THE DATA DEMANDS
OF DEVOPS

PROJECT ACCELERATION AND AUTOMATION
REQUIRES DATA AS DRIVER, NOT DRAG

WHITE PAPER
# TABLE OF CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>What is DevOps?</td>
<td>3</td>
</tr>
<tr>
<td>The Data Gap</td>
<td>3</td>
</tr>
<tr>
<td>Data as a Service</td>
<td>5</td>
</tr>
<tr>
<td>Provision and Branch</td>
<td>5</td>
</tr>
<tr>
<td>Bookmark and Share</td>
<td>6</td>
</tr>
<tr>
<td>Refresh and Reset</td>
<td>7</td>
</tr>
<tr>
<td>Governance</td>
<td>8</td>
</tr>
<tr>
<td>Conclusions</td>
<td>9</td>
</tr>
</tbody>
</table>
Today, the demand for new applications is growing at an unprecedented rate throughout lines of business and across industries. Customer expectations for mobile and e-commerce capabilities are transforming software development speed and quality into a competitive differentiator for even the most unlikely businesses. Everywhere, more code must be shipped than ever before, faster and at higher quality.

These pressures have forced a search for new methods, practices, and solutions that can allow organizations to accelerate application development and maintain quality standards, without any additional resources. The DevOps movement has shown particular promise in helping to meet these challenges, leading 43% of F1000 respondents in an IDC survey to adopt DevOps practices, with another 40% are investigating them.

WHAT IS DevOps?

According to Gartner, DevOps is “not a market, but a tool-centric philosophy that supports a continuous delivery value chain.” DevOps supports continuous delivery and a fast flow of features from concept to customer with tools that decrease deployment friction and accelerate testing feedback and validation at every phase of the process. These objectives are achieved with solutions that accelerate environment standup and enhance environment reproducibility.

Environment standup times are the key constraint in software development timelines. While code can be easily versioned, shared, and pushed with tools like Git, environment provisioning is a complex and manual process, requiring multiple touch points from administrators, and extensive, delicate configuration work.

Environment reproducibility is the key constraint on rapid and iterative testing, just as standup times are for initial code work. Having parallel identical environments multiplies developer flexibility, allowing low-cost experimentation in both development and testing phases. But until recently, development and testing environments could not be such faithful copies.

The DevOps ecosystem of tools is transforming that landscape. Technologies such as Chef, Puppet, and Docker have automated environment standup and configuration. This automation both accelerates environment standup and enhances environment consistency.

THE DATA GAP

However, even organizations with cutting-edge DevOps practices are finding that standup and reproducibility constraints still apply to data.

A tool like Docker may be able to stand up a lightweight application instance with consistent configuration, using minimal hardware and requiring no operations time. However, development and testing environments require full and faithful copies of production data. And they need that data to be delivered at the same pace and with the same automation as VMs are configured and cloud infrastructure is made available.
Current data management technology is not up to the challenge. With existing solutions, you can have your data slowly, at poor quality, or both.

If high quality data is the highest priority, organizations can opt to create full clones of production data. But these processes take as much or more time than all other stages of environment setup. In order to get a full clone of production, a backup admin has to get data out of production, system and storage administrators need to authorize and set up infrastructure, and (if the data is relational) a DBA must set up the database. Furthermore, in a continuous deployment world, the features in production today will not be the same as those in production last month or even week. Data changes even faster. So even a perfect copy of production from weeks or months ago—and traditional data management techniques will take that long—is a poor approximation of data today.

If, on the other hand, rapid access to data is the priority, organizations can employ shared data environments. But with this strategy, conflicts occur when more than one stakeholder contends for the same resources at the same time. The result is often a low quality, chaotic environment in which data changes from different projects collide with each other, yielding unreliable code and untrustworthy tests.

Solutions like subsetting or synthetic data are often also mentioned in discussions about providing data to developers and testers. However, a subset or a synthetic data set is not an accurate copy of production. Testing on a full production copy must be relegated to a special pre-production phase of the SDLC, which undermines the DevOps emphasis on consistent environments, regular tests, and continuous adjustments to hit project targets.

**Figure 1: Legacy data delivery solutions demand harsh tradeoffs.**

If high quality data is the highest priority, organizations can opt to create full clones of production data. But these processes take as much or more time than all other stages of environment setup. In order to get a full clone of production, a backup admin has to get data out of production, system and storage administrators need to authorize and set up infrastructure, and (if the data is relational) a DBA must set up the database. Furthermore, in a continuous deployment world, the features in production today will not be the same as those in production last month or even week. Data changes even faster. So even a perfect copy of production from weeks or months ago—and traditional data management techniques will take that long—is a poor approximation of data today.

If, on the other hand, rapid access to data is the priority, organizations can employ shared data environments. But with this strategy, conflicts occur when more than one stakeholder contends for the same resources at the same time. The result is often a low quality, chaotic environment in which data changes from different projects collide with each other, yielding unreliable code and untrustworthy tests.

Solutions like subsetting or synthetic data are often also mentioned in discussions about providing data to developers and testers. However, a subset or a synthetic data set is not an accurate copy of production. Testing on a full production copy must be relegated to a special pre-production phase of the SDLC, which undermines the DevOps emphasis on consistent environments, regular tests, and continuous adjustments to hit project targets.
DATA AS A SERVICE

In order to deliver on the promise of DevOps and hit continuous release targets for even the largest, most complex and integrated applications, organizations need solutions that provide the same flexibility for data as for code bases, the same automation and repeatability for data as for configurations. They need Data as a Service. DaaS solutions offer a single, integrated platform that serves up faithful copies of source data as easily as codes or configurations, and have sophisticated features to enable collaboration, project agility, and strong governance.

PROVISION AND BRANCH

The most fundamental capability of any DaaS solution is the ability to deliver multiple copies of data promptly and with sophisticated automation. In order to deliver true DevOps capabilities, data standup should take no more time and effort than containerized code delivery or automated configuration: a few keystrokes and a few minutes.

DaaS solutions often deliver this capability through sophisticated block sharing and virtual files. Instead of moving data from system to system, DaaS solutions keep a single repository of record, and then create virtual data instances by pointing to the correct set of blocks within the repository. That allows data provisioning to occur rapidly and automatically, whether from production or non-production sources, and decouples time and effort from the size of the data set.

Figure 2: Data branching from production and non-production keeps data and code in sync.
BOOKMARK AND SHARE

DevOps isn’t just about self-sufficiency. It’s also about sophisticated collaboration. Without a DaaS solution, often data can be the bottleneck to efficient collaboration.

Consider, for example, the problems that immobile data can pose to a test-fix workflow. Such a workflow begins when a developer originates a piece of code (1). She then passes that code to a tester. He executes tests, running the code on his own data environment, and finds an error (2). He then returns the code to the initial developer with information about the bug. The developer, running the code against her own data, cannot replicate the bug (3). So she sends it to the QA person, who again finds the same bug and cannot approve the code (4). These exchanges can repeat until the different data is identified as the cause.

With a legacy solution, the developer must either take the tester’s environment, disrupting his workflow, or wait weeks for fresh data, disrupting hers. This delay can stretch development timelines, or disrupt urgent break-fix activities.

With a DaaS solution, users can save data at any point, and share a copy of that data with any other user, with the same few clicks they would use to share code. In the workflow, the developer would be able to replicate and remediate the initial error immediately after it has been identified, allowing the repaired code to pass into the push without disrupting her or her colleague’s work.

Figure 3: With legacy solutions, data is locked with the user instead of traveling with code.

Figure 4: With DaaS, free data movement accelerates test-fix cycles.
REFRESH AND RESET

Along with initial environment setup and collaborative debugging, test cycles are some of the most voracious consumers of data in the software development lifecycle. With legacy data delivery methods, testers often have to wait many hours for data to be provisioned to their test environment, in order to run a fifteen-minute test.

A DaaS solution can refresh an environment in minutes, accelerating the test cycle by a factor of ten. It can even reset the environment, restoring it to the state immediately prior to the test (but after any preparatory exercises). Long wait times for data and repeated set-up activities can be replaced with rapid, effortless reset.

Figure 5: With DaaS, test environment setup takes minutes instead of hours, and is the minority of test cycle time.
GOVERNANCE

The DevOps movement drives cross-functional collaboration to meet the needs of both developers and operations staff. An effective DaaS solution will serve both groups’ stakeholders.

To do that, it needs a distinct set of permissions and management interfaces, so that administrators and other operations staff can carefully manage existing infrastructure and resources, even as Development and testing staff spin up their own environments as-needed. A well-designed DaaS tool will not only save administrator time and effort by automating some of the dullest and most repetitive data-delivery tasks, it will also provide a full view of the team's resources for optimal management.

Figure 6: Sophisticated DaaS solutions have strong IT Operations oversight roles, so that resources are consumed sensibly.
CONCLUSIONS

The growing acceptance of the DevOps philosophy, and the maturing ecosystem of associated tools, promises to revolutionize software development across industries, replacing outdated processes and models with collaborative teams that can truly deliver business value at digital speeds. Data as a Service solutions will be a key component of this revolution, enabling the full stack of environment creation, sharing, and management, leading to an overall doubling of project delivery.
The Data Demands of DevOps
Project Acceleration and Automation Requires Data as Driver, Not Drag

You can find the most up-to-date technical documentation at:
www.delphix.com/support

The Delphix website also provides the latest product updates.
If you have comments about this documentation, submit your feedback to:
help@delphix.com

Delphix Corp.
275 Middlefield Road, Suite 210
Menlo Park, CA 94025
www.delphix.com

© 2015 Delphix Corp. All rights reserved.
The Delphix logo and design are registered trademarks of Delphix Corp.
in the United States and/or other jurisdictions.
All other marks and names mentioned herein may be trademarks of
their respective companies.