State of Test Data Management
## Contents

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>Executive Summary</td>
<td>5</td>
</tr>
<tr>
<td>The Current State of Test Data Management</td>
<td>8</td>
</tr>
<tr>
<td>Today’s Test Data Challenges</td>
<td>12</td>
</tr>
<tr>
<td>Top Considerations for a Test Data Management Approach</td>
<td>22</td>
</tr>
<tr>
<td>Building a Comprehensive Toolset</td>
<td>34</td>
</tr>
<tr>
<td>Case Study: Fannie Mae</td>
<td>40</td>
</tr>
<tr>
<td>About the Survey</td>
<td>41</td>
</tr>
<tr>
<td>About Delphix</td>
<td>43</td>
</tr>
</tbody>
</table>
Executive Summary

In today’s digital era, every company must bring high-quality applications to market at an increasingly competitive pace. While companies have adopted agile and DevOps methodologies in pursuit of this goal, many have significantly underinvested in test data—which has emerged as a constraint in the race to innovate. Once viewed as a back-office function, test data management (TDM) has emerged as a critical business enabler for enterprise agility, security, and cost efficiency.

With the proliferation of DevOps, a heightened focus on automation, and requirements to secure data across global teams of employees and contractors, IT organizations must expand the charter of traditional TDM to meet the needs of today’s development and testing teams. This report explores the top challenges that IT organizations face when managing test data, including exclusive results from a recent survey conducted by Techwell. The report also highlights the top evaluative criteria to consider when implementing new technology solutions as part of a TDM strategy.
Key Findings

The TDM market is rapidly maturing, with nearly half of all survey respondents indicating that their organization is taking steps to improve their TDM capabilities.

Adoption of DevOps and other agile methodologies is giving rise to new TDM approaches, tools, and strategies that solve data delivery challenges.

Advanced TDM teams are expanding the portfolio of datasets they can provide, enabling measurable outcomes such as 30% faster releases, 15% fewer software defects, and 100% adherence to data privacy laws and regulations.
Who's responsible for TDM?

- **50%** Centralized TDM Group
- **16%** Quality Assurance
- **10%** Individual Project Teams
- **11%** IT Operations
- **4%** There is no clear owner
- **5%** Don’t know
- **4%** Other
The Current State of Test Data Management

TDM refers to the function that creates, manages, and delivers test data to application teams. Historically, application teams have manufactured data for development and testing in a siloed, unstructured fashion. Only 5% of the respondents in the survey indicated that TDM was centralized at their organization. Instead, TDM is usually owned by QA teams (50% of the time), project teams (16%), or IT operations (10%).
As the volume of application projects has increased, many large IT organizations have recognized the opportunity to gain economies of scale by consolidating TDM functions into a single group or department—enabling them to take advantage of innovative tools to create test data and operate much more efficiently than siloed, decentralized, and unstructured TDM teams. As increasing centralization has begun to yield large efficiency gains, the scope of TDM has since expanded to include the use of subsetting and synthetic data generation, and most recently, the use of masking to manipulate production data.

The TDM market has recently shifted to a new set of strategies, largely driven by an increased focus on application uptime, faster time-to-market, and lower costs. TDM is rapidly maturing alongside other IT initiatives such as DevOps and cloud (as indicated by 28% and 24% of respondents, respectively). According to the survey results, 45% of companies are currently taking steps to improve TDM, such as by developing an enterprise-wide strategy or piloting TDM within a few groups. Another 43% of respondents indicated that they are confident or very confident that their organization will take steps to improve TDM in the next 12 months.
of surveyed firms are piloting dedicated TDM solutions or are developing an enterprise-wide strategy.

What initiatives are IT leaders investing in alongside TDM?

- DevOps: 28%
- Cloud: 24%
- Security: 12%
Application development teams need fast, reliable test data for their projects, but many are constrained by the speed, quality, security, and costs of moving data across software development lifecycle (SDLC) environments. Below are the most common challenges that organizations face when it comes to managing test data.
It takes 3.5 days and 3.8 people, on average, to fulfill a request for a new environment.

At 1 out of every 5 organizations, it takes over a week.
Environment Provisioning Is A Slow, Manual, And High-Touch Process

Most IT organizations rely on a request-fulfill model, in which developers and testers often find their requests queuing behind others. Because it takes significant time and effort to create a copy of test data, environments often take days or weeks to provision. On average, survey respondents indicated that it takes 3.5 days to provision a new environment, from request to fulfillment. Nearly 1 out of every 5 respondents reported that it takes a week or more. Often, the time to turn around a new environment is directly correlated to how many people are involved in the process. The survey results indicated that, on average, there are 3.8 people involved in the setup of a non-production environment, and at over one-third of organizations, there are more than five administrators involved. Not only does this process place a strain on operations teams, it also creates time sinks during test cycles, slowing the pace of application delivery.

How long does it take to deliver a test dataset?

<table>
<thead>
<tr>
<th>Time Frame</th>
<th>% Respondents</th>
</tr>
</thead>
<tbody>
<tr>
<td>&lt; 1 Day</td>
<td>25</td>
</tr>
<tr>
<td>1-3 Days</td>
<td>20</td>
</tr>
<tr>
<td>4-5 Days</td>
<td>15</td>
</tr>
<tr>
<td>6-7 Days</td>
<td>10</td>
</tr>
<tr>
<td>&gt; 7 Days</td>
<td>5</td>
</tr>
<tr>
<td>Total</td>
<td>100</td>
</tr>
</tbody>
</table>
Development Teams Lack High-Fidelity Data

Development teams often lack access to test data that is fit for purpose. For example, depending on the release version being tested, a developer might require a dataset as of a specific point in time. But all too often, he or she is forced to work with a stale copy of data due to the complexity of refreshing an environment. This can result in lost productivity due to time spent resolving data-related issues. Survey participants indicated that 15% of all software defects are data-related, on average.

What percentage of defects are data-related?

![Bar chart showing the percentage of defects that are data-related. The chart indicates that the majority of respondents believe that between 11% and 30% of defects are data-related.](image-url)
On average, 15% of software defects are DATA-RELATED.
Average Breach = $3.8M

but only 24% of companies are MASKING THEIR DATA
Data Masking Adds Friction To Release Cycles

For many applications, such as those processing credit card numbers, patient records, or other sensitive information, data masking is critical to ensuring regulatory compliance and protecting against data breaches. According to the Ponemon Institute, the cost of a data breach—including the costs of remediation, customer churn, and other losses—averages $3.8 million.¹ However, masking sensitive data often adds operational overhead; an end-to-end masking process may take an entire week, which can prolong test cycles.

Storage Costs Are Continually On The Rise

IT organizations create multiple, redundant copies of test data, resulting in inefficient use of storage. To meet concurrent demands within the confines of storage capacity, operations teams must coordinate test data availability across multiple teams, applications, and release versions. As a result, development teams often contend for limited, shared environments, resulting in the serialization of critical application projects.
Top Considerations for a Test Data Management Approach

To address common test data challenges, IT organizations must adopt the right tools and strategy to efficiently make the right test data available to project teams. A comprehensive approach should seek to improve TDM in each of the following areas:

- **Data delivery**: reducing the time to deliver test data to a developer or tester
- **Data quality**: meeting requirements for high-fidelity test data
- **Data security**: minimizing security risks without compromising speed
- **Infrastructure costs**: lowering the costs of storing and archiving test data

The following sections highlight the top evaluative criteria for a TDM approach.
Self-service capabilities should extend not just to data delivery, but also to control over test data versioning. For example, developers or testers should be able to bookmark and reset, archive, or share copies of test data without involving operations teams.
Data Delivery

Creating a copy of production data for development or testing is often a time-consuming, labor-intensive process that usually lags demand. Organizations must build a solution that streamlines this process and creates a path towards fast, repeatable data delivery. Specifically, application team leaders should look for solutions that feature:

- **Automation**: Modern software toolsets already include technologies to automate build processes, infrastructure delivery, and testing, among other DevOps capabilities. However, organizations often lack equivalent tools for delivering copies of test data with the same level of automation. A streamlined TDM approach eliminates manual processes—such as target database initialization, configuration steps, and validation checks—providing a low-touch approach to standing up new data environments.

- **Toolset integration**: An efficient TDM approach should unite a heterogeneous set of technologies, including masking, subsetting, and synthetic data creation. This requires both compatibility across test data tools and exposed APIs (or other clear integration mechanisms to DevOps tools) to enable a factory-like approach to TDM.

- **Self-service**: Instead of relying on IT ticketing systems, an advanced TDM approach puts sufficient levels of automation in place that enable end users to provision test data via self-service. Self-service capabilities should extend not just to data delivery, but also to control over test data versioning. For example, developers or testers should be able to bookmark and reset, archive, or share copies of test data without involving operations teams.
Taking An Optimized Approach

As companies adopt more iterative release methodologies to become more agile, data is the fastest, easiest way to accelerate test cycles. A well-orchestrated approach to TDM has the potential to transform the overall application development process by slashing wait-times for data, allowing testers to execute more test cycles earlier in the SDLC and identify defects when they are easier and less expensive to fix.

Testing in a traditional scenario vs. with an optimized TDM approach
A well-orchestrated approach to TDM has the potential to transform the overall application development process by slashing wait-times for data.
The use of subsets can result in missed test case outliers, which can paradoxically increase rather than decrease project costs due to data-related errors.
Data Quality

Operations teams go through great efforts to make the right types of test data—such as masked production data or synthetic datasets—available to software development teams. As TDM teams balance requirements for different types of test data, they must also ensure data quality is preserved across three key dimensions:

- **Data age**: Due to the time and effort required to prepare test data, operations teams are often unable to meet a number of ticket requests. As a result, data often becomes stale in non-production, which can impact the quality of testing and result in costly, late-stage errors. A TDM approach should aim to reduce the time it takes to refresh an environment, making the latest test data more accessible.

- **Data accuracy**: A TDM process can become challenging when multiple datasets are required as of a specific point-in-time for systems integration testing. For instance, testing a procure-to-pay process might require that data is federated across customer relationship management, inventory management, and financial applications. A TDM approach should allow for multiple datasets to be provisioned to the same point in time and simultaneously reset between test cycles.

- **Data size**: Due to storage constraints, developers must often work with subsets of data, which aren’t likely to satisfy all functional testing requirements. The use of subsets can result in missed test case outliers, which can paradoxically increase rather than decrease project costs due to data-related errors. In an optimized strategy, full-size test data copies can be provisioned in a fraction of the space of subsets by sharing common data blocks across copies. As a result, TDM teams can reduce the operational costs of subsetting—both in terms of data preparation and error resolution—by reducing the need to subset data as frequently.
Data Security

Masking tools have emerged as an effective and reliable method of protecting test data. By irreversibly replacing sensitive data with fictitious yet realistic values, masking can ensure regulatory compliance and completely neutralize the risk of data breach in test environments. But to make masking practical and effective, organizations should consider the following requirements:

- **Complete solution:** Many organizations fail to adequately mask test data because they lack a complete solution with out-of-the-box functionality to discover sensitive data and then audit the trail of masked data. In addition, an effective approach should mask data consistently while maintaining referential integrity across multiple, heterogeneous sources.

- **No need for development expertise:** Organizations should look for lightweight masking tools that can be set up without scripting or specialized development expertise. Tools with fast, predefined masking algorithms, for example, can dramatically reduce the complexity and resource requirements that stand in the way of consistently applying masking.

- **Integrated masking and distribution:** According to the survey, only about 1 out of 4 organizations are using masking tools because of challenges delivering data downstream. To overcome this challenge, masking processes should be tightly coupled with a data-delivery mechanism. Many organizations will also benefit from an approach that allows them to mask data in a secure zone and then easily deliver that secure data to targets in non-production environments, including those in offsite data centers or in private or public clouds.
According to the survey, only about 1 out of 4 organizations are using masking tools because of challenges with delivering data downstream.
It is not uncommon for organizations to maintain non-production environments in which 90% of the data is redundant. A TDM approach should aim to consolidate storage and slash costs by sharing common data across environments.
Infrastructure Costs

With the rapid proliferation of test data, TDM teams must build a toolset that maximizes the efficient use of infrastructure resources. Specifically, a TDM toolset should meet the following criteria:

- **Data consolidation:** It is not uncommon for organizations to maintain non-production environments in which 90% of the data is redundant. A TDM approach should aim to consolidate storage and slash costs by sharing common data across environments—including those used not only for testing, but also development, reporting, production support, and other use cases.

- **Data archiving:** According to Bloor Research, as many as 6 versions of a gold copy should be archived for testing different versions of an application.2 A TDM approach should make it feasible to maintain libraries of test data by optimizing storage use and enabling fast retrieval. Data libraries should also be automatically version-controlled in the same way that tools like Git exist for code versioning.

- **Environment utilization:** At most IT organizations, projects are serialized due to contention for environments. Paradoxically, at the same time, environments are often underutilized due to the time to populate an environment with the appropriate test data. A TDM solution should decouple data from blocks of computing resources through intelligent use of “bookmarked.” Bookmarked datasets—which can exist as of any point in time—can be loaded into environments on demand, making it easier for developers and testers to effectively time-share environments. As a result, an optimized TDM strategy can eliminate contention while achieving up to 50% higher utilization of environments.

No single technology exists that fulfills all TDM requirements. Rather, teams must build an integrated solution that provides all the data types required to meet a diverse set of testing needs.

Once test data requirements have been identified, a successful TDM approach should aim to provide the appropriate types of test data, weighing the pros and cons of each.
**Production data** provides the most complete test coverage, but it usually comes at the expense of agility and storage costs. For some applications, it can also mean exposing sensitive data.

**Subsets of production data** are significantly more agile than full copies. They can provide some savings on hardware, CPU, and licensing costs, but it can be difficult to achieve sufficient test coverage. Fewer than 1 out of 10 survey respondents indicated that 80% or more of their testing requirements were met using subsets. About 1 out of 3 organizations indicated that they are using subsets because they are easier and faster to create than full datasets.

**Masked production data** (either full sets or subsets) makes it possible for development teams to use real data without introducing unsafe levels of risk. However, masking processes can elongate environment provisioning. Also, masking requires staging environments with additional storage and staff to ensure referential integrity after data is transformed.

**Synthetic data** circumvents security issues, but the space savings are limited. While synthetic data might be required to test new features, this is only a relatively small percentage of test cases. In the survey, 12% of respondents reported that 80% or more of test requirements were met using synthetic data. If performed manually, creating test data is also prone to human error and requires an in-depth understanding of data relationships both within the database schema or file system, as well as those implicit in the data itself.
What percentage of testing requirements are met using subsets?

- < 20%: 20-40% respondents
- 20-40%: 41-60% respondents
- 41-60%: 61-80% respondents
- 61-80%: > 80% respondents
- Don’t know: N/A respondents

Why does your organization use subsets?

- Testers don’t need a full database: 0% respondents
- Limited storage available for full datasets: 5% respondents
- Faster and easier to create a test dataset: 10% respondents
- CPU and memory savings: 15% respondents
- Ability to remove sensitive data: 20% respondents
- We don’t use subsets: 25% respondents
- Other: 30% respondents

What traditional TDM tools are you using?

- Masking: 0% respondents
- Subsets/ETL: 5% respondents
- Synthetic Data Generator: 10% respondents
- Test Automation: 20% respondents
- Service Virtualization: 25% respondents
- Other: 30% respondents
Faster release cycles and time-to-market

3.5 days to refresh an environment vs. 10 minutes via self service

Higher quality releases and reduced cost

15% vs. 0% data-related defects

Ensured data privacy and regulatory compliance

VS. data secured in non-production
Building A Dynamic Data Platform

By building a dynamic data platform with Delphix, companies can transform how they manage and consume data. IT operations teams can mask and deliver data one hundred times faster while using ten times less space. Delphix also works with existing TDM tools and easily integrates with DevOps tools, enabling development teams to automatically access and control data via self service. The net result? More projects can be completed in less time using less infrastructure.
Case Study: Fannie Mae

To support their goal to go from idea to production in less than a month, Fannie Mae formed a cross-functional TDM team consisting of analysts, modelers, DBAs, developers, and testers with a deep understanding of data, as part of a larger DevOps approach. Capabilities include generating synthetic data, subsetting production data, and copying full volumes of production data for end-to-end integration testing.

By incorporating TDM capabilities into their DevOps approach, Fannie Mae enabled development teams to manage data by themselves. They reduced the wait-time for environments from weeks to minutes using virtualized databases, allowing developers to execute more test cycles and reduce overall cycle times by 30%. In addition, Fannie Mae reduced their data-related defect rate from an estimated 15-20% to less than 5%, which led to significantly less time spent on defect detection and analysis rework. From an operations perspective, in 2016 the company also saved more than 100 TB by eliminating redundant data in non-production.
About The Survey

The survey was conducted in October 2016 for two months. Over 225 respondents were surveyed across 17 industries, with significant representation from IT services (33%), financial services (17%), and healthcare (10%). Enterprises varied in size: 35% of respondents were from small and medium enterprises (under 1,000 employees) and 65% from large enterprises (over 1,000 employees).

The breakdown in job function was:

- 80% from QA/Testing
- 9% from development
- 4% from operations
- 1% from infrastructure
- Remaining percentage “Other” (project managers, consultants, etc.).
Respondents by company size

- Fewer than 1,000
- 1,001 – 5,000
- More than 25,000
- 5,001 – 10,000
- 10,001 – 25,000

Respondents by primary job function

- Quality Assurance/Testing
- Application Development
- IT Operations
- IT Infrastructure
- Other
About Delphix

Delphix’s mission is to free companies from data friction and accelerate innovation. Fortune 100 companies use the Delphix Dynamic Data Platform to connect, virtualize, secure and manage data in the cloud and in on-premise environments.

For more information, please visit delphix.com