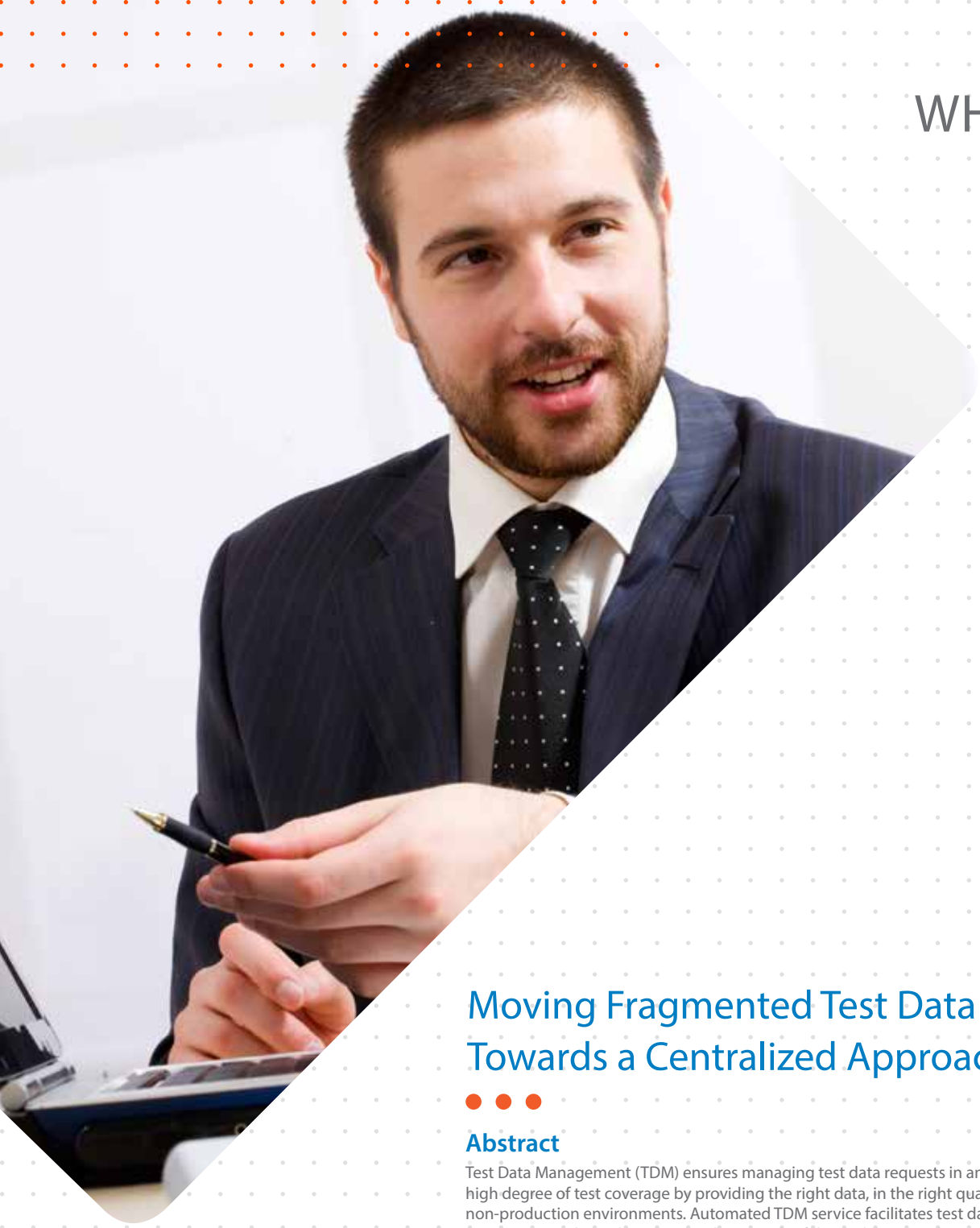
WHITE PAPER



Moving Fragmented Test Data Management Towards a Centralized Approach

Abstract

Test Data Management (TDM) ensures managing test data requests in an automated way to ensure a high degree of test coverage by providing the right data, in the right quantity, and at the right time, in non-production environments. Automated TDM service facilitates test data management across test environments through a structured approach of data subsetting, cleansing, gold copy creation, data refresh, and sensitive data masking.

Typically, a centralized TDM system with well-defined processes is more effectual than the traditional manual or decentralized approach, but in some cases, a decentralized approach is adopted. This paper takes a deeper dive into the considerations for the centralization of TDM processes within enterprise ITs.

Infosys
be more

Introduction

In most organizations where TDM is at its infancy, test data-related activities are done by the individual project teams themselves. There will not be a dedicated team identified or process defined to handle test data requests. Such projects with a primitive TDM approach possess several drawbacks:

- Lack of a defined ownership for the test environment and test data setup: results in unintentionally losing the test data setup or data overstepping
- Unavailability of data setup for testing end-to-end scenarios: Lack of data setup between inter-dependent and third-party applications
- Lack of referential integrity defined in the databases: Absence of primary, foreign, relationships defined in the database makes it difficult to identify related tables and generate the correct test data set
- Insufficient data available for performance load testing: Manually generating bulk data is a tedious task and less feasible
- Increased number of defects due to incorrect test data: Leads to re-work and losing time unnecessarily analyzing issues caused due to incorrect test data used for testing
- Outdated test data in QA database: Periodic refresh of test data does not happen from production
- Inability to provision data since data is unavailable: Lack the mechanism required for generating synthetic data
- Risk of exposing sensitive data to testing teams: Sensitive fields need to be masked before provisioning for testing
- Multiple copies of data: Storage costs can be reduced by maintaining required gold copies and refreshing and reusing gold copies after major releases

Having a well-defined practice for handling all the test, data-related, requirements across all non-production environments in an organization is the essence of TDM. Aimed to address all the above stated issues, it will bring in more control and make TDM more effective.

Based on the TDM requirement type, organizations can opt for either a de-centralized or a centralized approach. This paper gives a detailed view of both approaches and highlights how the centralized approach is more efficient and beneficial.

Centralized TDM

Centralized TDM deals with consolidating the test data provisioning for all non-production environments across the organization. It provides a systematic approach to analyze and provision test data.

Pros

- Well-established TDM team with a workflow-based mechanism for managing test data requests
- Reduced latency in provisioning test data with quick turnaround time
- Automated end-to-end approach with tools and processes
- Reduced infrastructure cost by storing only the required data for provisioning in gold copy database
- Reduced risk of incorrect test data, resulting in lesser defects
- Resolution of data overstepping issues by the TDM team
- Periodic refresh of the gold copy makes the latest data available for testing by the QA team
- Reusable masking configurations and test data generation scripts provides quick turnaround time
- Easy handling of complex end-to-end test scenarios that require data setup across heterogeneous data sources having federated relationships through a centralized test data managment
- Creation of bulk data which is relationally intact for non-functional testing requirements is achieved using automated solutions
- Varied techniques available for creating synthetic data in scenarios where source data is not available for provisionin

Cons

- Considerable time and effort is required to consolidate the TDM across various portfolios
- High knowledge acquisition effort required to understand the different application data models
- Sporadic bottlenecks and dependency on the TDM team in case of high workload from all LOBs

Decentralized TDM

It is not necessary that all applications in different portfolios in the organization's landscape have to be covered under the consolidated TDM umbrella. There are instances where some applications can follow the de-centralized TDM approach. This is mostly determined by the level of integration between the applications, technologies supported, data sensitivity, environment constraints, etc. For example, data in HR, infrastructure applications, etc., may be independent and not related to marketing, sales, inventory, or corporate data. These systems, hence, can adopt a de-centralized TDM approach and need to be handled outside the centralized umbrella.

Pros

- Minimal effort required to set up TDM for individual applications
- Good understanding of the respective application data models, which makes the team capable to address the test data requests quickly

Cons

- Multiple copies of data without ownership because individual teams store separate copies of production data.
- Unmasked sensitive data in non-production environments can lead to a security breach
- Less uniformity in standards and processes
- Increase in data overstepping issues
- Minimal automation may be present with lack of coordinated processes
- Limitations in setting up data across multiple data sources due to decentralized systems. Data set up in one application may not be in sync with other inter-dependent applications

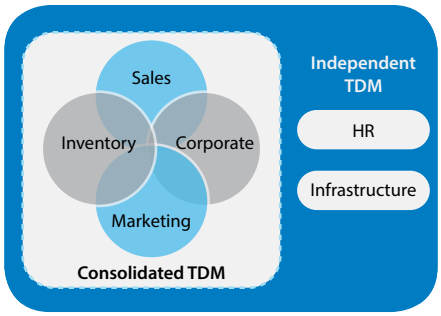


Figure 1: TDM Landscape

Centralized TDM implementation approaches

Primarily, there are two approaches for implementing centralized test data management within an organization:

- Big Bang approach: In this approach, all major applications under the TDM scope in the organization are identified, test data requirements across applications are analyzed, and gold copies for these applications are created at one go. A TDM team is set up to address test data needs for all the applications.

  This approach will take considerable time for the initial setup and knowledge of the application stack across the organization's portfolio is a must.

  Another key challenge with this approach is keeping up with the database (DB) changes happening in production during the initial setup
- Incremental approach: In this approach, based on the business requirements, TDM is established for an application or a prioritized set of applications. Separate test data management implementations will be carried out which can be progressively integrated. The TDM team will address the test data needs as soon as the gold copies for the applications are set up.

  In this approach, TDM is more manageable, and can reap early benefits. TDM set up for smaller set of applications takes lesser time compared to the Big Bang approach.

A phased approach for TDM implementation

Centralized and automated test data management implementations can follow a phased approach. Each stage has a defined set of activities to achieve the goals and these stages are:

- Analysis
- Design
- Implementation
- Steady State

From the initial assessment phase, it moves to a stabilized stage, expanding TDM services to other domains and portfolios in the organization, and working for continuous improvements on the way.

The timelines proposed in the diagram are highly indicative. Time duration for each phase will depend on factors like:

- TDM requirement complexity
- Number of portfolios or applications involved
- Knowledge or understanding about the application landscape or database models

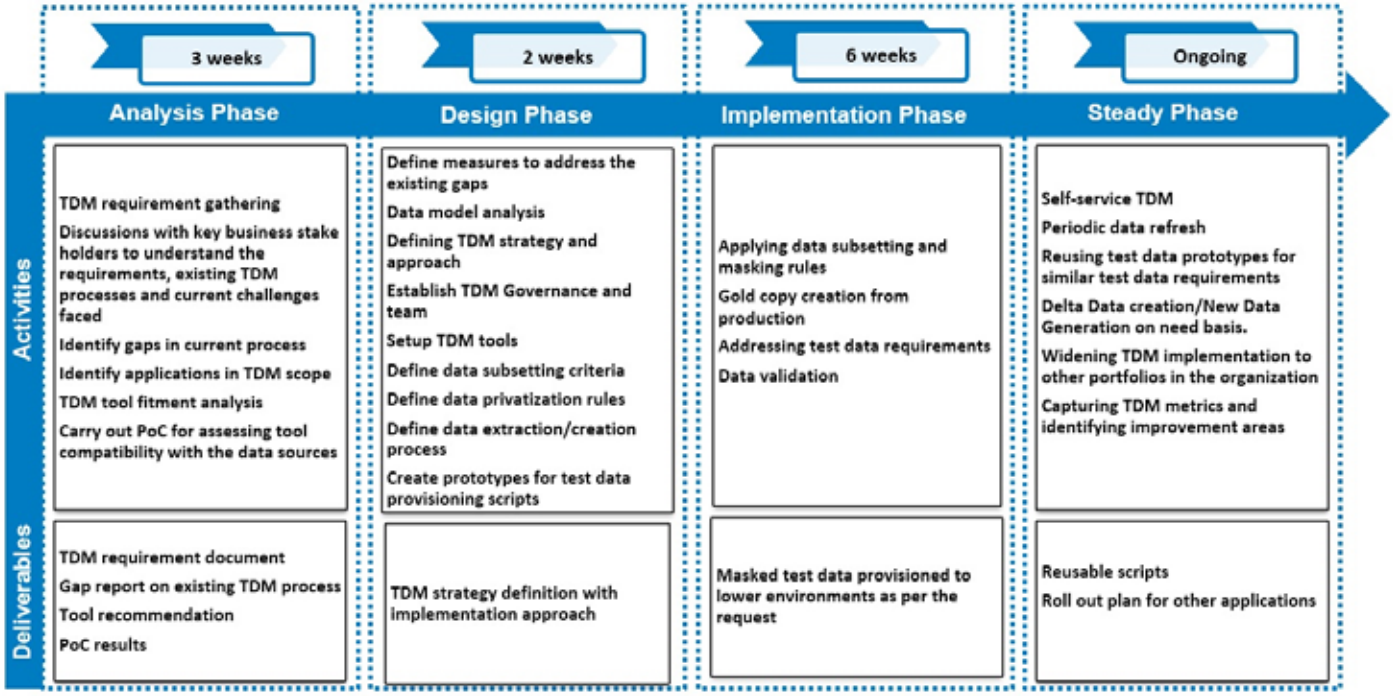


Figure 2: Phased TDM Approach

When to go for centralized TDM?

- Applications or technologies used are mostly compatible with tools in the market
- Scope of TDM for applications across various portfolios continue throughout the life cycle of the application
- Incoming TDM requests for application or application clusters are fairly high
- Technologies are widely supported and not disparate
- High number of inter-dependent systems which require data set up across systems for end-to-end testing

When to go for decentralized TDM?

- The nature of portfolios or departments within the organization are highly de-centralized
- A specific TDM process is required for a prioritized set of applications within a short span of time
- Scope of TDM is limited within the project and does not continue after the project is complete
- Disparate or obsolete technologies used in the project are not supported by common TDM tools
- Limited number of dependent / external applications
- Need for test data provisioning is very low and the requests flow is manageable

Common TDM challenges and resolutions

1. **Inconsistent data relationship**

   Well-defined data relationship between database objects is a key factor for data subsetting, masking, and data provisioning. It is often observed that in case of legacy applications, relationships are not present in the database layer. The business rules and logical constraints may be applied at the application level, but will be poorly defined at the database level. Logical database model architectures may not be available in most cases.

Impact

- Data subsetting, data masking, and data provisioning get affected
- Data integrity will not be maintained

Resolution

- Understand the application and database structure, relevance of tables, and how they are related with help of SME / DBA
- Analyze and understand the database structure using data model artifacts
- Validate the logically-related entities and confirm with business analyst

2. **Unclear test data requirements**

   Teams requesting data sometimes lack information aboutwhich data sources would have the related data that needs to be set up. In some scenarios, test data requirements can be very complex, like for testing an end-to-end scenario with data spread across multiple databases or with data spread across tables.

Impact

- Inaccurate test data

Resolution

- Understand the requirement from QA perspective
- Understand the database entities involved and the relationships

3. **Lack of application knowledge**

   System or application knowledge, especially the data sources under the TDM scope, is a prerequisite for the TDM team. If teams possess a limited knowledge about the application,

it will result in writing incorrect test cases, raising ambiguous test data requirements, and finally, provisioning inaccurate data.

Impact

- Inaccurate test data
- Increased defects due to incorrect test data

Resolution

- Understand the application with the help of SMEs
- Understand the database entities involved and the relationships

4. **Corrupted gold copy**

   Most projects will have a gold copy database available from where data will be provisioned to the lower environments. If the gold copy is not refreshed periodically, or the data in the gold copy has been tampered with, it can cause issues while provisioning data.

Impact

- Inaccurate test data

Resolution

- Periodically refresh gold copy database
- Restrict access to gold copy database

5. **Data overstepping**

   If the same set of test data is used by multiple teams for testing, it can lead to conflicts and the test results will not be as expected.

Impact

- Affects test execution
- Incorrect test results
- Rework in test data provisioning and test execution

Resolution

- Data has to be reserved
- Centralized TDM team can handle the test data requirements

6. **Identifying correct sensitive fields and masking techniques**

   While masking any application database, it is important that the correct sensitive fields are identified for masking. Also, what is important is that relevant masking techniques are applied to these fields. For example, email id should be masked in such a way that the email id format is retained. Otherwise, while using the masked email id, it might break the application. Another point to consider is while masking the primary key columns, the masking has to be consistently applied for the child tables also where the primary key columns are referenced.

Impact

- Data inconsistency across tables
- Unnecessary masked data

Resolution

- Identify sensitive fields belonging to the category PII, PHI, PCI, financial data, etc.
- Apply relevant masking techniques that will preserve the format of the data

Best practices

Some of the best practices that can be adopted while implementing test data management processes in projects are listed below:

- Automate TDM processes with reusable templates and checklists
- Improve test data coverage and test data reuse by provisioning and preserving the right data
- Analyze TDM metrics and take corrective actions
- Data refresh to gold copy can be automated using scripts
- Batch mode of data masking can be implemented to improve performance without exposing sensitive data to testing teams
- Test data can be used for configuring the masking rules, which can be replaced with production data for actual execution. Thus, production data is not exposed to the execution team
- Reusable configuration scripts for masking similar data (for example – similar data for a different region)
- Developing automation scripts to automate any manual TDM-related activities
- Developing data relationship architecture diagrams for the most commonly used tables for provisioning which can be used as a reference

Summary

Reduced cost and improved management with faster time-to-market are the key points for any successful program. Centralized and automated test data management provides an organized approach in managing test data requirements across the organization in a better and more efficient way. Only the required masked, subsetted, and reusable data sets, are stored as gold copies centrally, which are used for provisioning by the testing teams. Most of the TDM tools available in the market offer web-based solutions, which act as a single interface for both the testing and provisioning teams. Testing teams can place the test data request and provisioning teams can address the request from a single portal. All test data requests are tracked using a single solution. A centralized, automated TDM system with streamlined processes introduce increased accuracy and predictability to the entire testing process. Implementing centralized test data management is certainly beneficial over the de-centralized approach.

Glossary

Acronym	Definition
TDM	Test Data Management
PII	Personally Identifiable Information
PHI	Personal Health Information
PCI	Payment Card Information
SME	Subject Matter Expert
PoC	Proof of Concept

Case study – centralized TDM for a leading pharmacy client

Overview

The client has a complex IT landscape with data spread across multiple portfolios including marketing, sales, corporate, pharmacy, and supply chain. Some of the applications across the portfolios have a federated relationship with related data. The TDM service engagement requirement was to establish a well-defined TDM process and governance, which will address all the test data-related requests for the projects under different portfolios, gradually expand the TDM services to newer portfolios, and finally consolidate under the same umbrella.

Problem statement

- Identify the test data and data masking requirements in different portfolios and application databases
- Perform gap analysis for the existing TDM processes
- Establish a defined test data management process and governance
- Implement an automated TDM process using the right tools
- Test data provisioning for functional, automation, and performance testing teams
- Metrics-based approach for the evaluation of test data management implementation

Challenges

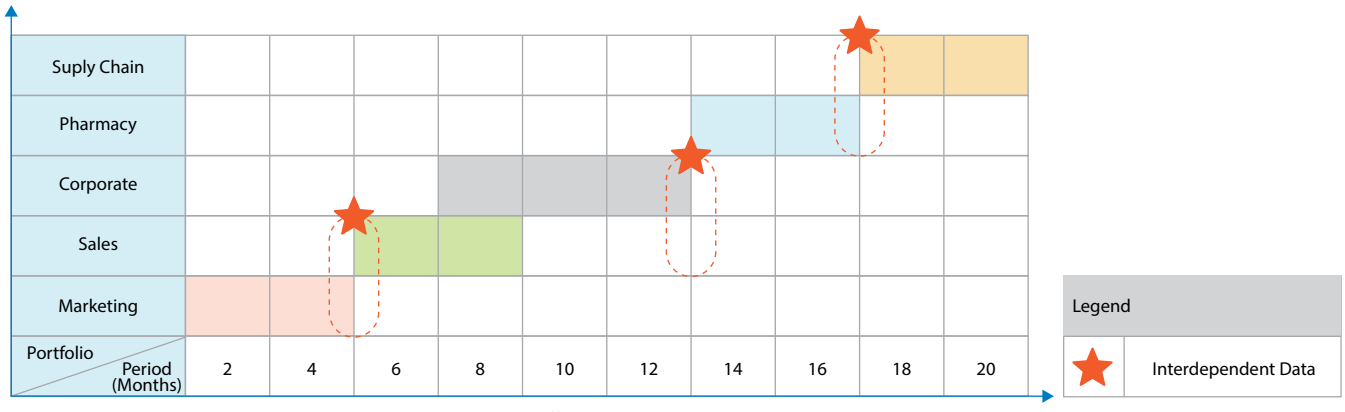
- Complex IT landscape with heterogeneous data source types
- Lack of defined test data management processes / strategy
- Manual TDM activities for data subsetting and masking
- Lack of integrated data across systems
- Sensitive data being moved to a non-production environment without masking
- Huge cycle time for generating test data, impacting test execution schedules

Solution approach

- Established a centralized TDM team to provision test data for functional and non-functional testing
- Deployed a web-based, self-service tool for the testing teams to place the data request and provisioning
- Masked data is provisioned to testing teams ensuring compliance to PIPEDA (Personal Information Protection and Electronic Documents Act)
- Established automated TDM processes and capabilities across portfolios
- End-to-end testing made easy by synching up test data across interdependent applications

Benefits / value-adds

- 20% reduction in test data provisioning cycle time
- Production data not exposed to testing teams
- Repository of reusable masking and test data generation scripts
- Automated TDM services reduced test data related defects to zero resulting in quality deliverables



TDM Service Rollout Map

Seema Varghese is a Technical Test Lead with Infosys, having 11 years of IT experience, including leading teams and developing testing expertise in different domains (retail, pharmacy, and telecom). She has worked in data migration and data warehouse testing projects. She also has experience handling TDM and data masking projects.

For more information, contact askus@infosys.com

Infosys
be more