

Master Data Management —The Build vs. Buy Decision

SWP245D

Setting the Stage

Master Data Management (MDM) is top of mind with today's companies. They recognize that effective management and analysis of master data and hierarchies across the enterprise will enable them to be more agile and effective in planning and executing their business strategies, and more accurate in measuring and improving their business performance. Along the path to recognition, most companies have run headlong into difficulties in defining and synchronizing master data across their various operational systems and business performance management (BPM) applications. For many, the answer to these challenges lies in Master Data Management, a software solution for proactively managing the commonly shared data pertaining to customers, products, accounts, locations, and other business domains. For these leading-edge companies, the only question left is whether to build or to buy an MDM software solution. In this paper we will explore the trade-offs of building versus buying and provide guidelines for making the right decision for your organization.

The Case for MDM

This white paper assumes that the need for MDM has been established and is well understood within your organization. If not, or if you'd like more information about the role that MDM plays in successful organizations, you may find it helpful to review The Case for Master Data Management, a white paper from BPM Partners. To request a copy of this white paper, visit www.microsoft.com/mdm.

Determining the 'Real' Requirements

Before embarking on a build versus buy assessment, it's critical to develop a clear understanding of MDM requirements and solution objectives.

The following are generally acknowledged as being core requirements for MDM:

- Creation of a data model and repository to manage specific master data, including members, hierarchies and attributes
- Online viewing and editing of master data
- Integration with downstream systems for harmonization of master data (the end goal)

What many IT organizations fail to recognize are the additional requirements related to extensibility, data governance and usability that are critical success factors of an MDM solution. If the delivered application cannot be rapidly expanded for new data sets or requirements then it will become outmoded and unsuccessful as the business process changes. If appropriate data governance controls are not in place to impose security and auditability, then additional work and/or systems will be required to ensure regulatory compliance.

The following latent requirements will ensure application viability and success by satisfying the users and business processes on an ongoing basis:

Model Adaptation

This refers to the ability to rapidly expand and restructure the data model itself to mimic a rapidly changing business. Any inflexible, fixed-model MDM solution—whether home-grown or commercial—will surely be abandoned at the first significant change in the business, if it is not in fact obsolete before it is even implemented.

Secure Collaboration

A usable solution must facilitate the "business process" of collecting information not only from source systems, but also from the knowledge workers of the organization. It must therefore provide a collaborative yet secure working environment. Granular security must ensure that only users with proper security can view and update specific members, attributes and hierarchies.

Usability and Performance Optimization

It is important for users to be able to see data sets and hierarchies as they are updated. To ensure acceptance, the solution must utilize an intuitive web interface and, at the same time, provide optimized performance. Users should accomplish their edit tasks with minimal clicks and screen navigation. Pages should load quickly and efficiently, regardless of data size and complexity.

Notification and Workflow

The system must have automated notification and workflow management features to solicit worker knowledge in a timely fashion, to gather approvals, and to ensure that complex validation issues are resolved by the appropriate users. One-click access should be provided, taking the user directly from an e-mail notification to the data element requiring action.

Business Rules and Validation

To ensure that subscribing systems receive data and structures that are complete and correct, prescribed interrelationships and standards must be enforced as the master data is collected and updated. Conditional logic must support the lifecycle of data members because required attributes and acceptable values may vary as products, customers or accounts age from new to inactive.

Import and Export

The system must provide mechanisms for (1) initial bulk loading of data by business users; and (2) ongoing integration of feeds from source systems with full transaction control and integrity checking. Export capabilities must provide for denormalized and flattened output views of members and hierarchies, produced in the various formats needed by subscribing systems.

Audit and Traceability Compliance

Data structures drive the financial and non-financial reporting results of the company; therefore, it is crucial that the MDM system enforce SOX and other regulatory compliance. An MDM solution must incorporate audit and traceability features that provide control and transparency to current, past and future master data and structures, as well as to their incremental evolution.

Version Control

Business processes typically follow repeating cycles, such as the month-end close process, or a quarterly forecasting process. A version history must be maintained to support these business cycles. Past versions must be preserved to prove past reported results.

This myriad of often overlooked requirements must be considered up front to expose future hidden costs of custom development.

Comparing Time and Cost

The first considerations in build vs. buy are the overall implementation time and costs, as well as the ongoing costs of enhancement, maintenance and technical support.

For purposes of an accurate comparison, it is important to first set a baseline expectation of the phases and costs associated with the custom development process. A mature custom development process will

incorporate at least the following major project stages for initial implementation, *and for each subsequent round of enhancement*:

- Requirements analysis and application definition
- Research and design
- Development
- Incremental unit testing and review
- Quality assurance testing
- User acceptance testing
- Deployment and training

The cost of internal developers cannot be ignored, as well as the time contributed by business users in the definition, testing and other phases of the project. The use of outsourced development resources will not dramatically alter cost estimates for purposes of comparison with purchased software.

For purposes of evaluating purchased software, implementation includes the software purchase price, implementation activities and maintenance cost. Here, the primary factor is the flexibility of the product to fit your business, both at initial implementation and on an ongoing basis.

Offshore Development

While offshore development (OSD) is a good alternative to expensive consultants and may be more cost effective in some cases than using internal developers, OSD typically achieves its greatest success and economy with large-scale projects that are well constrained and articulated. MDM does not fall into this category as it is closely tied to rapidly changing business processes.

Time to Complete — Core Requirements

- Build: 6 - 12 months, elapsed
- Buy: <1 month, elapsed

Although the advancement of development tools and techniques may support rapid development efforts of as little as two months, recognize that development itself may represent less than half, or even as little as one third, of the total "build" initiative when you account for significant additional tasks defined above as part of the custom development process.

Considering the "core" requirements of MDM and assuming that one of the "latent" requirements would also be tackled, you could easily see a custom development effort of 4 to 6 man-months. If you encapsulate all time spent across the business, including the time required for the supporting activities of business users, project managers, architects, quality assurance staff and others, you can expect a total effort of

Customization Considerations

Be wary of MDM software vendors who propose lengthy service engagements to implement the initial subject domain. This would contradict the assumption of easy customization, and could be a red flag regarding ongoing usability of the software—indicating that master data and structures are not likely to be business user-maintainable.

12 to 18 man-months. And while elapsed project time might be compressed to 6 to 12 months, initial requirements gathering, and even the marshaling of resources, can significantly delay the start of a custom development effort.

Requiring your MDM solution to be completely and readily customizable—as it should be, based on the latent requirements for a successful long-term solution—will add substantial development time to a custom development effort; yet this same feature in a purchased software product will ensure rapid implementation and easy tailoring to meet your specific business needs. MDM solutions that support rapid model adaptation and integration will have proven initial implementations within a month.

Cost of Implementation — Core Requirements

Build: \$\$\$\$
Buy: \$\$\$

When comparing up-front implementation costs, consider all development and implementation activity for building a solution, then compare this with the cost of software licensing and implementation effort for purchased software.

Those involved in a custom development project are generally well-compensated professionals with an average fully absorbed cost of \$100 to \$150K annually. Therefore, 12 to 18 man-months (as defined for a minimal effort that incorporates one of the established latent requirements for an MDM solution) would translate to \$100 to \$225K of human cost. And while the time spent by staff outside of IT may not be tracked as hard-dollar costs, the cost to the business when users support analysis, design, testing and deployment tasks over an extended period may be even greater in terms of opportunity loss.

As a relative benchmark, an MDM software solution that does meet all of the latent requirements can be purchased and implemented well within the same cost range, and will be far more usable from the outset. This rapid implementation will drive a number of benefits beyond reduced cost. The time to achieve a

return on your investment (ROI) will be much less—typically within the first year. The up-front costs of licensing are quickly offset by productivity gains and process improvements. Additionally, with a shorter implementation window, the risk of major changes and disruptions within the business during the course of the project is greatly reduced, thus ensuring a consistent project team, dedicated effort and stable requirements—all of which contribute to controlling costs.

MDM vs. ERP

Typically, custom software development is justified on the basis of cost savings amounting to several times the annual cost of one employee. This is based on the astronomical cost of purchasing and customizing an ERP system, in which case, is a compelling counterweight to the risks and overhead associated with custom development. However, an MDM solution does not fall into the same category as ERP.

MDM software should be provided with a fully extensible framework for the data model and business rules and all of the screens, views and other functionality should be driven from the model. Therefore, the usual paradigm is reversed; custom development, not purchased MDM software, will drive spiraling customization costs.

Risk of Failure or Overrun

Build: High
Buy: Almost none

The many latent MDM requirements described above are often omitted from the scope of an initial custom development project in order to speed delivery of a solution. While such a solution may achieve some level of success early on based on meeting initial requirements, it is unlikely that it will support rapid future evolution of the model without introduction of a subsequent implementation cycle, including all phases of development, testing and deployment.

On the other hand, when approached properly, the rapid customization and implementation capabilities of purchased MDM software can deliver a proof-of-concept (POC) in less than a week that will ensure the solution incorporates end-user acceptance criteria, integration requirements and acceptable performance levels with the client's actual data and volumes. The POC may be conducted prior to software licensing and thus eliminate the key risk factors associated with full implementation. Additionally, achievable expectations will be set for all involved in the implementation.

Resource Optimization

Custom development may provide an advantage in functional applications that are unique to a particular business or industry, and where the development staff has strong internal knowledge of the functional area. In contrast, MDM applications have a consistent set of requirements across a broad range of industries, and traditional IT departments do not have experience and best practices in this area. Furthermore, there is a significant opportunity cost in redirecting development resources that provide their greatest value in support of specialized mission-critical operational systems.

Cost of Implementation — Latent Requirements

Build: \$\$\$\$\$\$
Buy: \$

Many organizations will severely limit scope to achieve a manageable "build" implementation. Waiting to tackle key requirements related to extensibility, data governance and usability will ultimately lead to failure and/or cost overruns for your MDM solution. If the delivered application cannot be rapidly expanded for new data sets or requirements based on changing business conditions, then it will become outmoded and unsuccessful as the business process changes. If appropriate data governance controls are not in place to impose security and auditability, then additional work or systems will be required to ensure regulatory compliance.

In tackling all core and latent requirements, it is unlikely that an initial "build" effort can approach the success of a mature product in a "buy" approach, especially in an initial attempt. "Build" efforts will rely on reduction of scope for a multi-phased development approach and several phases will drive complex and lengthy lifecycle development processes. The overlapping of features exponentially drives up the complexity of software, requiring significantly more testing, performance tuning, documentation and maintenance as the application grows.

Therefore, while the cost of using additional software features in the "buy" approach is a negligible incremental cost, the incremental cost of extending the "build" solution to meet the latent requirements can be several times the initial cost of the core implementation.

Cost of Maintenance and Enhancement

Build: \$\$\$
Buy: \$

It is not a matter of "if", but "when" and "how much", MDM application requirements will change and grow since they are driven by your rapidly changing business environment. When the time comes to improve the business process by expanding beyond the original scope of the MDM system, the required changes to the data model will add significant cost to the custom development effort.

If only minor changes were introduced each year, perhaps only 4 to 6 weeks of new development would be required—but this is only the tip of the iceberg. Add to that all the related technical management activities as well as the communication, review and testing required by the business users, and 12 to 18 weeks of total effort can be expected. Also, if schemas change significantly to accommodate new functionality, you will have to consider the cost of data migration. Assuming one minor revision cycle per year, a minimum annual maintenance cost of \$25K to \$55K would be expected for a custom development effort. Of course, this cost does not include any major functionality enhancements as outlined in the previous section. One major revision cycle per year would reasonably drive a cost range of \$75K to \$165K annually.

By contrast, properly designed MDM software will allow business users to change the structure of the data model, its associated business process, and related security as dictated by changing business requirements, without the penalty of a custom development cycle or migration of data. These tasks, as well as testing and deployment activities, should be able to be performed by a non-technical "application administrator" rather than requiring specialized DBA or development skills. This difference in approach will create significant ongoing cost savings and even prevent a system from becoming obsolete for failing to keep up with the business.

Carrying forward our assumptions of initial cost, maintenance for purchased MDM software can be expected to range from \$20K to \$50K annually. In contrast to a custom development effort, a purchased solution would include one major release each year that introduces substantial functionality enhancements, several optional minor releases introducing usability improvements, and effortless upgrade routines for a seamless transition to the new version. Additionally, with the ability of business users to respond to changing or expanding business requirements, support activities will be minimal.

Total Cost of Ownership

Build: \$\$\$\$\$\$\$\$\$\$
 Buy: \$

In considering overall cost of ownership, account for the initial costs of development or purchase plus all implementation costs, as well as recurring support, maintenance and enhancement costs over a period of at least five years. This will be a good indicator of the total cost of living with the solution.

By considering only a limited set of initial "core" requirements, "build" may appear similar in cost to "buy"; however, the vastly shortened implementation times and high success rate with "buy" will yield fast and proven ROI. If expansion of the solution requires a full lifecycle development process, the "build" cost becomes a multiple of the number of subject areas tackled. For example, tackling Products, Customers and Chart of Accounts as three MDM subject areas can lead to a "build" approach which is at least three-times more expensive than "buy." Furthermore, adding the complexities of the latent requirements described for extensibility, data governance and usability will require multiple phases with a "build" cost that is at least 10 times the cost of "buy."

Licensing Considerations

When evaluating an MDM software product, ensure that license fees do not escalate with additional users or data volumes. Increasing users or data within the context of an existing model does not add new value sufficient to warrant a major cost increase. Additionally, no premium should be paid for additional test / development servers, as these are a standard requirement in most organizations.

Comparing Results

The second consideration for build vs. buy is comparison of the projected results of each approach. Given the higher total cost of ownership, longer implementation time and increased risks represented by a custom development process, consider whether the fit and benefits can outweigh the alternative.

Tailored Fit of Solution

Objective	Build	Buy
Meets initial business requirements	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Customized to unique business model and process	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Meets future business requirements	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Adapted by the business as the model and process change	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Custom Development

- Custom development is often sought when packaged products do not support unique business processes or data structures.
- A "custom-built" solution is actually less customizable than the packaged software solution, due to time and cost inherent in the software revision and deployment cycle.
- Users tend to stop requesting changes to business process or data structure due to back-log of requests and the inability of IT to respond in a timely manner—hence, they stop using the system.

Purchased Software

- Appropriately customizable MDM software will not assume a pre-existing data model or a fixed scope; it is designed for tailored implementation.
- Flexible MDM software is designed to cope with the changing nature of your business.
- MDM software provides the ability for nontechnical business users to revise the model without introducing a cycle of development, testing and deployment.

Best-Practice Adoption and Process Management

Objective	Build	Buy
Encapsulates industry best-practices	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Many users/companies driving common enhancements	<input type="checkbox"/>	<input checked="" type="checkbox"/>
ROI in first year	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Custom Development

- Custom development resources will often tackle MDM problems for the first time in their careers, and the requirements are often misunderstood.
- The focus is put on removing pain around master data integration and updates; however, the opportunity to incorporate best practices and new data modeling paradigms is missed.

Purchased Software

- The purchased solution will encapsulate best practices and the software provider will have a proven track record and focus in MDM.
- MDM software goes beyond removal of pain to provide value-added master data, hierarchies and attributes to subscribing applications, delivering significant productivity gains.

Performance and Optimization

Objective	Build	Buy
Design proven in use by numerous sites	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Designed by BPM implementation and integration specialists	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Architecture optimized for collaborative master data review and maintenance	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Custom Development

- The ultimate performance of the application is an unknown factor in custom development and could add significantly to development and testing efforts, especially when balancing rapid web-based delivery with efficient handling of large record sets.

Purchased Software

- MDM software is designed and tested to deliver performance along with functionality for contextual viewing and navigation among large hierarchies and member lists, with transactional integrity.

System Integration

Objective	Build	Buy
Designed to integrate with existing systems	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>
Designed to integrate with future systems	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Tested and certified with numerous third-party applications	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Custom Built Solution

- Required quality and automation of integration with proprietary OLAP databases and relational schemas relies on availability of internal expertise.

Purchased Software Solution

- With purchased MDM software, integration work is already done; full-fledged MDM solutions are already tested and certified for a broad range of third-party applications.

Support and Maintainability

Objective	Build	Buy
Maintenance owned by business users	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Model edits are a core function of the solution—do not drive a development cycle	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Available community of users and consulting partners	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Online help and documentation	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Training curricula and open-enrollment courses	<input type="checkbox"/>	<input checked="" type="checkbox"/>
Online support portal and dedicated technical support	<input type="checkbox"/>	<input checked="" type="checkbox"/>

Custom Built Solution

- User training and support, along with ongoing maintenance, all rely on overtaxed development resources; in addition to attendant high costs and delay in responding to changing business requirements.

Purchased Software Solution

- User-maintainable, with published help and documentation, training, and online support for ongoing customization; at the same time, broad community of users and consulting partners continually drive overall product enhancement toward common goals.

Conclusion

The "build" solution will easily approach 10 times the cost of "buy", but most likely, the "build" solution will be abandoned and replaced long before that point.

By considering only a limited set of initial requirements, "build" may appear similar in cost to "buy"; however, the vastly shortened implementation times and high success rate with "buy" will yield fast and proven ROI. Expansion of the "build" solution will require multiple full lifecycle development processes, so the implementation costs are multiplied by several phases. The mounting costs of enhancing a "build" solution for MDM is the very reason companies opt to replace their "build" approach with "buy." With the right perspective, the "build" approach should be avoided from the start.

MDM solutions must faithfully represent the unique master data model of each organization. As such, it might be tempting to apply the usual arguments for a custom development "build" approach. However, we have shown that a purchased solution that meets both core and latent MDM requirements will not pre-

clude or slow the customization of the model to unique business requirements and, in fact, will facilitate such customization by the business users themselves. Further, purchasing MDM software with the right approach and pricing model will ensure the lowest total cost with the lowest possible risk in all phases, thereby providing significant and proven ROI within a much shorter time frame.

We have also shown that, when the software vendor is focused specifically on MDM and not other BPM and ERP tools, maintenance becomes a particularly good investment as product enhancement is driven by the vendor's broad customer base, expanding partnerships and industry progression. The market-leading vendor will offer case studies that document success with proven best practices, encapsulated in the product, that demonstrate the effective methodologies of their consultants and partners.

As with the mature BPM tools and applications on the market today, buying a highly customizable MDM solution, driven by business users, is the key to getting the best value. In this case, a "buy" decision, with its lower total cost of ownership, fastest time to benefit, and a plan for perpetual improvement, is preferable to a custom development "build" approach.



ABOUT STRATATURE • Stratature, a wholly owned subsidiary of Microsoft, is a global provider of master data management software. Stratature helps organizations execute on enterprise and departmental performance management, satisfy corporate governance, and improve enterprise reporting by providing a comprehensive dimension, hierarchy and master data management solution in a single, fast-to-implement, and easy-to-maintain product.

Stratature's unique approach delivers a highly productized solution enabling large-scale collaboration, granular security, and enterprise deployment. Providing the highest functionality set and the lowest cost of ownership, Stratature presents optimal return on investment and the best price/performance of any solution available on the market.

Working through a network of technical alliances, distributors, resellers, and business consulting partners, Stratature provides its highly successful master data management solution, +EDM, around the world.

For more information, visit Stratature online at www.stratature.com.

© 2001 - 2007 Stratature, Inc. All rights reserved. No part of this publication may be copied, modified or transmitted for any purpose without the express written permission of Stratature. Stratature and +EDM, are trademarks of Stratature, Inc. All other brand and product names are the trademarks of their respective companies.

SWP245D