The Data Download

As a business intelligence consulting company, we pride ourselves on being able to deliver on our projects as well as provide good quality content to our readers.

Published: January 31, 2017
Written by: Datasource Consulting

For businesses seeking optimal functionality, Test Data Management (TDM) is a crucial component of the organization’s IT structure and ongoing business strategy. Without TDM, your business is at risk for production environment application failures and subsequent breach fines, which have increased 23 percent over the past two years and can cost up to $500,000 per incident (Source: 2015 Cost of a Data Breach Study: Global Analysis).

At Datasource, we hear from multiple companies that the absence of proper test data is the chief culprit behind lackluster testing plans and problems in production. Applications require different types or volumes of test data, which makes the process even more challenging. Insufficient test data causes nearly one-third of the incidents in QA/non-production environments, and is a major reason why nearly two-thirds of business applications reach production without being properly vetted.

Overcoming Common Hurdles

To overcome common hurdles to building a strong test data management plan, consider these best practices:

1) Review test data requirements

Today, leading businesses have permanent test data management systems in place. To bolster strategy, consider resetting your TDM plan during the next major transition in your program, such as implementing a new data warehouse or QA environment. The new strategy should provide specific plans for data profiling. During this step you’ll need to understand your data, business models, and provisioning. Identify test data requirements based on test cases, keeping in mind that proper test data sets could involve a single application or multiple applications. For instance, data may need to be run through a CRM system as well as inventory and financial management applications.

2) Extract the subset

Test data should also come from multiple data subsets. It’s important to create a balance between realistic test databases, where your team can run test data efficiently, and data subsets large enough to reflect realistic production environments.
This phase is also ideal for outlining test data governance and metrics. To facilitate improved testing and training in different applications, gather input and recommendations from various groups across the organization. This can include DBAs, system architects, tool experts, data leads and more.

3) Mask business sensitive data

Always implement data masking. The risk of data breaches is high, particularly during the testing phase, and failures can lead to heavy fines, lost customer data, and customer loyalty. In addition to healthcare-related privacy requirements such as HIPAA, PCI Data Security Standard (PCI DSS) requires all companies that accept, process, store, or transmit credit card information maintain a secure environment.

Test data masking also ensures compliance with the Personally Identifiable Information (PII) Privacy Act. PII is defined as any data that could potentially identify a specific individual. In addition to customer data, remember that you need to de-identify sensitive test data that may come from corporate or employee data stores.

Knowing how to properly mask test data may require help from security experts to help you set up a test management tool. They may also offer tips on how to refresh test data as your organization goes through testing processes. This is an important step in maintaining an optimal and proficient testing environment.

4) Consider automation

The use of test data should be carefully and centrally managed to determine the cost and effort required to create testing data, maintain data masking, and ensure test data is properly organized to minimize storage and resource needs. There are several types of test data management automation options, including full TDM tools that manage, organize, and report on centralized testing efforts, as well as SQL scripts and test case automation solutions.

Consider automated options for comparing baseline test data against results from sequential testing. Some automated management tools let you define custom fields and permissions, and create specific workflows and notifications. Test reporting tools allow you to track the test coverage and they provide the ability to store manual and automated testing assets, view test cases, and link to test requirements, tasks, and defects. Consider the optimal level of automation for your deployment and business to save time, reduce costs, and gain greater resource flexibility.

5) Refresh the environment

Refreshing the environment can prove challenging, as the applications in use may experience unanticipated side effects. It’s crucial to create an environment that you can refresh correctly and continuously. Unlike data masking, this process is not a one-time occurrence. However, you can automate a refresh to your environment once your foundation is in place.

If you would like to talk about next steps for implementing a test data management plan, contact us, we’d love to chat!