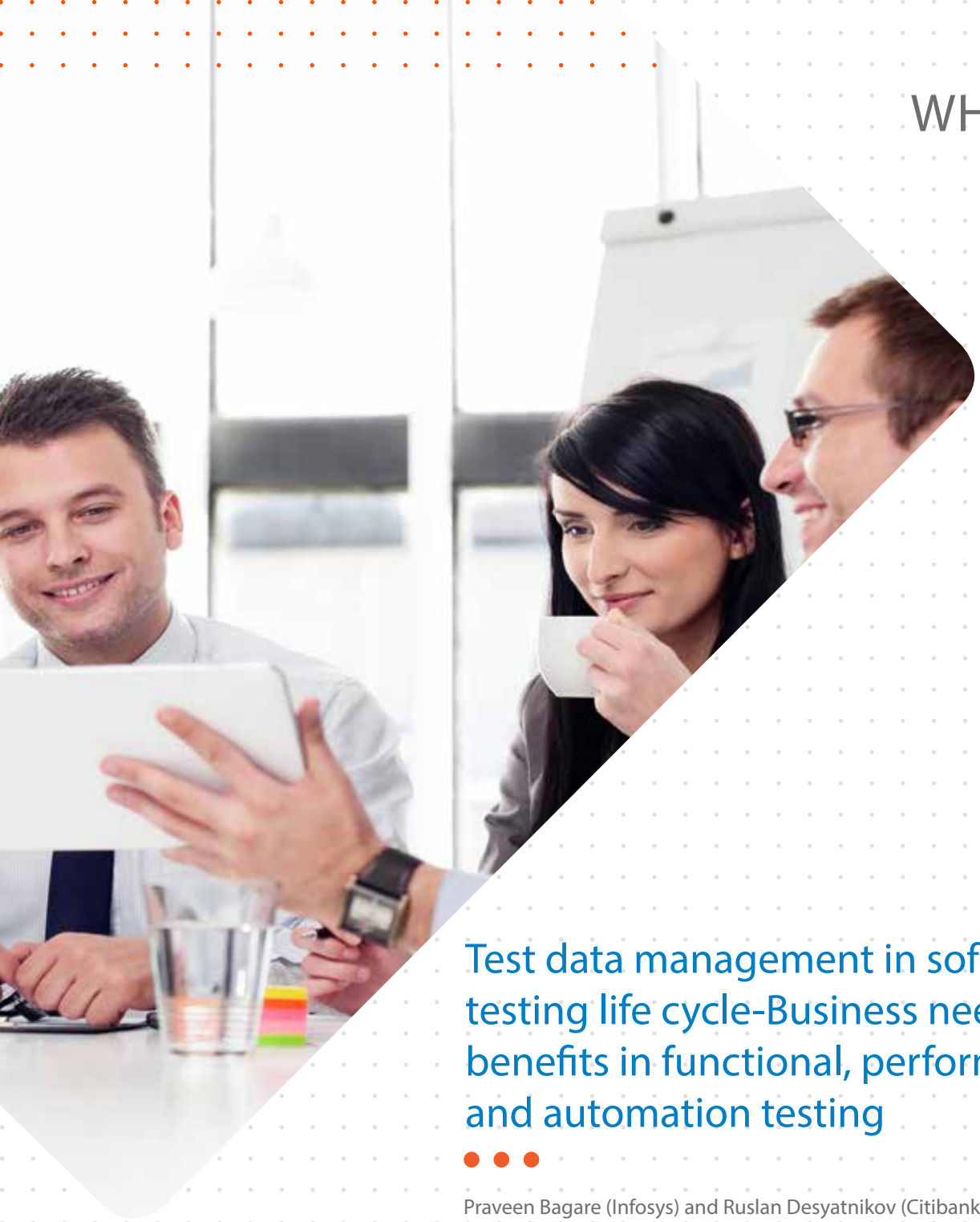
WHITE PAPER



Test data management in software testing life cycle-Business need and benefits in functional, performance, and automation testing

Praveen Bagare (Infosys) and Ruslan Desyatnikov (Citibank)

Infosys
be more

Abstract

The testing industry today is looking for ways and means to optimize testing effort and costs. One potential area of optimization is test data management. Testing completeness and coverage depends mainly on the quality of test data. It stands to reason that without high quality data testing assurance is unattainable. A test plan with several comprehensive scenarios cannot be executed unless appropriate data is available to run the scenarios.

The best data is found in production since these are actual entries the application uses. While using production data, it is always prudent to create a sub-set of the data. This reduces the effort involved in test planning and execution and helps achieve optimization. However, live data is not always easily available for testing. Depending on the business, privacy and legal concerns may be associated with using live data. Often the data is not complete and therefore cannot be used for testing. It is best to avoid the use of raw production data to safeguard business and steer clear of expensive lawsuits.

The challenge of TDM lies in obtaining the right data effectively. Before proceeding on this path, we need to find answers to some pertinent questions: Will there be a positive return on the investment? Where do we start implementing Test Data Management (TDM)? Should we start with functional testing or non-functional testing? Can test automation help?

The practice of not including TDM steps in the Testing Life Cycle often leads to ignorance towards TDM on part of the testing team. This paper attempts to explain why testers in the functional, non-functional and automation test arenas need the TDM service. We also discuss the test data challenges faced by testers and describe the unparalleled benefits of a successful TDM implementation.

Significance of Test Data Management

TDM is fast gaining importance in the testing industry. Behind this increasing interest in TDM are major financial losses caused by production defects, which could have been detected by testing with the proper test data. Some years ago, test data was limited to a few rows of data in the database or a few sample input files. Since then, the testing landscape has come a long way. Now financial and banking institutions rely on powerful test data sets and unique combinations that have high coverage and drive the testing, including negative testing. TDM introduces the structured engineering approach to test data requirements of all possible business scenarios.

Large financial and banking institutions also leverage TDM for regulatory compliance. This is a critical area for these institutions due to the hefty penalties associated with non-compliance. Penalties for regulatory non-compliance can run into hundreds of thousands of dollars or more. Data masking (obfuscating) of sensitive information and synthetic data creation are some of the key TDM services that can assure compliance.

What is Test Data Management?

Test data is any information that is used as an input to perform a test. It can be static or transactional. Static data containing names, countries, currencies, etc., are not sensitive, whereas data pertaining to Social Security Number (SSN), credit card information or medical history may be sensitive in nature. In addition to the static data, testing teams need the right combination of transaction data sets/conditions to test business features and scenarios.

TDM is the process of fulfilling the test data needs of testing teams by ensuring that test data of the right quality is provisioned in suitable quantity, correct format and proper environment, at the appropriate time. It ensures that the provisioned data includes all the major flavors of data, is referentially intact and is of the right size. The provisioned data must not be too large in quantity like production data or too small to fulfill all the testing needs. This data can be provisioned by either synthetic data creation or production extraction and masking or by sourcing from lookup tables.

TDM can be implemented efficiently with the aid of well-defined processes, manual methods and proprietary utilities. It can also be put into practice using well-evolved TDM tools such as Datamaker, Optim or others available in the market.

A TDM strategy can be built based on the type of data requirements in the project. This strategy can be in the form of:

- Construction of SQL queries that extract data from multiple tables in the databases
- Creation of flat files based on:
  - Mapping rules
  - Simple modification or desensitizing of production data or files
  - An intelligent combination of all of these

The Challenges of Test Data Sourcing

Some of the most common challenges faced by testing teams while sourcing test data are:

- Test data coverage is often incomplete and the team may not have the required knowledge.
- Clear data requirements with volume specifications are often not gathered and documented during the test requirements phase.
- Testing teams may not have access to the data sources (upstream and downstream).
- Data is generally requested from the development team which is slow to respond due to other priority tasks.
- Data is usually available in large chunks from production dumps and can be sensitive in nature, have limited coverage or may be unsuitable for the business scenarios to be tested.
- Large volumes of data may be needed in a short span of time and appropriate tools may not be at the testing team's disposal.
- Same data may be used by multiple testing teams, in the same environment, resulting in data corruption.
- Review and reuse of data is rarely realized and leveraged.
- Testers may not have the knowledge of alternate data creation solutions using a TDM tool.
- Logical data relationships may be hidden at the code level and hence testers may not extract or mask all the referential data.
- Data dependencies or combinations to test certain business scenarios can add to the difficulties in sourcing test data.
- Testers often spend a significant amount of time communicating with architects, database administrators, and business analysts to gather test data instead of focusing on the actual testing and validation work.
- A large amount of time is spent in gathering test data.
- Most of the data creation happens during the course of execution based on learning.
- If the data related to defects is not found during testing, it can cause a major risk to production.

TDM Offers Efficient Solutions and Valuable Benefits

An effective TDM implementation can address most of the challenges mentioned above. Some of the key benefits that a business can gain by leveraging the TDM services are:

Superior quality	Minimum time	Reduced cost	Less resources
• Optimal data coverage is achieved by the TDM team through intelligent tools and techniques based on data analysis strategies	• The TDM service employs a dedicated data provisioning team with agreed service-level agreements (SLAs) ensuring prompt data delivery	• Condensed test design and data preparation effort helps achieve cost savings	• Database or file access provided to the TDM team facilitates data privacy and reuse
• Test data requirements from the TDM team enable the testing team to capture these effectively during the test planning phase. Version-controlled data requirements and test data ensure complete traceability and easier replication of results	• Compact test design and execution cycles can be achieved for reduced time to market	• Minimized test data storage space leads to reduction of overall infrastructure cost	• Professionals with specialized skills, sharp focus on Test Data and access to industry standard tools contribute to the success of TDM
• Detailed analysis and review of data requirements ensure early identification of issues and resolution of queries	• Automated processes lead to less rework and reduced result replication time		• The TDM team also wears the system architect's hat, thus understanding data flow across systems and provisioning the right data
• Synthetic data can be created from the ground up for new applications	• TDM tools such as Datamaker can speed up scenario identification and creation of the corresponding data sets		
• Errors and data corruption can be reduced by including defined TDM processes in the Testing Life Cycle and by adopting TDM tools			
• Clear data security policies increase data safety and recoverability			
• Well-defined process and controls for data storage, archival and retrieval support future testing requirements			

TDM in Functional Testing

Most challenges mentioned earlier are part of the day-to-day struggle of functional testing. The most commonly seen challenges are: low coverage, high dependency, access limitations, oversized testing environment, and extended test data sourcing timelines. Successful implementation of TDM in functional testing projects can alleviate most of these issues and assure the completeness of testing from the business perspective.

In functional testing, TDM is governed by numerous factors highlighted below.

Coverage: Exposure to all the possible scenarios or test cases needs to be the key driving factor for data provisioning in functional testing. The data provisioned for this testing must cover:

- Positive scenarios (valid values that should make the test case pass)
- Negative scenarios (invalid values that should result in appropriate error handling)
- Boundary conditions (data values at the extremities of the possible values)
- All functional flows defined in the requirement (data for each flow)

Low volumes: Single data sets for each of the scenarios are sufficient for the need. Repetitive test data for similar test cases may not be required and can prove to be a waste of time. This can help reduce the execution time significantly.

High reuse: Some test data such as accounts, client IDs, country codes, etc., can be reused across test cases to keep the test data pool optimized. Static and basic transaction data for an application can be base-lined so as to be restored or retrieved for maintenance release testing at regular intervals, depending on the release frequency.

Tools or utilities: As utilities have the capacity to create large volumes of the same type of data, they may not be very useful in functional data preparation. If the tool is capable of creating a spectrum of data to meet all the data requirements and the data can be reused across releases, then it is beneficial to use the utility or tool for TDM implementation.

Optimal Environment Size

Data in the testing environment should be a smart subset (a partial extract of the source data based on filtering criteria) of production or synthetic data. It should comprise data for all testing needs and be of minimal size to avoid hardware and software cost overhead. Sub-setting referentially intact data in the right volume into testing environments is the solution to overcome the problem of oversized, inefficient and expensive testing environments. This is the key to reduce the test execution window and minimize data-related defects with the same root cause.

Data Requirements Gathering Process

During test case scripting, the test data requirements at the test-case level should be documented and marked as reusable or non-reusable. This ensures that the test data required for testing is clear and well documented. A simple summation of the same type of test data provides the requirement of test data that needs to be provisioned.

A sample test data requirement gathering, as per this method, has been demonstrated in Appendix I.

Feasibility Check

TDM is suitable for functional testing projects that:

- Spend over 15% of the testing effort on data preparation or data rework
- Use regression test cases which are run repeatedly across releases (so that the test data methodology identified can be reused)
- Indicate that a high data coverage TDM solution can be identified and implemented

TDM in Performance Testing

Test data preparation in performance testing is most impacted by issues like large volume requirements with significant coverage, high data preparation time, limited environment availability and short execution windows. The TDM team with their tools and techniques can provide solutions for bulk data generation with quick refresh cycles ensuring on-time high quality data provisioning to address these steep demands.

Test data for performance testing is characterized by:.

High volumes of data: Performance testing always needs large volumes of test data by virtue of the way it works. Multiple users are simultaneously loaded onto an application to run a flow of test executions. Hence multiple data sets are required in parallel, in large numbers, based on the workload model.

Quick consumption of test data: Since the load or stress on the application is induced by multiple users, the data provisioned for them is consumed rapidly. This leads to quick exhaustion of large volumes of the test data.

Very short data provisioning cycle: Since the data is consumed very quickly in performance testing, a new cycle of execution requires the data to be replenished before start. This implies that the TDM strategy must make sure that the test data can be recreated, extracted again or provisioned in a short period of time so as to avoid the impact of the unavailability of data on the testing cycle.

Workload distribution: Performance testing in today's time does not use one type of data repeatedly. If the use cases comprise multiple types of data, with a corresponding percentage of each occurrence, a similar workload model is built for the testing environment. Thus smart data needs to be provided for such a workload model which covers the multiple types. This multiplies the complexity of data creation.

Feasibility Check

TDM is definitely suitable for most of the performance testing projects. The strategy needs to be well designed to make sure that the challenges listed above are addressed appropriately and a satisfactory return on investment (ROI) is achieved soon.

TDM in Automation Testing

Some of the biggest challenges in automation testing are associated with creating test data. Included in this challenge is the inability to create data quickly from the front-end, fast burning up of large volumes of data during test runs, limited access to dynamic data and partial availability of the environment.

Implementing a well-designed TDM strategy can support multiple iterations of dynamic data in short intervals of time by synthetically creating or extracting data, using TDM tools.

Test data for automation testing is driven by factors such as:

Automation of test data creation: The data required for automation testing is usually created by using one of the automated processes, either from user interface (UI) front-end or via create or edit data operations in the database. These methods are time-consuming and may require the automation team to acquire application as well as domain knowledge.

Fast consumption of test data: Similar to performance testing, automation testing also consumes data at a very quick pace. Hence the data provisioning strategy must accommodate fast data creation with a relatively short life cycle.

High coverage: As with functional testing automation testing also requires test data for each of the automated scenarios. The data requirement may be restricted to the regression test pack and yet covers a large spectrum of data.

Feasibility Check

TDM as process implementation is certainly recommended for an automation project. The automation team must provide the data requirement in clear terms and the TDM team can ensure provisioning of this data before the test run begins. Automation tools have abilities to create, mock and edit data but the TDM team's expertise and tools can add significant value to the proposition.

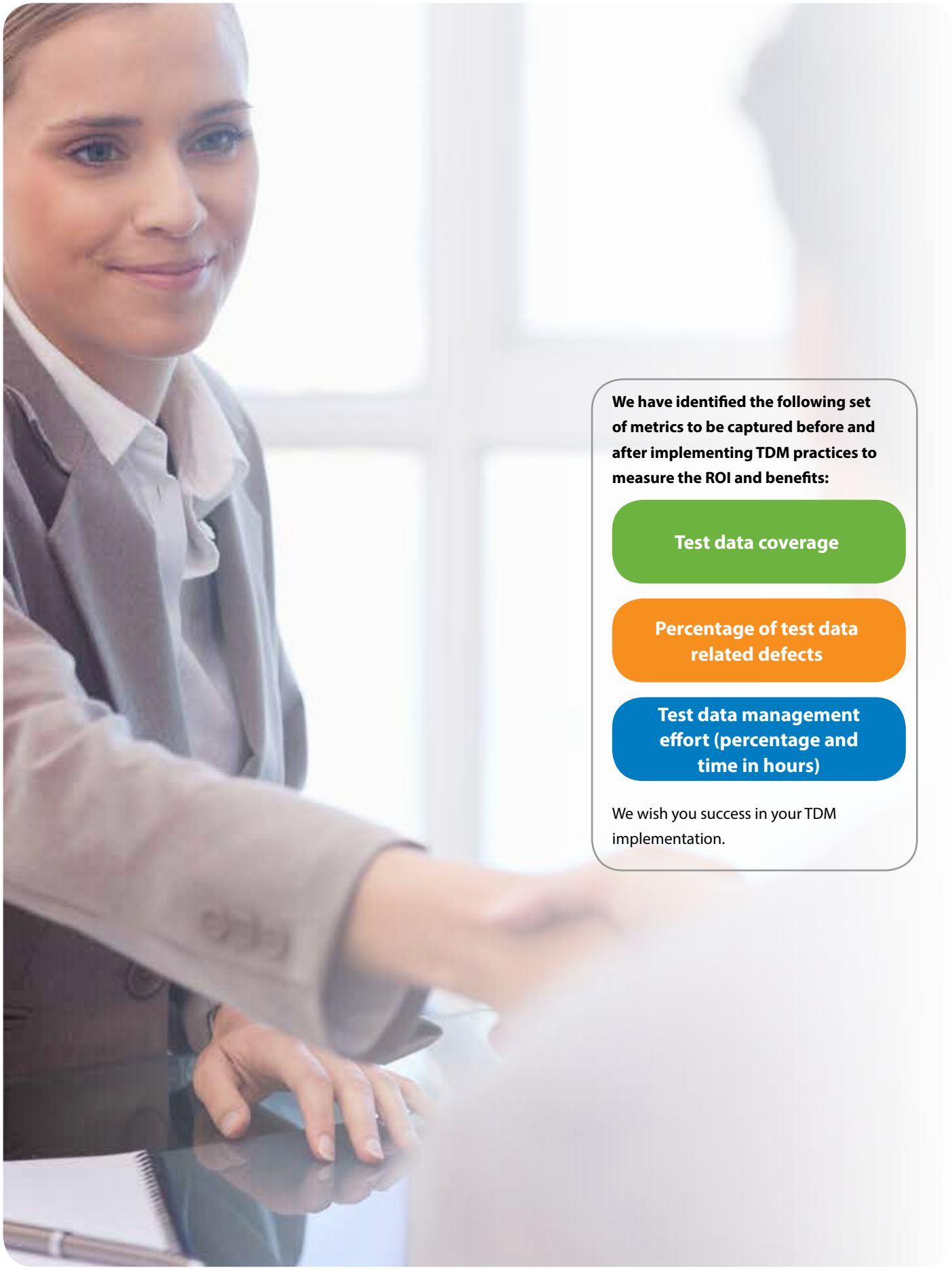
Conclusion

In functional testing, increasing data coverage plays a significant role in providing the TDM value-add. The sheer volume of test data that is repeatedly used in the regression suites make it an important focal area from the ROI viewpoint. The right TDM tools can help provision a spectrum of data and ensure continuous ROI in each cycle.

TDM implementation in performance testing projects can deliver quick benefits and the improvement can be significantly highlighted as large volumes of similar data can be created swiftly and efficiently.

Automation testing can benefit from TDM implementation. Tools such as Quick Test Professional (QTP) can create data via user interface but need significant functional knowledge and are slow in nature. TDM solutions can save time and cost by keeping the data ready. Robust data creation methods and tools can be used to achieve these goals.

Based on our learning, interactions and experience gleaned from the TDM world, we can confidently affirm that functional, automation and performance testing can leverage TDM implementation to overcome their respective challenges and achieve optimization. Each type of testing presents its own unique challenges and benefits, but there is a common theme – TDM is a major enhancement and addition to the tools and techniques available to the testing team. This practice can help realize gains at the bottom-line with cost reductions, improved turnaround time and fewer data related defects in test and in production.



We have identified the following set of metrics to be captured before and after implementing TDM practices to measure the ROI and benefits:

Test data coverage

Percentage of test data related defects

Test data management effort (percentage and time in hours)

We wish you success in your TDM implementation.

Appendix I

An illustration of 'Sample Test Data Requirement Gathering'

The last three columns depicted in the table below should be a part of the test case documentation and must be updated during test case authoring.

No.	Test Case	Test Data Requirement	Reusable?	Remarks
1	Test Case 01	NAM Bank Account Number	Y	with balance > $1,000
2	Test Case 02	Client ID	N	-
3	Test Case 03	NAM Bank Account Number	Y	with balance > $100,000
4	Test Case 04	ASIA Bank Account Number	Y	Account open date > 01 Jan 2013
5	Test Case 05	ASIA Bank Account Number	Y	-

Data requirement summary for the above illustration:

- Two 'NAM Bank Account Number''with balance > $100,000'
- Two 'ASIA Bank Account Number'
- One client ID

Reference

http://www.gxsblogs.com/wp-content/blogs.dir/1/files/SEPA-Penalties-Table1.png?8f0a21

For more information, contact askus@infosys.com