

Test data management

The fuel that powers software testing

What is test data management (TDM)?

Test data management is a critical activity in every testing cycle. It typically involves creating and maintaining non-production data that closely mimics the real-life product data that the underlying applications are expected to deal with in a production environment.

According to the 2022-23 World Quality Report, 41% of organizations worldwide “still use a project-centric approach to test data provisioning.” In these organizations, there's no enterprise-wide policy to ensure test data is handled and generated in a reliable and efficient manner. Without an enterprise-wide test data provisioning policy, leadership will encounter huge challenges in ensuring the quality of the software testing teams in their organizations. **In fact, only 20% of organizations have fully implemented an enterprise-wide test data provisioning strategy.**

Why is TDM important?

Many organizations globally struggle substantially with TDM. In fact, while test case execution may be somewhat automated, test data generation itself continues to be a largely manual and tedious process. **TDM can effectively consume 50% of the time spent within a testing cycle.**

This time consumption creates a severe roadblock in quickly deploying and delivering software. Plus, it derails the cost and efficiency of teams.

The quality of test data defines how effectively the application will handle the challenges and complexity of live data.

A comprehensive and relevant test dataset can reliably test all expected user behavior and help factor in all the exception and deviant paths. This means that the software can handle non-happy paths in a well-defined way. This translates to a more positive customer and user experience.

Effective TDM also aids quality assurance teams in reducing the risks of penalties and legal regulations that the software application is subject to.

Types of test data



Static test data

Static test data is typically used to check the code to determine if it's missing any mandatory requirements or fields necessary to initiate and/or terminate a business transaction. For example, test data like Name, Date of Birth, and Address may be needed to onboard a new customer or to enroll a new patient.



Transactional test data

Transactional test data is structured data that's typically generated and captured via enterprise systems (such as CRMs, ERPs, or personal banking systems) during the lifetime of business transactions. Some examples of this type of test data include credit cards, bank accounts, healthcare claims, and patient enrollment forms. This type of test data is used to check for the functional correctness of the software.

Why is TDM a pain point for testing teams?



Lack of quality data

Testers can't use production data due to privacy, ethical, and regulatory constraints. However, most of the tester-driven logic can't mock real-life production data. This means that the test data lacks variety and will likely leak defects into production environments. Sometimes, organizations decide to skip testing entirely due to a lack of test data and decide to tackle the bugs as they come in a live production environment. **However, using this method creates issues, as the cost of fixing a bug in production can be as high as 30X compared to a non-production / user acceptance testing (UAT) environment.**



Ineffective data masking mechanisms

Organizations can't use personal identifiable information (PII) data in 'as-it-is' format in testing. They're subject to regulations such as General Data Protection Regulation (GDPR) and the Health Insurance Portability and Accountability Act (HIPAA), which prevents them from using PII data. There needs to be an effective way to manage and mask PII-sensitive data. More often than not, organizations struggle to cope with data masking at scale.



Ineffective data ingestion and extraction

Test data often has to be collected across numerous sources, such as databases and flat files. Provisioning of various synchronous and asynchronous connectors to extract test data is often a laborious task for most organizations.



Data relevance

Data relevance often presents a large challenge in achieving effective test data management. Data that was once relevant to a collection of test cases (called a test set) may become obsolete due to a regulatory or compliance change. This is a particularly significant challenge in testing business applications in business sectors such as insurance, banking, and healthcare.



Data versioning

Most often than not, test data is manually generated for each incremental iteration of the test cycle. Given this manual generation, there are high chances of inconsistent test data sources and inconsistent edge cases that creep into the software testing cycle.



Test data storage

Often the test data is maintained and stored as per the policies of the respective individual test projects. **Roughly 89% of organizations still store test data in an on-premises fashion. This means that test data could be locally picked up from sources such as Excel files, SharePoint, local DropBoxes, etc. As a result, auditing the test data quality becomes a very tedious challenge.**

How does UiPath Test Suite make test data management a hassle-free experience?













UiPath Test Suite facilitates testers and the entire business with quick, reliable ways and means of finding the right test data for their test cases. UiPath offers a host of features, such as **UiPath Task Capture** and the **UiPath Heatmap**, to generate and identify relevant test data based on your actual software usage data.

UiPath Test Suite is designed to automate test data provisioning and integration into Continuous Integration/Continuous Deployment (CI/CD) pipelines. The idea is to enable testing teams with the ability to cover all types of scenarios (positive / negative / edge cases) for a test case. This is facilitated by Test Suite's ability to generate synthetic test data. **Synthetic test data is generated proactively and on-demand, ensuring repeatable and reliable test execution.**



Challenge

Solution with UiPath Test Suite

- | | |
|--|---|
|  Lack of quality data |  Synthetically generate new data (such as names, numbers, strings, dates, addresses, etc.)
Embed rules into your test case logic (for example: no hard-coded value in test scripts) |
|  Data masking |  Mask PII information quickly and easily with UiPath encryption activities |
|  Data ingestion |  Database and WebAPI activities to connect to a range of data management systems via OBC/APIs
Data services to bring test data into a single, secure, and scalable environment via a simple drag-and-drop
UiPath Apps to enable business teams to engage with automations via interactive and self-explanatory forms. This is especially useful for huge transformation projects in SAP, Oracle, and Salesforce, where engaging with business users is critical to the success of testing projects |
|  Data aging |  Automatically generate new test data to execute test cases using a various sequence of data |
|  Data versioning |  Version control each iteration/execution of test cycles and test data
Data Manager panel allows you to manage various types of data such as variable, assets, and queues |
|  Test data storage |  Import and use your existing data sets in Test Manager/Orchestrator
Import test data via a variety of formats such as JSON, CSV, and Excel
Test data queues in Test Suite act as a container to hold queue items to be consumed in test scripts |

To learn more about test data management, download the e-book:

<https://www.uipath.com/resources/automation-whitepapers/test-data-management-with-test-suite>