





Automating MDM Processes: Build vs Buy

MDS Business Rules vs Maestro Workflow

 *Microsoft Master Data Services (MDS) is a powerful master data management (MDM) platform, but it lacks an enterprise class workflow capability to support the processes of creating and managing master data. Companies that want to use MDS as an enterprise MDM platform may be looking at a choice between building workflows on top of the business rules and notifications available in MDS, or buying an integrated off-the-shelf solution, such as Master Data Maestro Workflow. This paper looks at the role of workflows in enterprise MDM and presents the challenges inherent in governance-driven workflow management. It details the specific requirements for initiating, orchestrating and managing the contribution and approval tasks of the workflow data steward; the requirements for performance and workload management; and the requirements for the development and maintenance of the underlying workflows, forms and management tools; and presents comparative build vs buy cost estimates for meeting these requirements. With this information in hand, you will have a solid understanding of exactly what you would have to build to deliver the required workflow functionality on top of MDS, and the inherent limitations of that approach. You will also have a detailed picture of what you gain from investing in a Master Data Maestro Workflow solution for MDS.* 

Executive Summary

Microsoft Master Data Services (MDS) is a powerful master data management (MDM) platform, but it lacks an enterprise-class workflow capability to support the processes of creating and managing master data. While basic workflows can be manifested through the use of business rules and notifications in MDS, this approach quickly breaks down under even modestly complex workflow requirements. What is needed is an enterprise-class workflow capability that is scalable, flexible, and tightly integrated with the MDM solution.

Companies that want to use MDS as an enterprise MDM platform may be looking at a choice between building workflows on top of the business rules and notifications available in MDS, or buying an integrated off-the-shelf solution, such as Master Data Maestro Workflow. In our introductory paper in this series [“The MDS Desktop User Interface: Build or Buy?”] we discussed what is involved in making a build-versus-buy decision when it comes to the MDS user interface (UI). At a high level, that paper laid out important user experience considerations for two primary MDM user scenarios, power data stewards, and casual or workflow data stewards. A second paper [“The MDS User Interface: Path to Optimal MDM Value”] set out detailed user experience requirements for supporting **power data stewards**, as a basis for making a build vs buy decision for the MDS user interface.

In this paper, we take a deeper dive into requirements for initiating, orchestrating and managing the contribution and approval tasks of the **workflow data steward**, whose role in the enterprise MDM solution is one of infrequent or limited interaction with the data, typically focused on specific types of tasks, and requiring impetus to take action when input is needed. We will also look at the requirements for workflow management, and for the development and maintenance of the underlying workflows, forms and management tools. These requirements are intended to reveal the totality of the buy vs build decision – including the 90 percent of the iceberg below the surface – and provide you with important insights for navigating the decision-making process. With this information in hand, you will have a solid understanding of exactly what you would have to build to deliver the required workflow functionality on top of MDS, and the inherent limitations of that approach. You will also have a detailed picture of what you gain from investing in a Master Data Maestro Workflow solution for MDS.

Introduction

The Role of Workflows in MDM

An enterprise-wide governance function should define the processes through which master data will be managed. It is the role of MDM technology to implement and enforce adherence with those processes. Implementing robust workflows within your MDM solution will help control costs, minimize risk, drive efficiency, and grow productivity. This requires a much more sophisticated set of workflow capabilities than is provided by MDS' business rules, as well as better visibility into the performance of the organization, allowing the organization to amend governance processes as needed, based on how well the current ones are being executed.

The challenges inherent in governance-driven workflow management fall into four general categories:

1. Real-time processing

MDS business rules must be executed to pump data through the next step of a process after a user adds or modifies data. With an ideal workflow solution, records automatically progress through the workflow on a real-time basis as tasks are completed.

2. Complex business modeling

MDS requires many business rules with complex conditions and actions to model a business process, and a cumbersome UI makes it difficult to maintain the logic. Ideally, a drag-and-drop workflow designer should allow processes to be visually modeled in an intuitive way.

3. Process visibility

With MDS business rules, there is no out-of-the-box way to gain visibility into the state of data within a process, nor to see who or what may be holding up the process. What is needed for effective workflow management is visibility into each stage of the workflow, showing who is responsible for pending tasks, the task priority and the due date.

4. Workload management

With MDS business rules, designated recipients of notifications are statically defined, and cannot be modified dynamically. Ideally, tasks should be prioritized, organized and assigned based on current business needs, workloads, and resource availability.

Workflow in common MDM domains: Product and Customer are the two most commonly implemented MDM domains. As can be seen below, automated workflows are a critical need in both of these domains.

Product

The creation of a new Product, from initial inception to final approval, can involve many different actors across the organization. In the past, this process has been managed via email and/or spreadsheet, which most frequently results in a disjointed process, as there is no automation or technology oversight. Automated workflows allow management to create and assign tasks to the different groups of actors in the organization, and to initiate required action. As each assigned task is completed, the process immediately moves the work forward to the next task or tasks, to be handled by the appropriate actor. Formal approvals can also be included in the process, providing centralized control over determination of which finalized Products are made available to the rest of the organization.

Customer

For the Customer domain, the value of automated workflows can be clearly seen within the data onboarding process. As a new Customer is created, associated data on address, name, phone, and/or email may need verification. Based on the outcome of that verification, some stewardship may be required to address identified issues. Then the new Customer record may need to be matched, and, based on the matching results, possibly reviewed, to ensure that it is not a duplicate of a record already in the system. Additionally, new source and/or master Customer records may need to have category or hierarchy information attributes added.

To manage the workflow challenges within both of these common MDM domains, you will need the tools to model complex process requirements and drive collaborative activities within your organization.

Workflow Perspectives and Considerations

In order to evaluate a build vs buy strategy for integrating workflows into your MDM implementation, you will need to look at workflow requirements from several different perspectives:

- **Workflow development requirements for process automation, implementation and enforcement**

It is common for development teams to approach workflow development as simply an attribute of an application, making it “workflow-enabled;” for example, “making a portal to manage some products.” This approach begs the question of how to enable and facilitate collaboration, or how to adapt existing workflow processes to changing business requirements or system migrations. Further, it offers no governance visibility, nor tools for overseeing and managing processes and performance.

Consideration must be given to how you will orchestrate independent tasks to be done in parallel, only moving forward within the workflow once both tasks are complete; also, how will you create workflows that allow a number of users to work on a specific task, accommodating partial completion by one user, leaving the task open until another user completes it.

- **User requirements for casual or workflow-driven data stewards, whether participating as independent actors, or in collaboration with a team**

Workflow data stewards have infrequent interaction with the data, so they need direction; they need impetus whenever their input is required, ideally via a workflow-centric interface to orchestrate and enable their work. With this approach, specific tasks are pushed to specific users, who are presented with task-specific forms and/or web portals through which to complete their discrete tasks – whether contributing, approving, or reviewing – on an individual record, or on related records. When their contribution to the process is completed, the task is then automatically routed to the next user to complete their part of the workflow. To support these users, it is important to be able not only to initiate tasks, but also to transparently orchestrate required tasks across users in multiple departments, while moving work through the flow as efficiently as possible, without gaps or delays.

- **Management requirements: task and team management; workload and performance analysis; workflow change management**

All of the workflow tasks and user activities must be visible for management purposes; what tasks are in flight, which data stewards are responsible for what, how they are performing, where there is a need to balance workloads. The manager needs to be able to track exactly what has occurred and is occurring with the workflows at all times. Further, the manager must also be able to make changes in the workflows as needed to accommodate changing business requirements.

Workflow Requirements

The requirements pertinent to addressing each of these perspectives are presented below. Detailed requirements are enumerated for each perspective, and we also present comparative estimated costs to build vs buy for each.

1. Workflow Development Requirements

Cost to Build	\$\$\$\$\$
Maestro Cost	\$\$

Aspects of stewardship that can be automated should be automated by the workflow. Any activities that can be performed systematically without user interaction should be able to be executed behind the scenes. In order to do so, the workflow will need to be extensible, able to interact with other systems and applications.

1.1. Requirement: Workflow should leverage core MDM functions, such as matching and address verification

Systematic processes that can be run behind the scenes should be automated by the workflow. For example, as a core MDM function, a user shouldn't have to manually kick off address verification. At the point in the process where you have established that you want to run address verification, the workflow should start it automatically. The same thing applies to the matching function.

1.2. Requirement: Workflow should have the ability to automate calculations, as well as any data changes that can be done via calculation or logic

The workflow should be able to perform all CRUD operations on master data records, as a part of the process, for operations that can be performed and calculated behind the scenes. The user should not be required to manually perform calculations in order to derive and record field values; for example, adding values from two fields, then entering the sum in a third field. The workflow should be able to take the inputs, calculate the value, and update the data, without requiring user intervention.

1.3. Requirement: Workflow should have the ability to call out to/communicate with other non-MDM systems and processes

An example of this extensibility would be the workflow calling out to an external data quality service that the organization has licensed, for example, D&B, then bringing the returned data into the master data hub, all as part of the automated workflow process itself.

1.4. Requirement: Workflow backend capabilities should provide a way of handling workflow changes necessitated by business changes

When inevitable business changes require changes to be made to the workflows, the workflow backend should provide a mechanism for handling those changes such that any work already in process can be alternatively left to complete under the old flow or migrated to the new flow, at the same time allowing new flows to be defined and new tasks to be created under the new flows.

2. User Requirements

Cost to Build	\$\$\$\$
Maestro Cost	\$

Regardless of whether the user's role is one of contributing information or one of approving information, there are common requirements for the workflows that inform their activities.

2.1. Requirement: Common requirements for BOTH Contributing and Approving tasks

Regardless of whether the user's role is one of contributing information or one of approving information, there are common requirements for the workflows that guide their activities.

2.1.1. Requirement: Users receive proactive notification of required action

The user must be proactively notified by the workflow that there is something to be done as soon as the task becomes available to work on. This means that, for all user scenarios, notifications should be *generated* as soon as the task becomes available for the user to work on; the timing of actual *packaging* and *delivery* should be configurable based on the specific user scenario. The user looking for work that is ready to be performed should not have to wait, for example, until after business rules run overnight to see that a task is available. For high priority tasks – such as a task at the final approval state – a notification should be sent immediately to the approving data steward(s) for prompt action.

Workflows should also be able to notify users of particular events occurring within the workflow that aren't necessarily related to a task, such as a specific condition being arrived at in the workflow. For example, the workflow should be able to notify a manager that someone didn't complete their task, the task expired, and now it has to be escalated. The notification should be configurable to allow a power user to specify the message sent to the user as a part of the notification, and the timing of that notification.

2.1.2. Requirement: User is presented with a form specific to the required task

The presentation of contextual forms is a key requirement for workflow users. For example, if the user is required to contribute information about a product that has 200 attributes, the workflow should be able to present the user with just the relevant information, the ten things that user is concerned with for this specific task. Also, you will need to have a different form for each type of task. For example, the form presented for a Marketing task is different from the form for a Finance task, and you will need to develop/provide a way to manage the necessary definition of those forms. Ideally, the forms management function should be able to be accomplished by a power user, and not require a developer. For example, as the data model evolves, the process evolves, and a new attribute is identified that Marketing should be contributing data to, the power user should be able to easily edit the appropriate form, add the new attribute, and have everything automatically start showing up for the users of the Marketing form.

The forms must be renderable in both Web and desktop client environments so that both power users on the desktop client and casual users on the Web have the same experience within the appropriate client form factor. The form should also have

intelligence built into it that applies logic as to whether or not that form is complete. Required attributes should be flagged, so that the task can't be marked complete until the requirements of the form are met. You should also be able to flag specific attributes as having to be validated under business rules before the task can be marked complete. For example, not only is color a required attribute, but the value specified for color must also pass data quality checks – be an appropriate color for the item in question. Finally, data within the form should be laid out in such a way that it is logical to the user; ideally in a contextual two-dimensional form where data elements are arranged in rows and columns, logically.

2.1.3. Requirement: User is given specific duration of task

The workflow should include information on the timeframe within which a task must be completed, and should be able to track whether the designated timeframe has expired. When the task expires, the workflow should be able to detect the expiration and support a variety of alternate paths for handling the expiration in whatever manner the business requires; loopback and reassign the same task, create a different task for someone else to perform, ignore the task and just continue, and so on. If the result is that, when one task expires, you create an escalated task, you should also be able to link those tasks together, so that an expired task can be related to a new task that was created by it.

2.1.4. Requirement: User is provided with a portal to access assigned tasks

The workflow should provide a single view where the users can go to see a complete prioritized list of the tasks that are assigned to them. From within this view, they should be able to view their full workload, find any subset of that workload that they'd like to focus their efforts on, open assigned tasks and complete them. Therefore, the users should be able to sort and filter the list, to determine what they want to work on, in what order. For example, they should be able to filter by priority, by specific tasks, or by the model or the entity to which that the task is related.

2.1.5. Requirement: Users have the ability to collaborate on a task

Users should be able to collaborate within a workflow, within a specific task. The users must have visibility as to who else is working on the task, and whether any portion of the work has been completed; the users must be notified of the rank of importance of the task in relation to other workflow assignments. The users should have visibility to a dialog or timeline of events that have occurred on a given record or member. For example, for one user to be able to say, "I rejected it because...", and for another user to be able to access that information. The user should have visibility into all the changes that have been made on a particular master data record, and be able to see an accounting or history of what has actually been done at the data level.

2.2. Specific Requirements for Contributing information

A number of workflow requirements are specific to the task of contributing information to the model.

2.2.1. Requirement: Team of individuals must have the ability to work together to complete task, with the ability for one individual to partially complete a task, and pass it along to next contributor

This requirement includes the ability to assign a task to multiple contributors. The task may be accomplished through the combined efforts of these contributors (see “collaboration” under requirement 2.1.5), and any one of the contributors should have the ability to complete the task. The workflow should also allow a user to partially complete a contribution task, and not be required to be the one to complete it. In this way, one user would be able to open up the task, do what they can, save it, and the workflow would allow for another user to come back in and finish it.

2.2.2. Requirement: User must have the ability to record task completion info

The workflow must provide the ability to record who worked on a task, who completed it, and when it was completed. That information must then be exposed and available to management to provide visibility into overall performance metrics – who is effectively doing their job, who is not.

2.2.3. Requirement: User must have task completion requirements stipulated by the workflow

The workflow must be able to stipulate what is required for completion of a contribution task; for example, specify which attributes are required, and stipulate that the attributes must pass data quality checks prior to attributes being populated, in order for the task to be considered complete.

2.2.4. Requirement: User must have the ability to filter assigned tasks

The workflow should give the user the ability to filter assigned tasks to focus on a specific subset, by type of task, status, date, etc. Users should have a dashboard that gives them a high-level view of what’s on their plate over the coming days, weeks and even months, allowing them to balance their MDM workload with other activities.

2.3. Specific Requirements for Approving information

Many of the workflow requirements for contributing information to the model also apply to the task of approving information, with slightly different nuances.

2.3.1. Requirement: Multiple users must be able to be assigned as potential approvers, and must be notified of required minimum number of approvers

The workflow must allow the approval task to be assigned to multiple users, and needs to stipulate how many approvals are required to consider the task – the approval – complete. For example, in a financial department where it isn't desirable to have one person be able to both create and approve something, appropriate checks and balances should be factored into the workflow so that the approval task can be assigned to multiple people in the department, with the requirement that at least two have to approve it in order for the task to be considered complete.

2.3.2. Requirement: Users must be able to reject an approval, and have the workflow detect and handle the rejection

The workflow must know how to detect and account for a rejection, and have the necessary notification/escalation capabilities to take appropriate action, based on the rejection. This may include looping back in the workflow when there's a failure condition, or stopping altogether. The workflow may also be required to notify someone that there was a rejection, if one occurs.

3. Workflow Management Requirements

Cost to Build	\$\$\$\$
Maestro Cost	\$

All of the workflow data steward activities need to be stored somewhere and be visible for management purposes. The manager needs to be able to see what tasks are in flight, which data stewards are responsible for what, how are they performing, and where there is a need to balance workloads. The manager needs to be able to track exactly what has occurred and is occurring with the workflows at all times. This will require development of a database and data model for capturing and storing the information, as well as a front-end – e.g., a dashboard – and reporting mechanism to expose the data. The manager must also be able to respond to changing business requirements, making necessary changes in the workflows to accommodate these changes as needed.

3.1. Requirement: The manager must be able to track workflow data steward performance

The manager must have visibility into the performance of each data steward – to know, for example, which CSR is closing the most customer record cases, and who is not closing any. The manager must also be able to track the status of all workflows in a given process at all times. The available information should include metrics such as which data steward has the most escalations; how many workflows are in process and how many tasks are open at any given time; how many tasks are on each data steward's plate; and so on.

3.2. Requirement: The manager must have the ability to assign work and manage the data stewards' workloads based on performance tracking

If performance tracking reveals that the data stewards' workloads are unbalanced, or that task assignments need adjustment to maximize performance, the manager should be able to readily manage task assignments, reassignments and overrides from within the management dashboard.

3.3. Requirement: The manager must have workflow change-management capabilities

The manager must have the ability to quickly respond to changes in the business that require changes to the workflows. Such changes may require different workflow behaviors due to changed priorities; they may require modifications to forms; they may mean changing the people who have been assigned to specific tasks; or overriding task requirements.

The manager must have the administrative capabilities to establish task defaults in the initial configuration of the workflow, based on then-current business conditions. For example, a workflow might be configured by default to start out as a priority 2, and have five people assigned to do the work, using a specific predetermined form. If conditions change once the workflow is in flight, the manager may need to change the priority, or assign it to a different group of people, for example. In this case, the manager must be able to modify workflows and tasks in-flight, so that they reflect the latest conditions.

These kinds of management override capabilities also enable the manager to declare a task complete, even though default completion requirements are not met. For example, where color is a required attribute, but there is no color for a given product, the manager can declare that this case is an exception, override the requirement, and declare the task complete.

Conclusion

It is the role of MDM technology to implement and enforce adherence with enterprise-wide data governance processes. Implementing robust workflows within your MDM solution will enable you to automate the implementation of approved processes, helping you to control costs, minimize risk, drive efficiency, and grow productivity. This requires a much more sophisticated set of workflow capabilities than is provided by MDS, without costly custom development. Careful consideration of what will be required to meet workflow requirements for developers, users, and managers will help you determine whether it makes sense to undertake such development ('build') a solution, or take advantage of an off-the-shelf solution ('buy'). Even a conservative assessment of the build vs buy cost comparison reveals a ratio of 3:1 in favor of a buy decision, based on the out-of-the-box functionality of Maestro Workflow; that is assuming that the necessary internal development resources are readily available, and have deep knowledge and expertise in both MDS and master data management implementation. Time-to-value considerations further impact the buy vs build decision in favor of an off-the-shelf solution, as does the cost of ongoing maintenance of an internally-developed solution.

For more information about Master Data Maestro and the Maestro Workflow solution, visit www.profisee.com.