

WHITEPAPER

# The ROI of Database DevOps

## Maximizing the value of database development

DevOps is moving into the mainstream. As the 2017 [State of DevOps Report](#) from Puppet and DORA shows, there are big advantages for companies and organizations which adopt DevOps.

They can typically deploy changes, updates, and improvements 46 times more frequently, for example. Their change failure rate is also five times lower, and they can recover from failures when they do occur 96 times faster.

But what about the database? Where does that fit into the picture? Can it fit into the picture?

In terms of the advantages to be gained, they are as valuable for the database as they are for the application. The 2017 [State of Database DevOps](#) survey from Redgate revealed that the key drivers for automating the delivery of database changes are increasing the speed of delivery of those changes, and freeing developers to do more added value work.

It also found that 80% of companies plan to adopt DevOps across some or all of their IT projects within the next two years. So even if you're not doing it now, the chances are you will be exploring it soon.

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 database DevOps, and shows working examples of the ROI of introducing DevOps practices across the different stages of the database development and deployment process.

# Contents

|    |  |
|----|--|
| 2  | <b>Maximizing the value of database development</b>            |
| 4  | <b>Understanding the real business benefits of DevOps</b>      |
| 5  | Viewed through the CEO lens                                    |
| 5  | Viewed through the CIO lens                                    |
| 6  | Viewed through the IT Manager lens                             |
| 7  | Summary  |
| 7  | <b>Calculating the \$ value of database DevOps</b>             |
| 7  | The cost of software development                               |
| 8  | The cost of introducing processes                              |
| 8  | The savings gained   |
| 9  | The \$ value realized  |
| 9  | The business benefits  |
| 10 | <b>Defining the database development process</b>               |
| 13 | <b>Working example – development environment and practices</b> |
| 14 | <b>Working example – continuous integration and deployment</b> |
| 15 | <b>Working example – protecting and preserving data</b>        |
| 16 | <b>Summary</b>   |

## 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 2017 State of DevOps Report found that high performing organizations practicing DevOps spent 21% less time on unplanned work and unnecessary rework, and 44% more time on new work than their peers. Not only was their throughput higher but spending more time on enjoyable, value-added development 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 2017 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

## 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.

Over time, these benefits may also change. When the European Union's General Data Protection Regulation (GDPR) is introduced in May 2018, for example, the need to address audit and compliance issues may well enter the picture for those who manage data. CEOs will want to ensure compliance, CIOs will want a way of demonstrating compliance, and IT Managers will look for methods of maintaining compliance without affecting the performance of their team.

## Calculating the \$ value of 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 \$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 2017 survey](#) from Hays UK is £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 Europe, according to O'Reilly Media's [2016 European Software Development Salary Survey](#), the median salary is \$56,000 (the survey uses US\$ to avoid confusion across currencies), with the notable exception of Switzerland, which has salaries similar to the US. The average salary also reduces the further east across Europe you travel.*

## 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 development environments and practices, through continuous integration and deployment, 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 five 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. 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 the database development process

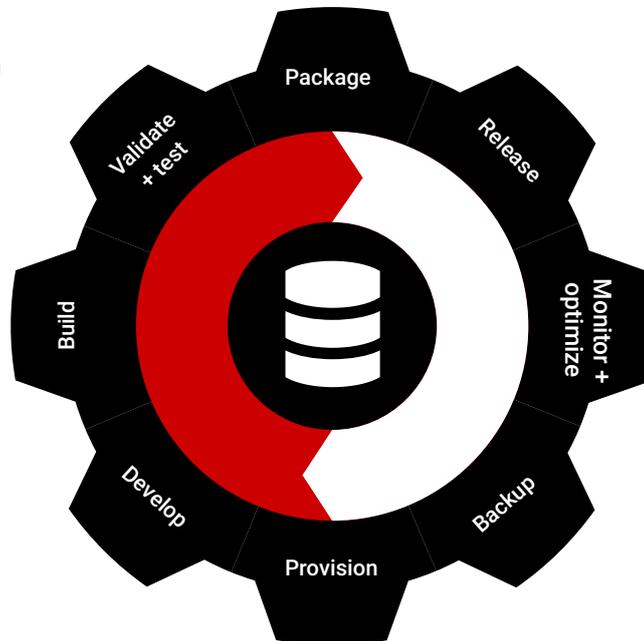
Before we look at some examples of the return on investment that can be gained from adopting database DevOps, we should consider the database development process itself.

As can be seen in the cog on the right, there are eight steps which broadly match those of application development.

It's worth looking at each of the steps, 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 2017 [State of Database DevOps](#) survey from Redgate revealed that 60% of application developers typically build database deployment scripts, while 39% deploy database changes to production.

The eight steps in the cog can be grouped into three stages, because adopting one step inevitably leads to a desire to adopt others, to make the most of this part of the DevOps process:

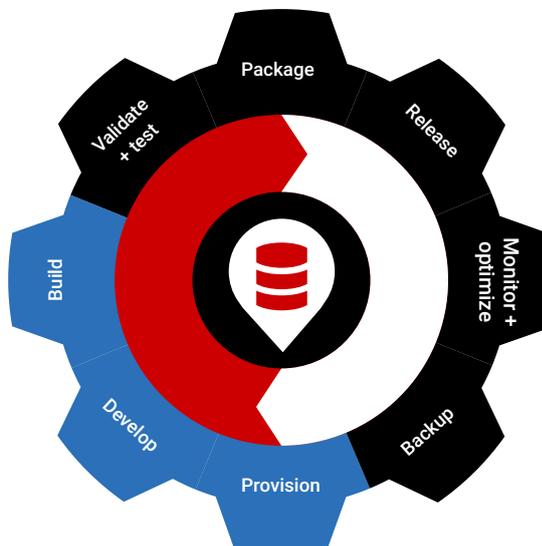
- Development environment and practices
- Continuous integration and deployment
- Protecting and preserving data



## Development environment and practices

This first stage is often the entry route into adopting DevOps. It's about encouraging good visibility among teams, with Development and Operations collaborating and sharing code, changes, or data. The consistent use of processes and automation also removes bottlenecks in the development process.

Automating the provisioning of database copies, for example, gives developers more freedom to test their changes. 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.

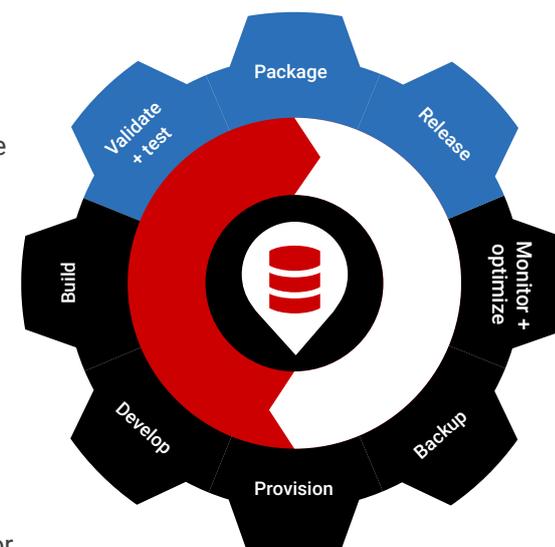


Even a relatively easy innovation like introducing a tool to standardize the way database code is developed and formatted can reap big rewards. Code becomes shareable. Developers can add to or amend code written by other developers with ease. Legacy code remains understandable code.

## Continuous integration and deployment

Validating and testing code with a 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 highlights problems much earlier in development and prevents errors occurring later on.

The continuous integration process can also be configured to produce a package for release, normally in NuGet format, which a release automation tool can use to make deployments faster and more reliable.



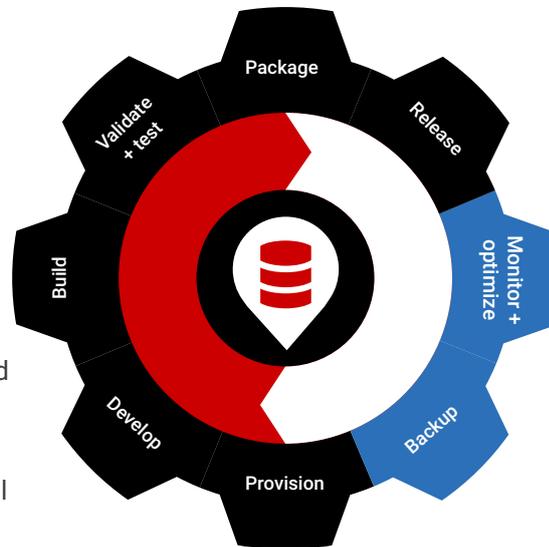
## Protecting and preserving data

This increased speed of delivery, along with the automation that DevOps introduces, makes monitoring an even more important element in the process.

The database moves from being a relatively static code base to one that changes little and often, with updates constantly rolling from development through testing and staging environments before being deployed to production.

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.

DevOps also stresses the requirement for a reliable backup process, as does the introduction of GDPR which has already been mentioned. The new regulation will add more weight to ensuring processes are in place to protect data and meet compliance and audit requirements by, for example, automating backups and access control.



## Working example – development environment and practices

The provisioning of realistic environments for developers to develop and test against is an essential part of a database DevOps process. Enabling developers to test changes against an up-to-date version of the database means they can find errors or performance issues earlier, before they make their way to production.

With production databases now up to 3TB or more in size, however, there are often problems in terms of time and space. Quite simply, it takes a long time to copy a database or backup, and it needs to be stored somewhere and refreshed frequently to keep pace with ongoing development.

There are, however, third party tools available, one of which is SQL Clone from Redgate. SQL Clone enables users to create full copies, or clones, of SQL Server databases and backups in seconds rather than hours, using only around 40MB of disk space per clone.

The cost of a SQL Clone small team license, which includes five user entitlements, along with support and upgrades for three years, is \$9,785. Because it introduces a new way of provisioning data 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 Moody's Analytics, which recently introduced SQL Clone to the database development process. Before its adoption, the two testers and three developers in the team were waiting an hour to be provisioned with database copies. With each tester needing copies two or three times a day, and each developer needing copies two or three times a week, it was slowing down development.

When SQL Clone was introduced, it reduced the provisioning time to 10 minutes for copies of all six databases involved. Taking a conservative estimate of using SQL Clone to provision four copies a day, this saves the company 200 minutes a day, resulting in yearly savings of 833 hours. The total \$ savings and the ROI can therefore be calculated as follows:

|  |           |
|--|-----------|
| <b>Savings gained:</b>                                   | \$54,145  |
| 833 hours @ \$65 per hour                                |           |
| <b>Less cost of introducing the process:</b>             | \$14,985  |
| Software cost + 80 hours acclimatization @ \$65 per hour |           |
| <b>Return on investment over one year</b>                | \$39,160  |
| In percentage terms, this delivers an ROI of 343%        |           |
| <b>Return on investment over three years</b>             | \$147,450 |
| In percentage terms, this delivers an ROI of 1,231%      |           |

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

|                  |   |
|------------------|---|
| CEO lens:        | <i>Faster time to market</i>                                    |
| CIO lens:        | <i>Good processes across IT and teams, including automation</i> |
| IT Manager lens: | <i>The faster speed and lower cost of a release</i>             |

## Working example – continuous integration and deployment

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 Redgate's SQL Toolbelt 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 the SQL Toolbelt 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 working with the SQL Toolbelt.

|   |           |
|---|-----------|
| <b>Savings gained:</b>                                    | \$130,000 |
| One year of a developer's salary @ \$130k per year        |           |
| <b>Less cost of introducing the process:</b>              | \$91,960  |
| Software cost + 400 hours acclimatization @ \$65 per hour |           |
| <b>Return on investment over one year</b>                 | \$38,040  |
| In percentage terms, this delivers an ROI of 41%          |           |
| <b>Return on investment over three years</b>              | \$298,040 |
| In percentage terms, this delivers an ROI of 324%         |           |

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

|                  |  |
|------------------|--|
| CEO lens:        | <i>Improved business efficiency</i>                    |
| CIO lens:        | <i>Increased team flexibility and agility</i>          |
| IT Manager lens: | <i>Improved frequency of new releases and features</i> |

## Working example – protecting and preserving data

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.

|  |          |
|--|----------|
| <b>Savings gained:</b>                                   | \$32,500 |
| 500 hours @ \$65 per hour                                |          |
| <b>Less cost of introducing the process:</b>             | \$26,100 |
| Software cost + 40 hours acclimatization @ \$65 per hour |          |
| <b>Return on investment over one year</b>                | \$6,400  |
| In percentage terms, this delivers an ROI of 24.5%       |          |
| <b>Return on investment over three years</b>             | \$71,400 |
| In percentage terms, this delivers an ROI of 274%        |          |

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

|                  |  |
|------------------|--|
| CEO lens:        | <i>A stable, reliable IT infrastructure</i>          |
| CIO lens:        | <i>Improved operational support and faster fixes</i> |
| IT Manager lens: | <i>A lower volume of defects</i>                     |

## Summary

This whitepaper has demonstrated that a viable return on investment for 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 improve their development environment and practices, while others will be in the middle of automating deployments.

Defining the return on investment also depends on what your aim is. We saw how AFA Försäkring automated its database deployment processes using the SQL Toolbelt and gained an ROI of 324% over three years. The SQL Toolbelt, however, contains 15 tools covering the whole database development and deployment process. A larger ROI is therefore waiting to be realized by, for example, using one of the other tools like SQL Monitor.

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

### [Solving Database Deployments whitepaper](#)

This whitepaper explains how you can extend DevOps practices to your SQL Server databases. By adopting best practices, and applying the right solutions, you can start to gain consistency across your applications and databases, increase efficiencies through your development and operations teams, and remove that last bottleneck.

### [The State of Database DevOps](#)

This report contains the results of a survey of over 1,000 SQL Server database professionals, across a range of industries and company sizes. We asked how many of them had adopted, or were planning to adopt, DevOps practices in their organization and how many of them had applied the same principles to their databases

### [Database DevOps Maturity Assessment tool](#)

Take a few minutes to complete our Database DevOps Maturity Assessment and you'll better understand how advanced your current processes are, receive recommendations for improvements, and see how your maturity level compares with that of your peers

## 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 the Marketing Manager and Interim Head of Marketing at Redgate Software. She has worked in the technology sector for many years, both as an adviser helping companies to adopt technology, and 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.

