when it comes to the database, whereas code developers are far more likely to readily embrace automation. DevOps for the database is an effective solution for overcoming the challenges of automating database deployment, yet it must be implemented correctly in order to avoid introducing additional risks.

DevOps for database solutions must utilize baseline-aware analysis in order to avoid the standard pitfalls of automation. For instance, standard compare-and-sync tools can identify changes between environments, but simply automating deployment without manually verifying where changes originated and which changes are safe to deploy could mean inadvertently overriding production hot fixes or work done in parallel by another team. If you're lucky, these errors will be identified before they lead to a catastrophic crash for your end users. Otherwise, such errors can cost hundreds of thousands of dollars to rectify.

Only DevOps for database solutions offering baseline-aware analysis can not only detect changes between two environments, but, by comparing both environments to the baseline, identify which changes should be overridden, which should be ignored, and which require manual intervention and merging. This enables true automation, while providing a safety net to prevent major errors from being released. What's more, the most robust DevOps for database solutions provide a complete audit trail, simplifying compliance requirements – an obligation that often plagues CIOs due to inconsistent documentation practices that may not accurately reflect out-of-process changes.

How CIOs Get the Best Results from DevOps for the Database

To maximize the effectiveness of any DevOps for database solution, CIOs should be advised to ensure that it includes the following:

- √ Three-way, baseline-aware analysis leveraging a single source of truth.
- √ An integrated database source control process.
- √ The ability to control out-of-process changes.
- √ A safety net for deployment automation preventing errors from being released.
- √ A complete audit trail history of all changes made to the database.

Implemented correctly, and especially when leveraging the above features, DevOps for database solutions offer a multitude of benefits, including increased productivity, error reduction, and substantial savings of both time and resources. This translates to the achievement of many CIO goals, such as shipping out rapid changes that coincide with application development, and keeping pace with user and market demands, as well as releases with far fewer risks and backouts, quicker time-to-market, and better customer service. These increases in both quality and productivity have a substantial impact on development costs and the company's bottom line, as well.

These solutions put your organization in a position to build a full delivery pipeline, allowing you to verify, build, deploy, and promote database changes just as you do with application code. By leveraging the same continuous delivery best practices that have proven successful for application code, your database is no longer left behind and can once again serve as the strongest link in your development lifecycle.

> *Focusing on changes rather than managing changes and dealing with re-work, boosted overall productivity of 250 developers. We estimate* ***we were able to do 15% more*** *with the same resources.*
>
> *We went from several fix-centric deployments a day, to one feature-centric deployment a week.*
>
> *The amount of incidents in production has declined as well.* ***We had 20% less incidents****.*
>
> *CIO @ Credit Card company*

DBmaestro TeamWork

DBmaestro is a pioneer and a leading DevOps for database solution provider. DBmaestro TeamWork enables Agile development and Continuous Integration and Delivery for the Database. TeamWork streamlines development process management and enforces change policy best practices, facilitating complete Database Lifecycle Management (DLM).

The DBmaestro TeamWork solution empowers Agile team collaboration, while fostering regulatory compliance and governance. This eliminates the challenges of automating database development, while simultaneously automating critical documentation practices and ensuring a complete audit trail history. With DBmaestro, organizations can facilitate DevOps for database by executing deployment automation, enhancing and reinforcing security, and mitigating risk.

DBmaestro's vision is to simplify and streamline deployment processes from the Agile development stage and reduce critical application downtime. Automation with fewer risks means quicker time-to-market, coupled with savings in time and resources otherwise spent back-tracking and implementing fixes. The ability to release changes rapidly to keep pace with market demands, while simultaneously making substantial improvements in both quality and productivity, translates to dramatic improvements to the company's bottom line: a top priority for any CIO.