

## Extending Function Point Estimation for Testing MDM Applications

### Executive Summary

Effort estimation of testing has been a much debated topic. A variety of techniques are used – ranging from percentage of the development effort to more refined approaches based on use case and test case points – depending on functional and technological complexity. Underlying testing normally focuses on end-user functionality.

Testing of master data management (MDM) applications is different. As such, it requires a different approach when estimating effort. In an MDM testing project, there are specific factors that impact estimation. They include:

- The effort needed to prepare scenario-specific test data and loading scripts.
- Script execution and data loading time.
- Availability of a separate MDM hub.

This white paper analyzes the impact of such factors, as well as the approach that should be adopted to estimate the effort needed for testing MDM solutions.

### Estimation Approach

System and integration testing in MDM focus on verifying the system functions, data quality, exception handling and integration of business

functions across the enterprise. The approach comprises the following steps:

- Collect input specifications.
- Compute MDM application size (this includes the ETL and MDM parts of testing) in function points.
- Determine the number of test cases for MDM testing (including ETL test cases).

The MDM test estimation approach highlighted in this document is aligned with the International Function Point User Group's (IFPUG) guidelines for function point analysis (FPA).

### Steps of Estimation Process Flow: Size Estimation

The input and output interfaces of the MDM application are counted, and the following general considerations are applied while calculating the function points:

- **Step 1:** Identify the Application Boundary for the MDM Project.

The application boundary determines the function points that need to be counted as part of the MDM application (including the ETL part). The application boundary indicates the border between the software being measured (in terms of testing) and the user and other

applications that integrate with the MDM application.

Figure 1 depicts the application boundary and counting scope of an MDM project. It contains the following:

- > ETL layer functionalities.
- > MDM and publish layer functionalities.
- > End-to-end application functionalities, including the ETL, MDM and publish layers.

• **Step 2:** Determine the Unadjusted Function Point Count

The unadjusted function point count (UFPC) reflects the specific countable MDM and ETL functionality provided to the user by the project or application. The user functionality is evaluated in terms of what is to be delivered by the application, not how it is to be delivered. Only user-requested and user-defined components are counted.

The UFPC can be counted by identifying and mapping different user-requested MDM functionalities using function point elementary processes. For example, an MDM testing requirement can be stated as, "Verification of customer master PKEY\_SRC\_ID formation as per business rule."

This requirement can be identified and mapped with a function point elementary process "External Output" (EO), as it involves querying and deriving data using business logic and, hence, fulfilling the necessary conditions for EO.

### Applying Size Estimation Technique in MDM Testing Projects

When it comes to testing types, the following options are considered for an MDM testing project:-

1. **Option A :** Database-intensive testing deliverables with data flow requirements for:
  - > Source to landing data loading (i.e., land process).
  - > Landing to staging data loading (i.e., stage process).
  - > Staging to base object data loading (i.e., load process).

Database-intensive testing is required to perform in each layer of data staging, as mentioned above. For example:

- > Data standardization and cleansing to be verified for the stage process.

### Identifying Application Boundary and Testing Scope

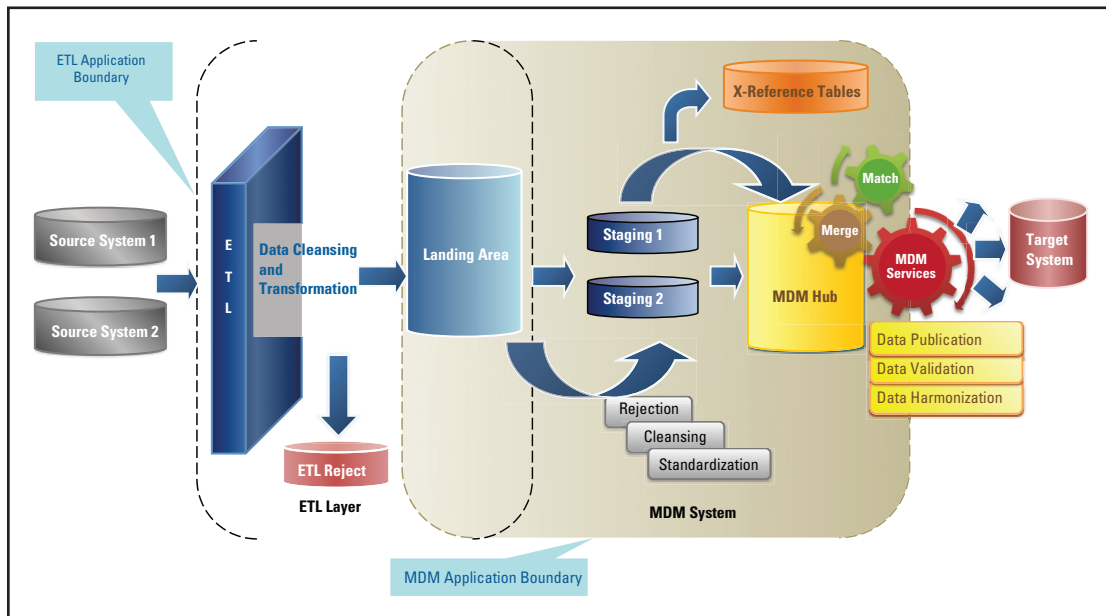


Figure 1

- Auto match and merge of data as per business rule to be verified for the load process.
- 2. **Option B:** UI console-based testing deliverables with the data steward-specific requirements, such as:
  - Manual match and merge of records as per business rule.
  - Trust rule verification for data from different sources.
  - Ability to create/edit new and existing records, etc.

Activities related to each of the above sections can be mapped directly with the elementary processes of function point analysis. For example, consider the following data standardization and cleansing requirement: "Customer address records should be free from junk characters (#, &, ^, %, !), and 'Street' should be displayed as 'STRT.'"

A simple SQL query will be implemented in the test steps in order to verify the above requirement. The query doesn't need to have any logical data derivation (e.g., concatenation or selecting a sub-string from the record) or mathematical calculation in order to verify the cleansing requirement. It is required to fetch the record as it is stored in the database as per the conditions stated in the requirement.

Hence, this functionality can be mapped against the FP elementary process "External Query" (EQ). Figure 2 provides a pictorial view to identify the elementary processes of function point analysis in such data migration activities.

- **Step 3:** Determine the Value Adjustment Factor
  - The value adjustment factor (VAF) indicates the general functionality provided to the user of the application. The VAF comprises general system characteristics (GSC) that assess the general functionality of the application. Examples of such characteristics are:
    - Distributed data processing.
    - Performance objective of the MDM hub.
    - Online data entry on the downstream applications.
  - The VAF can vary between 0.65 and 1.35.
- **Step 4:** Calculate the Adjusted Function Point Count (AFPC)
  - The adjusted function point count is calculated using a specific formula:
 
$$AFPC = VAF * UFPC$$
- **Step 5:** Normalize Using Test Cases
  - On obtaining the size of the application in terms of FP, the number of normalized test

## Identifying Elementary Processes for MDM Data Flow

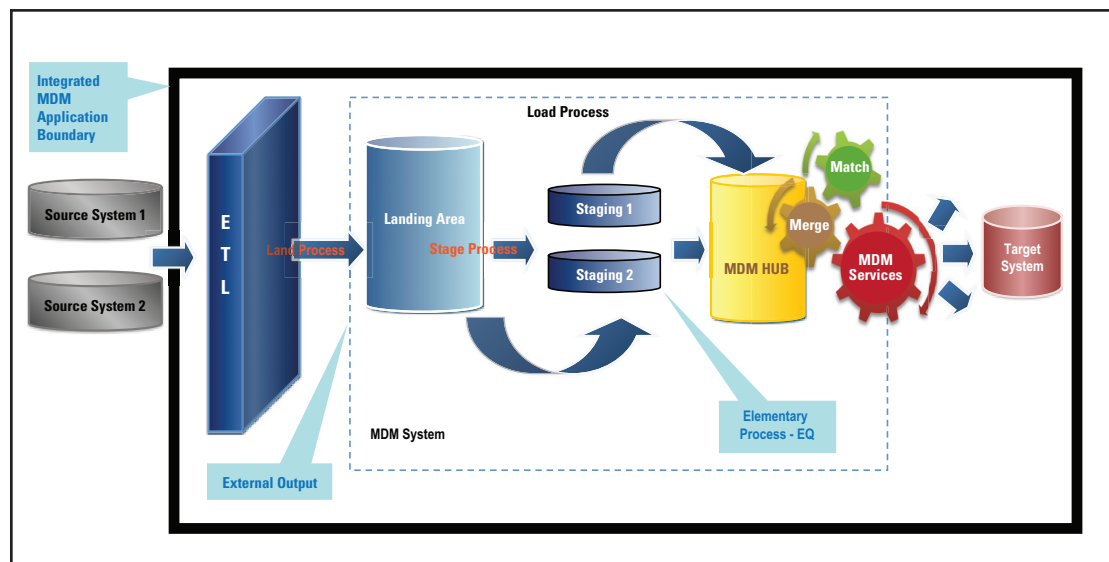


Figure 2

cases (manual) befitting the application is calculated using a formula proposed by and based on historical data from Capers Jones.

Adjusted Function Point Count	Number of Normalized Test Cases
AFPC	( AFPC ) ^ a

Note: 'a' is a factor that can be a range of value that varies with the AFPC.

### Effort Estimation

The effort estimation for an MDM testing project is computed on the basis of the Organizational Baseline Productivity (OBP) figures for MDM testing projects. The total effort required by the project based on productivity figures is as follows:

**Total Effort in Person Hours (PH) = Number of Normalized Test Cases / Productivity (in Normalized Test Cases per PH).**

It is a requirement to conduct a productivity baselining exercise within the organization that uses essential data from closed testing projects – namely, actual project size and effort data from the key members of closed projects. The final size is established in terms of normalized test cases and the effort in PH. The effort for test design and test execution needs to be captured separately in order to derive the productivity figure for each case. This yields the productivity data point for each case and project. The median value of these data points gives us the OBP for test design and execution.

**Common Factors for MDM – Testing Projects:** These factors always increase the effort required.

Factor Affecting Effort
Project management (strategy, planning, monitoring & reporting)
Quality assurance
Retesting, reworking & defect tracking
Training effort
Environment setup and integration with test management tool
Test data preparation

### Project Specific Factors for MDM: Testing

The impact of these factors varies from project to project. Based on the situation, these factors may increase or decrease effort.

Beyond total effort, a percentage of common factors and project-specific factors must be added in order to arrive at the final adjusted effort.

**Final Adjusted Effort = Total Effort + Total Effort \* (% of Common Factors + % of Project-Specific Factors)**

Factors such as initiation and planning, closure, number of iterations, etc. need to be considered separately and added to the above figure.

### Challenges

Having outlined the approach, it is still important to highlight that – unlike UI-intensive application testing – effort estimation for testing MDM applications is still a new concept. Estimation has many challenges, a few of which include:

1. Non-availability of industry-standard productivity values for MDM technologies.
2. Non-availability of detailed requirement specifications at the estimation stage.
3. The need for skilled function point counters for consistent size estimation, especially people with sufficient training and practice with counting rules.
4. The availability of subject matter experts for the application in order to get a logical view of the application.

### Final Notes

Based on the estimation approach highlighted in this paper, we have built a tool for MDM testing estimation. This tool not only provides simple interfaces to capture user inputs, but it also implements the calculations for effort estimation. Additionally, it addresses the majority of the challenges mentioned above by making realistic assumptions based on our rich experience with MDM application testing.

## About the Author

*Prabuddha Samaddar is a consultant who leads Cognizant's MDM Testing Team within its Customer Solution Testing Practice. He functions as Cognizant's MDM testing subject matter expert. Prabuddha has in-depth knowledge in different estimation techniques, such as function point analysis, and rich experience developing estimation models, writing white papers on estimation and presenting estimation capabilities to clients. He can be reached at [Prabuddha.Samaddar@cognizant.com](mailto:Prabuddha.Samaddar@cognizant.com).*

---

## About Cognizant

Cognizant (NASDAQ: CTSH) is a leading provider of information technology, consulting, and business process outsourcing services, dedicated to helping the world's leading companies build stronger businesses. Headquartered in Teaneck, New Jersey (U.S.), Cognizant combines a passion for client satisfaction, technology innovation, deep industry and business process expertise, and a global, collaborative workforce that embodies the future of work. With over 50 delivery centers worldwide and approximately 118,000 employees as of June 30, 2011, Cognizant is a member of the NASDAQ-100, the S&P 500, the Forbes Global 2000, and the Fortune 500 and is ranked among the top performing and fastest growing companies in the world. Visit us online at [www.cognizant.com](http://www.cognizant.com) or follow us on Twitter: Cognizant.



### World Headquarters

500 Frank W. Burr Blvd.  
Teaneck, NJ 07666 USA  
Phone: +1 201 801 0233  
Fax: +1 201 801 0243  
Toll Free: +1 888 937 3277  
Email: [inquiry@cognizant.com](mailto:inquiry@cognizant.com)

### European Headquarters

Haymarket House  
28-29 Haymarket  
London SW1Y 4SP UK  
Phone: +44 (0) 20 7321 4888  
Fax: +44 (0) 20 7321 4890  
Email: [infouk@cognizant.com](mailto:infouk@cognizant.com)

### India Operations Headquarters

#5/535, Old Mahabalipuram Road  
Okkiyam Pettai, Thoraipakkam  
Chennai, 600 096 India  
Phone: +91 (0) 44 4209 6000  
Fax: +91 (0) 44 4209 6060  
Email: [inquiryindia@cognizant.com](mailto:inquiryindia@cognizant.com)