SQL in the City Streamed
The State of Compliant Database DevOps
DevOps in Software

- Flickr – 10 deploys per day, 2009
- Etsy – 50 deploys per day, 2010
- Amazon – moved to AWS, deploy every 11.7 seconds
- Nordstrom's – from 22-28 week lead time to monthly
- Fidelity – From 2-3 days for a release to 1-2 hours
Key Findings - Year over Year

- **2016**
  - Deploy frequency: 200x
  - Lead time for changes: 2555x
  - Change fail rate: 24x
  - Total: 3x

- **2017**
  - Deploy frequency: 46x
  - Lead time for changes: 440x
  - Change fail rate: 96x
  - Total: 5x

- **2018**
  - Deploy frequency: 46x
  - Lead time for changes: 2555x
  - Change fail rate: 2604x
  - Total: 7x
Don’t Forget the Database

Integrating database work into software delivery **positively contributes to SDO performance**

Integrating database practices look much like integrating ops work in early days: communication, config management, including teams, visibility

Teams could think about this like “shifting left”
# New 2018 Performance Benchmarks

<table>
<thead>
<tr>
<th>Aspect of Software Delivery Performance</th>
<th>Elite&lt;sup&gt;a&lt;/sup&gt;</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Deployment frequency</strong></td>
<td>On-demand (multiple deploys per day)</td>
<td>Between once per hour and once per day</td>
<td>Between once per week and once per month</td>
<td>Between once per week and once per month</td>
</tr>
<tr>
<td>For the primary application or service you work on, how often does your organization deploy code?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Lead time for changes</strong></td>
<td>Less than one hour</td>
<td>Between one day and one week</td>
<td>Between one week and one month&lt;sup&gt;b&lt;/sup&gt;</td>
<td>Between one month and six months&lt;sup&gt;b&lt;/sup&gt;</td>
</tr>
<tr>
<td>For the primary application or service you work on, what is your lead time for changes (i.e., how long does it take to go from code commit to code successfully running in production)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Time to restore service</strong></td>
<td>Less than one hour</td>
<td>Less than one day</td>
<td>Less than one day</td>
<td>Between one week and one month</td>
</tr>
<tr>
<td>For the primary application or service you work on, how long does it generally take to restore service when a service incident occurs (e.g., unplanned outage, service impairment)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Change failure rate</strong></td>
<td>0-15%</td>
<td>0-15%</td>
<td>0-15%</td>
<td>46-60%</td>
</tr>
<tr>
<td>For the primary application or service you work on, what percentage of changes results either in degraded service or subsequently requires remediation (e.g., leads to service impairment, service outage, requires a hotfix, rollback, fix forward, patch)?</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
"It's actually the most difficult part of the pipeline to automate because of the risk of failure," he says. "You can easily roll back a front-end — you can't do the same for your database because any transactions that have occurred during the deployment could potentially be lost. So, it's a crucial, integral part of your DevOps pipeline and I challenge any company that claims to be doing DevOps that only does it for the front-end but their back-end is still done manually. To me, they're faking it."

- Donovan Brown, Microsoft
Database DevOps Plans

- **13%** Already adopted across all projects
- **39%** Already adopted across some projects
- **30%** Plan to adopt across some or all projects in the next 2 years
- **18%** Not adopted and no plans to adopt within the next 2 years
Why Compliant Database DevOps?

“I DON’T ALWAYS TEST MY CODE, BUT WHEN I DO – I DO IT IN PRODUCTION”
"Based on industry surveys, the number we typically cite is $5,600 p/minute, which extrapolates to well over $300K p/hour"

- Andrew Lerner, Gartner

Rollbacks. We rarely do them. In fact, I can’t remember ever having done one. We avoid them through the approach in general: we deploy small and often.

… Why roll back when you can roll forward?
"At Google, our philosophy is that “rollbacks are normal.” When an error is found or reasonably suspected in a new release, the releasing team rolls back first and investigates the problem second. "

Google
DevOps Brings Teams Together
DevOps Adds Value

• Improved Customer Experience
• Increased Customer Satisfaction
• Increased Employee Satisfaction
• Improved Governance
## More Organizational Value

<table>
<thead>
<tr>
<th>Time Spent</th>
<th>Elite</th>
<th>High</th>
<th>Medium</th>
<th>Low</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>NEW WORK</strong></td>
<td>50%</td>
<td>50%</td>
<td>40%</td>
<td>30%</td>
</tr>
<tr>
<td>Unplanned work and rework</td>
<td>19.5%</td>
<td>20%^a</td>
<td>20%^a</td>
<td>20%^a</td>
</tr>
<tr>
<td>Remediating security issues</td>
<td>5%</td>
<td>5%^b</td>
<td>5%^b</td>
<td>10%</td>
</tr>
<tr>
<td>Working on defects identified by end users</td>
<td>10%</td>
<td>10%^c</td>
<td>10%^c</td>
<td>20%</td>
</tr>
<tr>
<td>Customer support work</td>
<td>5%</td>
<td>10%</td>
<td>10%</td>
<td>15%</td>
</tr>
</tbody>
</table>
Compliant Database DevOps

- Standardize team-based development
- Automate database deployments
- Protect and preserve data
- Monitor performance & availability
What does Compliant Database DevOps look like?
What does Compliant Database DevOps look like?

Standardize team-based development
Adopt the industry-standard tools for coding, comparison, and version control, to speed up and simplify team-based database development.
Automate database deployments

Implement a consistent, scalable, and repeatable process to automate database deployments.
Monitor performance & availability

Continuously optimize your processes by diagnosing and resolving causes of operational and performance issues, including deployments.
What does Compliant Database DevOps look like?

Protect and preserve data
Protect sensitive data as it moves through database environments.
Third parties can't be trusted.
Retail companies share customer data with market researchers, for example, and healthcare organizations share patient information with medical researchers.

Neither can insiders.
According to a 2016 study by the Ponemon Institute, upwards of 25 percent of all data breaches involve employee or contractor negligence. Whether through maliciousness or carelessness, the legitimate data access privileges of employees contribute to many data breach and leak incidents.
For Your Eyes Only: Why Data Masking Needs to Be in Every Data Security Strategy

Many business operations don’t need real data.
Plenty of organizations require data in order to build and test new programs or functions, as well as to test necessary patches and upgrades. It would be impossible to tell if a program is going to perform as it needs to if it can’t be tested with data. However, if it were tested with the actual data of users, customers, or employees, it would open up that data to the eyes of all kinds of employees or contractors who don’t require access to it.

The European Union said so.
The EU has new legislation coming into effect in May of 2018 that regulates how any organization storing or processing the data of any person in the EU can handle that data. Among many other requirements, the General Data Protection Regulation (GDPR) specifically mentions in Article 32 that data masking be used to pseudonymize sensitive data to help protect EU citizens from data breaches and other unauthorized access.
Replace sensitive data with realistic, anonymized, test data

- Multiple methods to sanitize sensitive information
- Use templates to remain consistent with existing data formats
- Replacement data sets included, such as Names, Zip Codes, Emails
- Populate test data as well as sanitize existing information
- Simple, repeatable, fast, and easy to use
- Supports SQL Server and Oracle databases
Data Masker vNext

A masking solution that fits into your organization’s process

Redgate is running an early access program to understand compliance and SQL Server data masking requirements, and ensure we’re building a solution that solves real problems.

Join the program
Benchmark your Database DevOps maturity level

See where you are, what you need, and what you can do next

Whether you're exploring the advantages of DevOps or you're already fully immersed in the journey, including the database brings additional advantages. 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.

1. Choose the area you're interested in
2. Answer the questions in each section, gaining insights along the way
3. Download your tailored report packed full of advice and recommendations

https://www.red-gate.com/solutions/maturity-assessment
Up Next

Adopting best practices to standardize team-based development
Kathi Kellenberger, Microsoft Data Platform MVP and Simple Talk Editor at Redgate