Overcoming the developer capacity dilemma

Presented by PlanetScale

In today's digital age, data is the lifeblood of any organization.

But as the volume and complexity of data have grown, so too has the challenge of managing it effectively. Data often becomes a blocker for the business to evolve as its sheer complexity drains developer capacity. Rather than focusing on increasing deployment frequency, developers may find themselves dedicating significant time to tasks such as managing servers, scaling a caching system, ensuring database availability, and more. All of this to get the data to a state where it can be leveraged to innovate. This can make it difficult for companies to deliver new products and services to the market.

The importance of removing the database as a blocker

DevOps practices and digital transformation improve product velocity and enable organizations to better leverage their data to build products, but this can also absorb developer capacity and make it difficult for existing technical talent to focus on innovating.

Incorporating the database into the DevOps workflow is how you maintain product velocity and optimize developer capacity to innovate.

The database plays a role in diminishing developer capacity. It often does not fit into the DevOps workflow, resulting in developer hours lost and bottlenecked product velocity. Database changes can result in problems that lead to downtime, performance issues, and other related issues that can impact the end-user experience.

Specifically, we estimate that **$1,153,308*** in developer hours can be lost on an annual basis per database due to inefficient database management. The impacts of database availability are just as severe with just 40 minutes of downtime resulting in **$224,000 in revenue**† loss. These numbers multiply as the number of databases an organization manages increases.

The companies that maintain a competitive edge are the ones that effectively manage the database. Incorporating the database into the DevOps workflow is how you maintain product velocity and optimize developer capacity to innovate.

In this document, we dive into the importance of removing the database as a blocker.

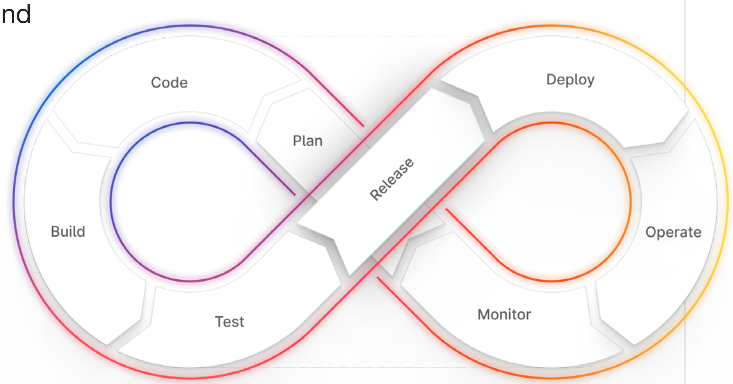
* This dollar amount was derived by taking some of the studies and metrics referred to throughout this document, including developer time spent managing infrastructure, the average time a developer works per week, and developer salary. The full methodology behind the calculation can be read in depth in the section titled "Diminishing developer capacity: The database's role" on page 8.

† Based on Gartner's estimate in their blog post "The Cost of Downtime".

Complex infrastructure hinders developer capacity

Pure product velocity has been a key driver of competitive advantage, with companies able to innovate by releasing new and improved versions of their products regularly.

This is why DevOps practices have grown in importance — they are designed to help organizations deliver software more quickly and efficiently, improving their time to market and their ability to respond to new market opportunities. By automating the build, test, and deployment process, DevOps practices enable organizations to release new versions of their product more frequently and without sacrificing quality, giving them a competitive edge in the market.



In today's market, delivering software rapidly is still a major driver of innovation. However, companies are now also facing the challenge of finding new technologies that allow them to work with their data more impactfully and effectively, as this is often one of the most difficult parts of building new products. This is where digital transformation comes into play.

Investing in new technologies and processes can help improve product velocity and allow companies to leverage their data to build better products, but it can also be a double-edged sword if not managed correctly.

As data grows, employees may be tempted to invest in specialized tools to manage the data for specific tasks. However, this approach can create additional challenges when it comes to maintenance and upkeep, absorbing developer capacity and making it difficult for your existing technical talent to focus on innovation-driving activities. Instead, the aim should be to adopt genericized tools that are performant, scalable, flexible, and capable of serving individualized use cases while maintaining a single source of truth.

The developer

Developers are a major contributor to innovation at a company — developer talent alone is projected to contribute $51,000 per developer to global GDP.* This is because they build and ship the very features that create differentiation from the rest of the market.

When infrastructure absorbs developer capacity, it directly robs your developers of the ability to integrate data into your product velocity.

Existing developer talent is now less able to leverage existing data to inform and ship new features that can improve the user experience, increase customer engagement, and generate revenue.

More tangibly, it can result in a reduction of **$300 billion in GDP contribution globally due to developer inefficiency**.*

Complex infrastructure and its management drains developer resources:

41.1 hours

Average developer work week*

13.5 hours

Time spent addressing tech debt weekly[†]

27%

Time spent maintaining existing data, applications, and systems[‡]

* Stripe "The Developer Coefficient" (2018)   [†] StackOverflow Developer Survey (2020)   [‡] Censuswide Survey (2022)

Diminishing developer capacity: The database's role

This problem becomes even more exacerbated when we take a look at the database as a component of the complex infrastructure that impacts developer capacity.

Some companies say it takes 7+ months to get a new feature or application out of the door.

Even when your developers are spending time developing new features, **57% of all application changes require a corresponding database schema change**.* These often take more than an hour to resolve. As you successfully increase product velocity, you also increase infrastructure maintenance as a byproduct.

Higher product velocity equates to more deployments, which often demand more database work. Here are a few examples of infrastructure work that needs to be covered as product velocity increases:

- Manually reviewing, validating, and pushing schema changes
- Monitoring and debugging query performance
- Enabling and scaling caching
- Ensuring high availability and data integrity
- Maintaining security and patching vulnerabilities

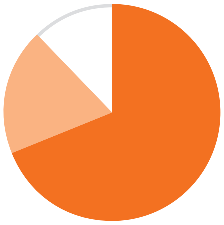
> **The hours spent performing this work multiply with the number of databases that you have, slowing development cycles even further.**

Some companies say it takes more than 7 months to get a new feature or application out of the door, and those with 20+ databases experience even more hindrance. So, as your business grows, you encounter a massive scaling opportunity. The more you reduce this work per database, the greater efficiency is achieved.

* "State of Database Deployments in Application Delivery" survey (2019)

Lost efficiency is a challenge commonly faced by many, but the much scarier challenge is that with increased complexity and scale comes the potential for mistakes which can directly lead to downtime, performance issues, and other related issues that can impact the end-user experience.

**Of those experiencing database change errors,
88% of professionals say it took more than an hour to resolve:***



69% Within a day ●

19% More than a day ●

12% Within an hour ○

No amount of speed can ever justify downtime. If your increased product velocity results in higher incident rates, you may have to slow down and start over. This is why it's crucial to consider the workflow around the database early.

What's more, the world of databases has practically been untouched by DevOps practices. A critical part of DevOps is being able to continually deploy code, which enables businesses to move quickly and iterate on their product. This is especially important in today's fast-paced and competitive market, where the ability to respond to new opportunities with speed is a major differentiator. While the benefits of DevOps practices have been well-documented, the database has been left out of this workflow.

No amount of speed can ever justify downtime.

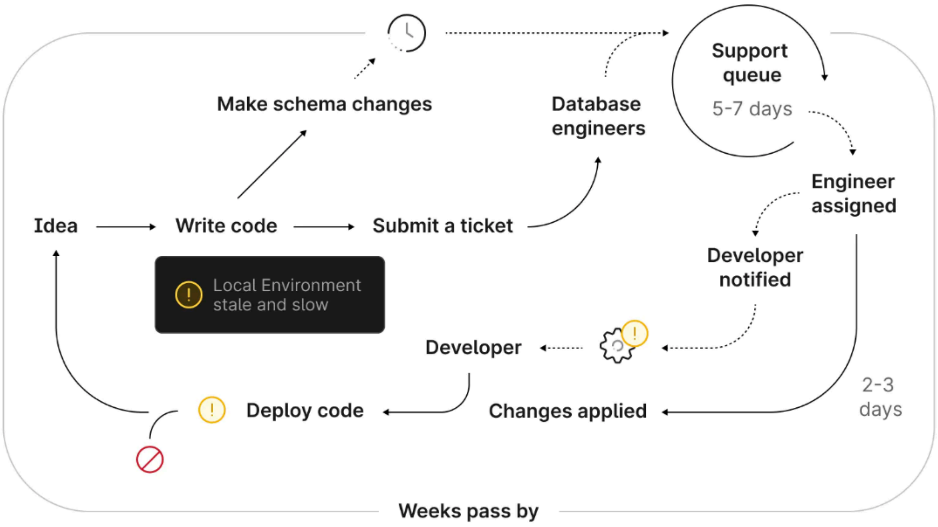
Databases have been historically left out of the DevOps cycle because of their stateful nature. The consistency of a database is critical to maintaining a stateful application and thus, it's not as easy to test and deploy changes in the ways that other components of your stack can. Teams are left tip-toeing around the database to prevent downtime or data loss and because of this, the database is often managed completely separately from the rest of the application stack to minimize risk.

* "State of Database Deployments in Application Delivery" survey (2019)

PlanetScale is the first database to allow you to roll back the state of your database alongside a deploy, fully enabling the DevOps lifecycle.

With the current state of legacy databases, engineers may need to apply database operations, such as schema changes, out of band of the typical CI/CD process, which can add significant time and unnecessary complexity to the process. In more complex or higher-scale environments, engineers may even need to open a ticket for a Database Administration team to apply schema changes to avoid downtime or data loss.

In contrast to the DevOps cycle above, the schema change process looks like:



This equates to more time allocated towards administering the database, which results in lost opportunities to innovate and can add weeks to a process that should be fast.

More time in database admin === Lost opportunities to innovate

Another component of database administration is maintaining security and mitigating vulnerabilities. Security patches and updates are important as they fix vulnerabilities and address other highly important security issues where user and other sensitive data is kept. Failing to do so can leave systems and networks vulnerable to attack, as we've seen in recent examples at LastPass and NHS Supplier Advanced.

LastPass exposed encrypted password vaults when their cloud storage was accessed and NHS Supplier Advanced suffered a ransomware attack that led to extended downtime and user data loss. For LastPass, this resulted in a permanently diminished reputation and the loss of user trust, while NHS Supplier Advanced lost revenue and critical data that impacted their patients and customers.

$224,000+

lost revenue during 40 minutes of downtime

The consequences of such attacks can be severe and can have a significant impact on the organization and its customers, including downtime and reputation loss. Resources must be allocated to prevent this, but this is another activity that can pull developers from tasks that directly contribute to innovation.

$1,153,308

annual opportunity cost of database administration*

Overall, time allocated to database administration equates to nearly **$1,153,308*** in opportunity cost per database annually. This number multiplies as the number of databases an organization has increases. This issue becomes even more exacerbated when we look at the revenue implications of downtime. With a conservative estimate of just 40 minutes offline, your business can **lose $224,000 in revenue** based on Gartner's estimate.[†] Even if your business loses much less, there are serious implications of database change errors.

* **Methodology**: With the metrics previously described in this document (1. The average time a developer works per week at 41.1 hours, and 2. 27% of developer time spent managing infrastructure includes administrative work), we concluded 11 hours as the average hours per week a developer spends managing infrastructure. We performed the following formula, where the hourly cost of developer hours is based on the average salary of a software developer in California:

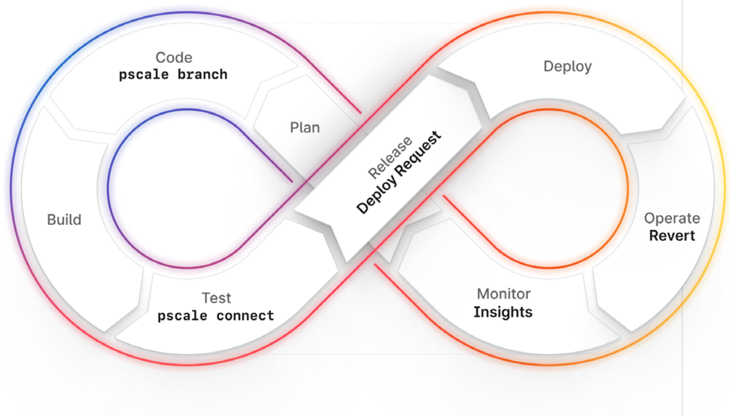
number of databases × weekly developer hours × hourly cost of developer hours = opportunity cost lost per week

[†] Based on Gartner's estimate in their blog post "The Cost of Downtime".

The challenges of managing databases as part of complex infrastructure can significantly impact a company's ability to innovate and move quickly. It's clear that finding ways to more effectively manage databases and incorporate them into the DevOps workflow can be a key driver of business success.

Increasing developer capacity with a modern MySQL platform

Neglecting the workflow around databases has a serious impact on the business. Technologies that promote complex, inefficient, and unreliable database practices can lead to costly downtime, reduced developer capacity, and overall decreased profitability. Those excelling in today's market are adopting technologies that drastically reduce the developer hours spent managing the database while eliminating downtime and speeding up the deployment process.



Database DevOps

We're just now getting to a point where companies can include the database in the DevOps workflow in a very meaningful way. There's a new cutting-edge opportunity to greatly improve developer productivity and ensure business continuity. To achieve high product velocity, avoid downtime, and maintain security standards, you need a database platform that enables developers to easily:

- Create a database without additional provisioning or planning
- Spin up isolated testing environments for schema changes
- Safely automate reviewing, validating, and pushing schema changes into production
- Branch database schema like application code
- Prevent downtime with built-in safeguards
- Debug and monitor all running queries

Simplifying your stack

The database should simplify your stack as opposed to adding unnecessary components that increase the complexity of managing it.

A modern database minimizes the need for specialized services and tools and their associated maintenance costs.

This simplifies database management for teams that should be focused on building and shipping new products and features, promoting efficiency and organization. This includes tools designed for:

- Data caching
- Backup and restores
- Online schema changes
- Load balancing and sharding
- Query monitoring and debugging

Reliable and scalable MySQL

There are many impactful benefits of adopting a MySQL database. One is the ability to lift and shift to any other MySQL database solution. This avoids vendor lock-in and gives your business the flexibility to migrate to another database without having to worry about complexity.

MySQL itself is a mature and reliable technology that is commonly used by hyper-scalers, including Uber, Slack, and many more.

When paired with Vitess, the open-source technology built to scale YouTube, it is able to handle high volumes of traffic, data volume, and scale without failure.

The table stakes

All databases should ensure high availability, data integrity, and that your security needs are met.

PlanetScale has developed a MySQL-compatible serverless relational database that improves the workflow around schema changes, query monitoring, testing, and more.

It's the only database platform that is built on the proven scalability of Vitess, the open-source technology built to scale YouTube.

Companies like MyFitnessPal and Barstool Sports have upgraded their database to PlanetScale for the very reasons explained above.

BARSTOOL SPORTS®

Barstool Sports saved millions avoiding downtime

Barstool Sports was suffering from repeated downtime with their previous cloud database solution. In one instance, **a 45-minute outage lost them a couple of million dollars**. Solving this issue was doable in-house, but it would require more time and engineering capacity. During one of their events, a query started misfiring and they had another complete outage. This was the moment they switched to PlanetScale and began saving time and money on database management. "Honestly we did not anticipate the savings from a time or money perspective. In the end, we saved 20-30% by switching to PlanetScale." Barstool says the added benefits of working with PlanetScale just keep coming. PlanetScale has opened up Barstool's engineering capacity and removed the database as a blocker.

> "In the end, we saved 20-30% by switching to PlanetScale."

Andrew Barba | CTO, Barstool Sports

Overcoming the developer capacity dilemma myfitnesspal™

MyFitnessPal got back $4.6 million in developer capacity with PlanetScale

Before PlanetScale, MyFitnessPal had **3 to 4 full-time staff working with the database**. Instead of spending time working with the data to better serve their end user, technical talent was engineering around schema migrations and maintaining the server components. This was impacting the company's ability to focus on working with their data to improve their product. The resources allocated to database management equate to upwards of **$4.6 million in opportunity cost***. Since switching over to PlanetScale, MyFitnessPal has increased its developer capacity and has enabled its engineers to deploy database schema like they do code.

> **"We want PlanetScale to bring to MyFitnessPal what Kubernetes brought to application delivery and deployment."**

Chris Karper | Engineering Director, MyFitnessPal

* Calculated using the same method defined in the "Diminishing Developer Capacity: The Database's Role" section.

Innovate with PlanetScale

As the volume and complexity of data grow at accelerated rates, infrastructure will continue to become needlessly complex, draining developer capacity and hindering the business's ability to grow and innovate. If left unaddressed, complexity will destroy valuable developer productivity, resulting in downtime and other issues that can hinder your ability to innovate and compete in the market.

The database doesn't have to contribute to this. It can help simplify components of your infrastructure without having to compromise on any trade-offs.

Adopting a modern MySQL database platform like PlanetScale can allow your organization to give your developers back their time to innovate.

PlanetScale is a company of builders who have lived and breathed these challenges while scaling today's largest and fastest-growing companies like GitHub, Facebook, Google, and more. Our platform is built by these folks to address all of the issues in a balanced and simplistic way.

It's the database they wish they had. And now, we bring it to you.

Get started today with PlanetScale, the most reliable way to scale your MySQL database in the cloud.

Call us at 1-408-214-1997 or send an email to sales@planetscale.com.

PlanetScale