

WHITEPAPER

The ROI of Compliant Database DevOps



The ROI of Compliant Database DevOps

Contents

Maximizing the value of database development	3
Understanding the real business benefits of DevOps	4
Viewed through the CEO lens	5
Viewed through the CIO lens	6
Viewed through the IT Manager lens	7
Adding data privacy to the lenses	7
Summary	8
Calculating the \$ value of Compliant Database DevOps	9
1. The cost of software development	9
2. The cost of implementing processes	9
3. The savings gained	10
4. The \$ value realized	10
5. The business benefits revisited	10
Defining a compliant database development process	11
Working example: standardized team-based development	14
Working example: automated database deployments	16
Working example: monitoring performance & availability	17
Working example: protecting and preserving data	18
Summary	20

Maximizing the value of database development

DevOps is moving into the mainstream. As the **2018 Accelerate State of DevOps Report** from DORA shows, there are big advantages for companies and organizations which adopt DevOps.

High performers can typically deploy changes, updates, and improvements 46 times more frequently, for example. Their change failure rate is also 7 times lower, and they can recover from failures when they do occur 2,604 times faster.

The database is also part of the same picture because the 2018 report, now in its fifth year, called out database development as a key technical practice which can drive high performance in DevOps. It revealed that teams which do continuous delivery well use version control for database changes and manage them in the same way as changes to the application. It also showed that integrating database development into software delivery positively contributes to performance, and changes to the database no longer slow processes down or cause problems during deployments.

This matches similar findings from Redgate's **2019 State of Database DevOps Survey**, which highlighted that the key drivers for automating the delivery of database changes are increasing the speed of delivery of those changes, freeing developers to do more added value work and, crucially, reducing the risk of losing data during the deployment process.

This last point is gaining more and more attention because a continuing stream of high-profile breaches and the introduction of the GDPR in Europe in 2018 has highlighted the need for companies to protect sensitive data. At first glance, this appears to be at odds with the desire to release changes to the database faster, but the automation and audit trails which DevOps introduces to database development can make compliance part of the process when planned and implemented in the right way.

Businesses want more than promises, however. They need to know the return on investment as well, with facts and figures that demonstrate what they will gain.

That could be a value-driven benefit – for example, unlocking the investment in development earlier by releasing to market faster, thereby realizing competitive advantage sooner. Or it could be a cost-driven saving, in terms of the \$ value of the hours of effort saved compared to traditional processes.

But what benefits do companies and organizations really seek when they begin to explore DevOps, what matters most to each of the stakeholders involved, and what is that \$ value waiting to be claimed?

This whitepaper reveals the benefits, and demonstrates how their appeal changes when viewed from the perspective of a CEO, a CIO, or an IT manager.

It then uses industry averages to provide a method of calculating the ROI of Compliant Database DevOps, and shows working examples of the ROI of introducing DevOps practices across the different stages of the database development and deployment process.

Understanding the real business benefits of DevOps

Ask 10 people involved in IT what the business benefits of DevOps are, and you're likely to receive a range of diverse answers. Partly, that's because DevOps is still relatively new. Partly it's because different stakeholders have their own particular focus. What matters to a CEO, for example, may well be at odds with the viewpoint of a developer.

To gain a deeper understanding of the benefits of DevOps, David Linwood, a highly experienced IT Director, undertook an MSc research project to discover the key success factors of DevOps. Over a six month period, he reviewed 88 DevOps papers, articles, and publications from industry commentators, and compiled a list of the most cited metrics. He then compared the views of the commentators with those expressed by IT professionals who actually practice DevOps.

The results offer a fascinating insight. The top seven benefits of DevOps, according to those who are involved in, or are introducing DevOps, are:

- The faster speed and lower cost of a release
- Improved operational support and faster fixes
- Faster time to market
- Higher quality products
- A lower volume of defects
- Improved frequency of new releases and features
- Good processes across IT and teams, including automation

What's interesting here is how these views differ from the most important benefits according to commentators. The faster speed and lower cost of a release, for example, was commented on only rarely by commentators, but was seen to be the joint first concern of practitioners. Similarly, commentators appeared to give little weight to improved operational support, but this was also the joint first concern of practitioners.

For the first time, however, a practical picture has emerged of the benefits that demonstrate the ROI of DevOps. Companies and organizations will, in general, benefit from the seven major advantages listed.

While most people would agree, David Linwood went deeper and added another aspect to his research, by looking at the benefits from the perspective of the different stakeholders involved.

CEOs, for example, are concerned about lower costs and higher revenues. CIOs, on the other hand, are more interested in cross-team collaboration. For managers and team leaders, higher performance and a reduction in defects are the major focus.

This enables the ROI of DevOps, in terms of the value-driven benefits to be gained, to be viewed through what David Linwood calls three different lenses.

Viewed through the CEO lens

The business outcome of DevOps is the CEO's area of interest. What's important to him or her, and the rest of the board, is how any investment in DevOps will lead to higher revenue and/or profitability. Whether that's delivered through tangible cost reductions, faster speed to market, or other improvements in business performance, ultimately the CEO will be interested in the value that this change can deliver to the organization as well as customers. And that value often varies according to the nature of the organization.

Preserving value is equally important. In its 2015 report, [DevOps and the Cost of Downtime](#), IDC calculated that, on average, infrastructure failures cost large enterprises \$100,000 per hour. The cost of critical application failures, meanwhile, was even higher, at \$500,000 to \$1 million per hour. So investments that can be proven to help minimize the financial risk caused by catastrophic IT failures will quickly capture board-level attention. That's before you even take into account the potential reputational damage, and the subsequent impact on shareholder value, caused by high-profile incidents of downtime or data breaches.

From a CEO perspective, the most important benefits are:

- Faster time to market
- Higher quality products
- A stable, reliable IT infrastructure
- Higher customer satisfaction
- Lower costs
- Improved business efficiency
- Higher revenue

Viewed through the CIO lens

At the next level down, the focus of the CIO or Head of IT is on the ingredients for success. How processes can be put in place to increase the throughput of the IT department, for example. Or how skilled IT staff can be recruited and retained to deliver quality services to the business.

The 2018 State of DevOps Report found that high performing organizations practicing DevOps get the most value-add time out of their days and spend the least amount of time doing non-value-added work. Not only was their throughput higher but spending more time on enjoyable, challenging development rather than rework and fixes also led to higher employee satisfaction levels.

If this sounds too intangible, what about measuring and quantifying the cost to the business of recruiting and training new members of staff? [A study by the Center for American Progress](#) found that the typical cost of turnover is 21% of an employee's salary. From a CIO perspective, the most important benefits are:

- Improved operational support and faster fixes
- Good processes across IT and teams, including automation
- Increased team flexibility and agility
- Happier, more engaged teams
- Cross-skilling and self-improvement
- Collaborative working
- Respect from senior management
- Rewards and feelings of success
- Freedom to experiment

Viewed through the IT Manager lens

Down in the engine room, for team lead roles such as IT Managers or Technical Leads, the focus is more firmly on the output performance of the IT department or team. They care about metrics like the speed of deployment or number of new releases delivered by their teams. They're also more interested in the ability to reduce defects, decrease downtime, or improve time to recovery, perhaps reflecting the pressure on them to maintain what is a core function for the business.

This view is reflected in the 2019 State of Database DevOps survey, which found that the greatest drawback identified with traditional siloed database development is the increased risk of failed deployments or downtime when introducing changes. This is closely followed by slow development and release cycles and the inability to respond quickly to changing business requirements.

From an IT Manager perspective, the most important benefits are:

- The faster speed and lower cost of a release
- A lower volume of defects
- Improved frequency of new releases and features
- Application performance
- The mean time to recovery (MTTR)
- Change failure rate
- Number and cost of resources
- Number of unused features in production

Adding data privacy to the lenses

David Linwood's research was undertaken prior to the enforcement of the General Data Protection Regulation (GDPR) in Europe, and before scandals like the [Facebook/Cambridge Analytica debacle](#) hit the headlines. Since then, business concerns about compliance with the legislation, as well as increasing unease from consumers and customers about the way their personal data is stored and processed, has resulted in data privacy and protection becoming a boardroom issue as well as an IT challenge.

It has also re-focused the attention of companies and organizations on ensuring they are compliant with existing legislation like the Health Insurance Portability and Accountability Act (HIPAA), Sarbanes-Oxley (SOX), and the Payment Card Industry Data Security Standard (PCI).

And perhaps unsurprisingly, it has prompted similar legislation to be proposed at both a national and local level. India's far-reaching Personal Data Protection Bill 2018 (PDP Bill) has been joined by the California Consumer Privacy Act (CCPA) and New York's Stop Hacks and Improve Electronic Data Security Act (SHIELD), among many others.

One common theme is emerging, which matches one of the major tents of the GDPR. The GDPR applies to any company or organization which handles the personal data of European citizens, even when the company or organization is based outside Europe. Similarly, compliance with the PDP Bill, the CCPA and SHIELD is a requirement for any company or organization storing or processing the data of the citizens the regulations protect.

As a direct consequence, compliance with one or more regulations will become increasingly common for many companies and organizations. This is an important point to make because DevOps encourages the release of small changes, often, and changes to front-end applications often mean the databases at the back end need to be updated as well. The very databases which store the personal data that needs to be protected.

Fortunately, the automation and audit trails which DevOps introduces can help companies and organizations remain compliant with data protection legislation, while also speeding up development and releases.

Where DevOps helps achieve compliance, a further benefit should therefore be added for CEOs, CIOs and IT Managers. It's relatively straightforward to see what that benefit would be, as viewed through each of their lenses.

- CEOs will want to know they can demonstrate compliance with data privacy regulations and thus minimize the risk of harming their business, reputation, brand and share price, all of which they and the Board are ultimately accountable for.
- CIOs will need reassurance that audit trails are in place to meet compliance requirements and that they can respond to any situation, including data breaches, in a consistent, repeatable and robust manner.
- IT Managers will be concerned with how they can meet and maintain compliance requirements, keeping data safe while still being able to deliver value faster.

Summary

When talking about how the ROI of DevOps can be measured in terms of the business benefits and added value to be gained, it's worth reflecting that there is not one answer. Instead, the answer depends on who is judging the value of those benefits. For any DevOps initiative, the concerns of each of the stakeholders involved should therefore be considered, including where relevant those relating to protecting the privacy of personal customer data.

Calculating the \$ value of Compliant Database DevOps

While considering the business benefits to be gained is one way to judge the return on investment of database DevOps, the more tangible financial benefit also needs to be considered.

A faster time to market may be a good customer story, but with no directly measurable monetary return, the value of DevOps can still be questioned. Which raises the question of how any cost savings should be calculated.

1. The cost of software development

The first step is to calculate the hourly cost of software development itself. The generally accepted method here is to take the average annual salary of a software developer, add a multiplier to account for benefits and employer costs, and then divide that figure by the number of working hours in a year to gain the hourly cost.

Using [data from the US Bureau of Labor statistics](#), the average annual salary of a software developer in the US is a little over \$100,000. Multipliers for benefits vary, but range from 30% to 50%. We'll take the lower percentage of 30%, resulting in an average annual cost per developer of \$130,000, and assume the number of working hours in a year is 2,000.

With these figures, the hourly cost of software development is: $\$130,000 / 2,000 = \65

In the UK, the average annual salary of a software developer in the [Hays 2018 UK Salary & Recruiting Trends Guide](#) is around £50,000 and, while the cost of health benefits is lower, there is often a higher provision for pensions. Using the same multiplier of 30%, the hourly cost is therefore £32.50.)

Across Western Europe, according to O'Reilly Media's [2017 European Software Development Salary Survey](#), the median salary is \$60,000 (the survey uses US\$ to avoid confusion across currencies). In Eastern Europe, the average falls to \$28,000.

2. The cost of introducing processes

The second step is to look at the cost of introducing processes. In terms of the database, this could range from standardizing team-based development to automating database deployments, and from monitoring performance and availability to protecting and preserving data.

At each step, some working practices will change, but one common thread runs through all of them: the new tooling that will need to be introduced to enable automation. As the State of DevOps Report points out: high performers report the lowest amount of manual work across all practices – and therefore the highest level of automation.

This is even more important considering IDC's DevOps and the Cost of Downtime report, mentioned earlier, which found that IT organizations that have tried to custom-adjust their current tools for DevOps practices have a failure rate of 80%.

Alongside any new tooling, the cost of acclimatizing to the adjusted way of working also needs to be included to reflect the true investment necessary.

3. The savings gained

The third step is to calculate the savings in time gained by adopting a particular area of DevOps. This will vary across organizations and needs to be more precise than quoting a broad benefit such as the change failure rate being seven times lower. While this is a vital element, as detailed in the State of DevOps Report, it is not yet a directly attributable \$ value.

That value needs to be expressed more accurately so that the real financial advantage to the business as a whole can be recognized.

4. The \$ value realized

Finally, the attributable \$ value can be calculated by comparing the savings gained with the cost of introducing the process. This should be calculated over one year, to demonstrate the immediate impact in the current financial year, and three years, to show what further savings can be realized in the longer term, once the initial cost of adoption has been accounted for. This ROI monetary value can also be converted to the ROI percentage using the following formula:

$$\frac{((\text{Total hourly savings} \times \text{hourly cost}) - \text{Cost of introducing process}) \times 100}{\text{Cost of introducing process}}$$

5. The business benefits revisited

Just as the business benefits should not be considered alone without a tangible ROI in terms of the cost savings realized, so they should be included here, along with how any relevant data privacy and protection concerns have been satisfied. That way, a complete picture can be gained of the total ROI.

Different organizations will seek different business benefits to measure the introduction of DevOps against, so the key is to return to the list of benefits and choose the most appropriate from each of the lenses so that every stakeholder can be satisfied.

Defining a compliant database development process

Before we look at some examples of the return on investment that can be gained from adopting Compliant Database DevOps, we should consider the database development process itself. As can be seen in the diagram below, there are four stages, three of which match those of application development.



Encouraging collaboration across and within teams is standard when introducing DevOps to application development, as well as automating processes wherever possible and monitoring both performance and availability. When database development is included in DevOps, there is an added requirement to protect personal data so that the faster speed at which database changes can be released does not compromise the safety and security of the data they contain.

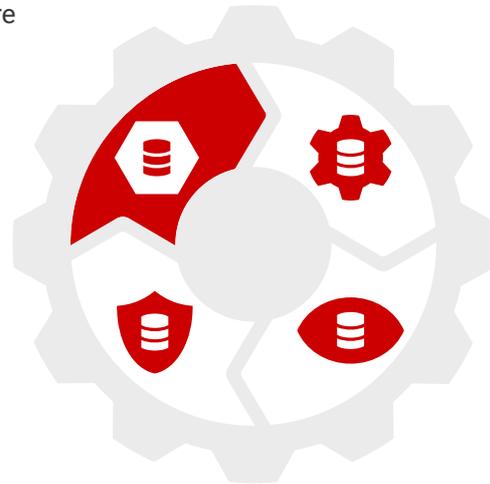
It's worth looking at each of the stages in turn, because as software development gathers pace, with companies and organizations moving to release features faster and more efficiently, so database development has moved from being the sole preserve of the DBA. Indeed, the 2019 State of Database DevOps Survey from Redgate revealed that 75% of application developers typically build database deployment scripts, while 47% deploy database changes to production.

Standardize team-based development

With IT teams under pressure to deliver software more quickly, it's a challenge to ensure the consistency and quality of code during database development. Even more so for large teams of application and database developers with different ways of working, or geographically-distributed teams in different locations and time zones.

Without the right processes or tooling in place, precious time is wasted on tedious rework. Even a relatively easy innovation like introducing a tool to standardize the way database code is developed and formatted, and apply static code analysis as it is written can reap big rewards. Code becomes shareable. Errors in code are highlighted immediately. Developers can add to or amend code written by other developers with ease. Legacy code remains understandable code.

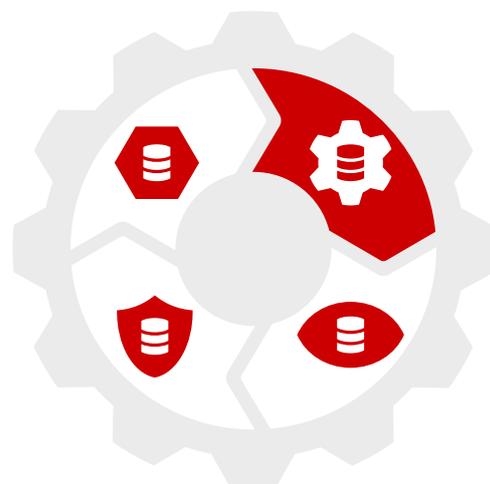
Similarly, introducing version control into the process means teams can work on different branches at the same time, while maintaining a single source of truth from which a database can be built from the code in the repository at any time.



Automate database deployments

Validating and testing code with an automated continuous integration process is becoming standard practice in application development, and it's entering the database arena as well. Each time a change is committed to version control, it triggers a process that builds the database and tests it to ensure the change is not a breaking change. This draws attention to problems much earlier in development and prevents errors occurring later on.

The deployment process can also be made transparent and safe by automating change management tasks like the creation of deployment scripts, and freeing the development team up for more valuable work. Importantly, with reliable, scalable, and repeatable processes in place for building, testing and releasing changes, companies can embed security best practice, and achieve data protection by design and default.



Monitor performance and availability

While the automation which DevOps introduces to many parts of database development minimizes errors and gives much better visibility across the whole process, there is a flipside. Instead of releasing changes to the database once or twice a quarter, changes can now be released at any time. It is therefore important to monitor systems constantly to understand if releases are causing a problem so that they can be fixed quickly.

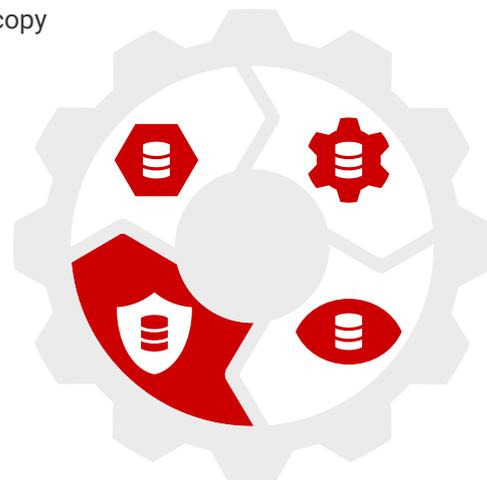


Performance monitoring will help by watching resource utilization (CPU, memory, I/O) changes over time, and determining the top ten or so worst performing queries, as well as responding to alerts about a drop in performance or an increase in deadlocks, and drilling down to the cause of the problem before it becomes an issue.

Monitoring for compliance also has a role to play by, for example, keeping track of failed logins and how many accounts are in the sysadmin group, as well as looking out for SQL injection attacks, changes in server and database settings, and modifications to permissions, users, and roles.

Protect and preserve data

In database development, having an individual copy of the production database to test proposed changes against and ensure they are not breaking changes is the preferred way of working for many developers. Those databases, however, contain the kind of personal data that needs to be protected to guard against data breaches.



The [2018 Data Breach Investigations Report](#) from Verizon, for example, revealed that 28% of breaches involved insiders, and this was particularly difficult to guard against. The data leaked in high profile breaches like those at Uber, Shutterfly, and the Veteran's Administration also came from environments outside Production, like Development and Testing, where security is often less strict.

Hence the rise in the interest in data masking, with Gartner's **2018 Market Guide for Data Masking** predicting the percentage of companies using data masking or practices like it will increase from 15% in 2017 to 40% in 2021.

Even when masked, the provisioning of frequently updated copies of the production database across development teams can, however, take up a lot of time, not to mention disk space. This is where the tried and testing virtualization technology built into the Windows operating system can help. There are now tools available which use the technology to create copies, or clones, of databases in seconds which, while only around 40MB in size for a 1TB database, work just like normal databases and can be connected to and edited using any program.

Used in conjunction with a masking tool, it means developers can work with copies of the production database which are truly representative of the original and retain the referential integrity and distribution characteristics. The personal data inside them, however, is protected, and an audit trail can be provided which demonstrates compliance with any relevant legislation.

Working example – standardized team-based development

To illustrate how the cost savings can be calculated, we'll begin with a straightforward example: the process of actually writing SQL code. This is often performed in SQL Server Management Studio, which has a tool called IntelliSense to make it easier.

There are third party alternatives to IntelliSense that have been developed to partly automate the writing of code and also encourage collaboration. SQL Prompt from Redgate, for example, autocompletes code, takes care of formatting, object renaming, and other distractions, and allows code to be shared and standardized across teams. It also has a static code analysis capability which identifies potential code issues and hidden pitfalls at the point code is written, and provides clear explanations and suggestions to improve the code.

At the basic and most measurable level, the tool speeds up coding. In a **test conducted by SQL Server MVPs** Steve Jones and Grant Fritchey, it was 50% faster than IntelliSense, and this is supported by independent reviews of SQL Prompt. Sean Van Der Walt, a Data Analyst at Multichoice, for example, says it has cut down the time spent writing SQL to a quarter or less, while consultant John Ek reports he is 30% more productive, so the 50% figure is a fair average.

The cost of a single SQL Prompt license, without any discount and including support and upgrades for 3 years, is \$539. Let's also add a 16 hour cost of learning to use it proficiently to gain the full advantages.

Being conservative, let's assume a developer spends 10% of the working day writing SQL code, which would equate to 48 minutes if the working day is 8 hours. Using SQL Prompt would save 50% of that time, or 24 minutes. Over one week, this would be 2 hours, and over one year this would equate to 100 hours. We have the figures. Now let's look at the savings to be gained:

Savings gained: 100 hours @ \$65 per hour	\$6,500
Less cost of introducing the process: \$539 + 16 hours acclimatization @ \$65 per hour	\$1,579
Return on investment over year In percentage terms, this delivers an ROI of 312%	\$4,921
Return on investment over three years In percentage terms, this delivers an ROI of 1,134%	\$17,921

Now let's go back to the business benefits we discussed earlier and select some appropriate benefits of SQL Prompt through the CEO, CIO and IT Manager lenses:

- CEO lens: Lower costs
- CIO lens: Increased team flexibility and agility
- IT Manager lens: The faster speed and lower cost of a release

The result is a complete RIO for introducing SQL Prompt to the database development process. Each stakeholder is happy in terms of the business benefits delivered, and the \$ value of the ROI is demonstrated as well.

Working example – automated database deployments

We've seen how cost savings can be gained when provisioning databases copies for use in development, but what if you were to introduce continuous integration and automated deployment for the database, just as you would for the application?

Automating the testing of database changes as part of a continuous integration process, for example, helps developers discover mistakes faster, makes fixing those mistakes easier, and provides opportunities for continuous learning and improvement. Similarly, automating change management tasks like the creation of database deployment scripts makes deployment processes safer and more transparent, and frees developers for more valuable work.

Perhaps most importantly, if you practice continuous integration and automated deployments for your application but leave your database behind, you won't realize the full advantages.

Software suites like those offered by Redgate contain a range of database development and deployment tools that plug into and integrate with the standard version control, continuous integration, and automated deployment tooling used for applications, making it easier to introduce.

A good example is major insurance company AFA Försäkring, which chose Redgate when it wanted to replace its manual database deployment processes and introduce automation across its 20-strong development team.

The team started by introducing version control for database development, before moving on to continuous integration and automated deployments. This transformed the way they work, speeding up the process while reducing errors – and doubling the number of deployments to production. Crucially, it resulted in substantial time savings at every stage of the database development process, freeing up the equivalent of at least one additional developer.

Let's look at the total savings gained across the team of 20 developers, with each requiring a two and a half day, or 20 hour, acclimatization period to get used to their new way of working.

Savings gained: One year of a developer's salary @ \$130k per year	\$130,000
Less cost of introducing the process: Software cost + 400 hours acclimatization @ \$65 per hour	\$76,000
Return on investment over one year In percentage terms, this delivers an ROI of 71%	\$54,000
Return on investment over three years In percentage terms, this delivers an ROI of 82%	\$174,000

The business benefits are different to those of introducing SQL Prompt:

CEO lens:	Improved business efficiency
CIO lens:	Increased team flexibility and agility
IT Manager lens:	Improved frequency of new releases and features

Working example – monitoring performance and availability

We saw earlier how monitoring databases becomes more important when introducing DevOps. The move from big, infrequent releases to small releases often means the environments in development, testing, staging, and production are under constant change.

Some measure of the level of that change is Skyscanner, which introduced database DevOps and went from releasing changes every six weeks to releasing them up to 95 times a day.

This is where a performance monitoring tool really comes into its own because, however effective the testing regime is, it's only when changes hit the production environment under real load that their true effect can be monitored. If problems do occur, time is at a bigger premium than ever because protecting and preserving data is crucial to any business.

Give the development team access to such a tool and the advantages increase further because they can correlate their own changes to any issue that arises, discover why it happened, and apply continuous improvements.

When Mamas & Papas, a UK-based retailer and manufacturer, was looking for a tool to monitor its large SQL Server estate, Redgate's SQL Monitor was chosen because of its

intuitive interface, configurable alerts, and built-in integration with Slack. This enabled the team to move away from mandatory manual checks every morning to one where alerts are automatically posted to a dedicated Slack channel, and monitor screens show at a glance any issues that are occurring. This has saved at least two hours a day and also changed the culture of the team and made their jobs more enjoyable and productive.

Given the requirement to introduce the whole team to SQL Monitor, let's calculate the financial return on investment by assuming a total acclimatization period of one week, or 40 hours, and total savings of two hours a day for 250 business days a year.

Savings gained:	\$32,500
500 hours @ \$65 per hour	
Less cost of introducing the process:	\$27,719
Software cost + 40 hours acclimatization @ \$65 per hour	
Return on investment over one year	\$4,781
In percentage terms, this delivers an ROI of 17%	
Return on investment over three years	\$69,781
In percentage terms, this delivers an ROI of 252%	

The business benefits here reflect the different needs SQL Monitor satisfies:

CEO lens:	A stable, reliable IT infrastructure
CIO lens:	Improved operational support and faster fixes
IT Manager lens:	A lower volume of defects

Working example – protecting and preserving data

Enabling developers to test changes against an up-to-date copy of the production database is an important part of a database DevOps process. It means they can find errors or performance issues earlier, before changes are released, and prevent problems deployment.

Those same production databases often, however, contain sensitive customer data which needs to be masked, and provisioning multiple copies to individual developers to test their changes against can take up a lot of time as well as disk space. Using anonymous data or limited data sets is an option but it means testing changes against a database that is

neither realistic, nor of a size where the impact on performance can be assessed. SQL Provision from Redgate resolves the problem by masking sensitive data with data that is truly representative of the original and retains its referential integrity and distribution characteristics. It then uses the virtualization technologies built into the Windows operating system to create copies, or clones, of the masked database in seconds. The clones are only around 40MB in size for a 1TB database, work just like normal databases and can be connected to and edited using any program.

The cost of SQL Provision for 3TB of production data is \$19,890 annually. Because it introduces a new way of provisioning database copies, there will also be a time cost before it becomes a natural element of the development process. Let’s add two weeks, or 80 hours, to introduce it.

In terms of the time it saves, consider KEPRO, a leading healthcare organization which needed to safeguard protected health information (PHI) to meet the requirements of HIPAA, which mandates industry-wide standards for the confidential handling of healthcare information.

Before introducing SQL Provision to the database development process, the provisioning of database copies was taking over 20 hours a week and masking sensitive data was a real problem. As soon as SQL Provision was introduced, it saved between 15-20 hours a week in provisioning database copies as well as many terabytes of disk space, and also now provides an audit trail to demonstrate compliance with HIPAA. Taking a conservative estimate of 15 hours a week, the savings in time add up to 750 hours a year. The total \$ savings and the ROI can therefore be calculated as follows:

Savings gained: 750 hours @ \$65 per hour	\$48,750
Less cost of introducing the process: Year 1 Software cost + 80 hours acclimatization @ \$65 per hour	\$27,070
Return on investment over one year In percentage terms, this delivers an ROI of 80%	\$21,680
Return on investment over three years In percentage terms, this delivers an ROI of 107%	\$75,440

Now let’s go back to the business benefits we discussed earlier and select some appropriate benefits of SQL Provision through the CEO, CIO, and IT Manager lenses:

- CEO lens: Compliance with data privacy regulations, in this case HIPAA
- CIO lens: Audit trails in place to meet compliance requirements
- IT Manager lens: The ability to keep data safe, yet deliver value faster

Summary

This whitepaper has demonstrated that a viable return on investment for Compliant Database DevOps can be established in two ways. First, by viewing the business benefits to be gained through the lenses of the different stakeholders involved. Second, by showing the actual \$ return, in both the short and medium term, using real-world working examples.

It should be noted, however, that different companies and organizations will be at different stages in their DevOps journey. Some, for example, will be seeking to automate database deployments, while others will have a requirement to protect and preserve data.

If you're exploring Compliant Database DevOps, the following resources may also help:

Compliant Database DevOps whitepaper

With new data protection laws coming into play and consumers more aware than ever before of how their privacy can be compromised, there is now a requirement for companies to adopt a compliant DevOps approach. One where protecting data is baked into the software development process from the beginning, by combining the agility of DevOps, the desire to include the database in DevOps, and the necessity to secure data throughout development.

Redgate's Compliant Database DevOps whitepaper outlines how organizations can achieve this standardizing team-based development, automating deployments, monitoring performance and availability, and protecting and preserving data.

The 2020 State of Database DevOps report

Now in its fourth year, the 2020 State of Database DevOps report provides insights into multiple tactics that can improve the quality of database code deployment, and can help build the foundations of your business case for introducing DevOps.

SQL Server Database Provisioning Report

Provisioning copies of databases for use in development and testing carries many challenges, including concerns around data security and regulatory compliance, as well as infrastructure issues like storage space.

But how important is database provisioning, who controls it, and how often is it needed? To find out the answers, Redgate conducted a survey among nearly 500 SQL Server professionals. The resultant SQL Server Database Provisioning Report reveals how organizations currently provision databases, identifies the challenges faced by database teams, and offers a unique glimpse into database provisioning in the real world.

About the authors



Matt Hilbert is a Technology Writer at Redgate Software and has over 20 years' experience writing about the latest innovations and advances. He has worked at some of the world's largest – and smallest – technology companies and is a regular contributor to technical publications.



Kate Duggan is a Marketing Manager at Redgate Software. She has worked in the technology sector for many years, both as an adviser helping tech companies commercialize their products, and in product marketing for software companies.



David Linwood is a highly experienced IT Director with over 25 years' experience across B2B, banking, financial services, and FTSE 100 corporate sectors. He recently completed an MSc in Management at Ashridge Business School, the research for which inspired this whitepaper.

You can read more about David's research in his series of LinkedIn articles at <https://www.linkedin.com/pulse/which-metrics-assist-devops-proving-its-roi-from-academia-linwood>