

Customer Data Integration: Buy vs. Build

An Oracle White Paper
October 2004

ORACLE®
—
SIEBEL
CUSTOMER RELATIONSHIP
MANAGEMENT

Customer Data Integration: Buy vs. Build

INTRODUCTION

Organizations that execute a customer data integration (CDI) strategy must decide early in the project lifecycle if they want to develop internally the enabling software or license a packaged solution from a vendor. This “buy vs. build” decision is quite important because it impacts the ultimate success of the strategy.

While internally developed solutions will always play a role in meeting IT requirements, recent surveys reflect strong preference for packaged applications. A Giga Information Group survey, for example, found that only 26 percent of the respondents planned to build internally for competitive advantage.¹ Companies are increasingly aware of the hidden costs of developing major software applications internally—the up-front design and development represent only a fraction of the entire cost of ownership of the solution over its lifetime. Custom-developed solutions that do not benefit from the rigorous software development, quality assurance, and testing practices of software vendors often require substantial subsequent investment for debugging and incremental functionality enhancement. When you factor in the probability that one-off solutions may fail to meet the necessary performance and scalability requirements when completed, the cost of internally built software infrastructure often gets out of hand, and the lessons from the field outlined at the end of this paper attest to the big risks of custom-built CDI solutions.

This paper compares the total cost of ownership (TCO) of custom-built CDI solutions to the TCO of solutions based on packaged software such as Oracle’s Siebel Universal Customer Master. In addition to the TCO arguments, this paper also examines other important factors that influence this decision, such as project risk and time to market. The paper finally points to the real-time experience and lessons learned from some companies that have custom built and implemented CDI solutions. The time and effort estimates provided in this paper were developed by Oracle’s Siebel product marketing and engineering organizations, and were cross-checked with the real-life experience of our services organization. These estimates should be treated as rule-of-thumb guidelines since every CDI project is unique.

¹ “GWITF Polling Results: CRM Implementation Indicates Preference for Customized Integrated Solutions,” Giga Information Group, June 2001.

TOTAL COST OF OWNERSHIP

The TCO of a CDI solution over its lifetime can be broken down into the following four major components:

- Design and development
- Configuration and data load
- Integration
- Ongoing maintenance and upgrade

Design and Development

Some organizations believe that they can develop a CDI solution internally at a fraction of the cost of a packaged solution. In most cases, such organizations discover later on that any up-front costs saved are incurred multiple times over the lifetime of the solution.

Building a CDI solution internally requires deep expertise in customer-facing business processes, customer data management, data quality, application architecture, scalability and performance, and integration. Although many IT departments have a cadre of very talented people, it is very hard to develop all of the above capabilities in-house without being focused exclusively on solving the CDI challenge.

Designing and developing a CDI solution requires (1) creating the application architecture, (2) developing the customer data model, and (3) creating the customer master functionality.

- Creating the application architecture is a critical task for the success of the entire CDI initiative because it determined the performance and scalability of the solution. Most IT departments will approach the task according to their level of expertise and the established corporate IT standards, for example, J2EE or .NET compliance. It is reasonable to assume that designing the application architecture will take an architect or a team of architects about 100 to 150 man-days.
- Developing the data model is one of the most challenging aspects of building a CDI solution internally. Some companies have been mired in yearlong discussions on what is the best way to represent a customer. Customer information is ubiquitous in the enterprise—it is found in virtually every IT system, but every function and IT system uses customer information in a different context, for different purposes, and therefore has a different perspective. It is not rare that five different people in five different company functions will give five different responses to the question: What are the most important customer attributes? For the vice president of marketing, the key attributes may be demographic characteristics and past purchasing behavior; for the vice president of finance, the most important attribute may be the credit risk. In the context of a large and diverse enterprise, it is very

challenging to come to a common understanding of who is a customer, what are the must-have attributes to describe a customer, and what other attributes should be stored in order to gain a true 360-degree view of the customer. Even if a company manages to agree on a set of attributes that optimally describe a customer for the entire enterprise, the technical challenge of designing a normalized data model that is optimized for fast performance is not to be underestimated. Designing the data model starts with a logical design—defining the entities and the relationships between them on the whiteboard. Then the physical design phase will translate this conceptual blueprint into physical database tables. The task will be further complicated if the customer data model needs to support multiple languages. In our estimate, the entire project of developing a customer data model for individuals and corporate accounts requires a team of about 10 people—three business analysts, one project manager, one data architect, one data modeler, one database architect, and four developers—working full time for about a year and a half (for a total of about 3,500 man-days).

- Companies that decide to build their CDI solution would have to create the following application functionality:
 - **Matching** is the ability to recognize duplicate customer records. Building a configurable match capability that can compare records on multiple attributes requires a team of about six people and a total of 150 to 200 man-days' effort.
 - **Intelligent merge** is the capability of creating a best-of-breed record from multiple source records. The automerge capability allows companies to merge records with high probability of being duplicates without pointing them to the attention of a customer data steward. Closely related is the unmerge capability that enables customer data stewards to split erroneously merged records. Custom-building these capabilities can take up to 200 to 250 man-days.
 - **Cross-referencing** maintains a record of the foreign keys under which customer records appear in the operational applications. For example, if a customer is known as GH89UI in the call center application, and customer 787-989 in a back-office application, the cross-referencing infrastructure will link these two foreign IDs. A customer master would also have to be able to generate a unique universal ID for each customer. Developing these two capabilities requires a team of three people and a total of 100 to 150 man-days.
 - **Hierarchy management** allows companies to model the hierarchies of their corporate customers. Very often companies will have to keep track of multiple hierarchies for the same customer simultaneously. For example, the sales and management hierarchy can be very different from the legal

hierarchy. Building support for hierarchy management can take a team of three people a total of 50 to 100 man-days.

- **Application authorization** is the ability to assign Create/Read/Update/Delete privileges to the operational applications with respect to the “best” customer record maintained in the customer master. This capability can be developed for about 50 man-days by a team of three people.
- **History and audit trail** enable customer data stewards to trace the history of the best-of-breed record. This capability is also the basis for some other pieces of functionality such as intelligent merge and unmerge. These capabilities require a total of 50 man-days of effort.
- **Integration to data cleansing and data enhancement services** enable the CDI solution to standardize the spelling and format of customer records and standardize the addresses to comply with postal standards. Providers of data cleansing services include Trillium, FirstLogic, Group 1 Software, and Ascential. Data enhancement services such as D&B allow improvement to be made to the quality of customer records with additional information. Integration to each data cleansing or enhancement service can take 20 to 50 man-days per service.
- **Integration framework** is the infrastructure of the customer master application that enables integration with the operational applications. In most large enterprises, the IT applications span a wide range of technologies and this integration framework should be capable of handling various modes of integration such as real-time, near-real-time, and batch mode. It should also be capable of supporting tightly coupled (programmatic) and loosely coupled integration interfaces (MQSeries, integration server). Since a CDI solution is one of the most integration-heavy projects an IT department can undertake, the quality and robustness of the integration framework is often a make-or-break factor of success. Developing such an integration framework internally can take a team of six people a total of 600 man-days.
- **User interface** is also important for a CDI solution, contrary to the belief of some IT departments and even vendors. Customer data stewards often have to make judgments about creating a best-of-breed record. They have to manually merge, unmerge, and cleanse customer records on a daily basis. This task is best achieved in a user interface in the customer master application directly. Developing a user interface from scratch can take a total of 200 to 300 man-days.

The task of custom-building a CDI solution can be daunting. Companies that decide to purchase a packaged solution can save most of the design and development costs, which may amount to as much as 5,400 man-days.

Task	Total estimated effort (in man-days)
Application Architecture	100
Data Model	3,500
Matching	150–200
Intelligent Merge, Automerge, Unmerge	200–250
Cross-Referencing, UUID	100–150
Hierarchy Management	50–100
Application Authorization	50
History and Audit Trail	50
Integration to Data Cleansing and Data Enhancement Services	100
Integration Framework	600
User Interface	200–300
Total Initial Design and Development	5,100–5,400

Configuration and Data Load

Since internally developed CDI solutions presumably reflect the precise business requirements of the company, there should not be any configuration necessary after the solution is designed and developed. In the case of a packaged software solution, the off-the-shelf software would need to be configured and customized to fit the particular business requirements. The configuration and customization of packaged solutions typically involves the following steps:

- Extend and customize the data model.
- Set up the screens and views for the customer data stewards.
- Define workflows, policies, and processes for managing customer data— from data import, to matching, deduplication, cleansing and archiving.

The work effort involved in the customization and configuration of a packaged solution will vary drastically depending on the solution itself and how widely the customer’s business requirements differ from out-of-the box functionality. It is our experience that customization and configuration of Siebel Universal Customer Master can be accomplished by a team of three people for a total effort of 100 to 150 man-days.

The load of data in the CDI solution is one of the trickiest and most time-consuming steps in the implementation process. How long it takes depends on the following factors:

- How much support does the CDI solution provide for the initial data load (that is, the quality and robustness of the integration framework developed during design and development)?
- Is all data initial cleansed as it is loaded? Respectively, how clean is the data to start with?
- How many sources is the data loaded from?
- How many customer records need to be loaded?

Siebel Universal Customer Master leverages the Enterprise Information Integration (EIM) framework to facilitate data load. Regardless of which strategy a company chooses, it will need to set up an extract-transform-load (ETL) process from each of the data sources to the customer master. In the case of the Siebel Universal Customer Master, the solution provides staging tables that facilitate the ETL processes.

On the low end (less than a million customer records, less one to five sources) the initial data load can be accomplished within less than a month or 20 to 30 man-days. On the higher end, this task can take hundreds of man-hours.

Integration

A CDI solution is often the most integration-heavy application in the IT environment. The effort necessary to integrate the customer master to the operational applications will vary widely depending on the following:

- What is the number of applications to be integrated?
- How “integration ready” are these applications? That is, do they have ready APIs that can be leveraged?
- What are the integration technologies/mechanisms used? For example, Web services, integration server, programmatic interfaces, an integration server, or packaged integration processes running on an integration server.

In most cases, integrating the customer master to a single operational application for bidirectional sync of individual and corporate account information can be accomplished within 80 man-days. Using packaged application processes such as Siebel Universal Application Network can cut this effort by up to a half.

Ongoing Maintenance and Upgrade

It is no secret that custom-built solutions start becoming more and more expensive over their lifetime. One-off solutions that have not undergone rigorous quality assurance and testing processes are often buggy, and fail to scale and perform as expected. All that means that high-quality developers must stay involved in fixing

bugs and making incremental functionality enhancements. The Gartner Group estimates that the maintenance of custom-built solutions requires about 18 percent of the initial design and development every year. In other words, custom-built solutions are paid twice over a five-year period.² For a custom-built CDI solution, companies need to reckon with potentially another 5,000+ man-days in labor cost over the first five years after the solution is put in place. This number becomes even more formidable when keeping in mind that CDI solutions tend to have very long lifetimes. They are integrated to so many operational applications that they become the lifeblood of the company, and the cost of replacement becomes very high. Customer information files in the banking industry are often 30+ years old.

For a packaged solution the ongoing maintenance and upgrade cost includes

- Software maintenance, which is usually 15 to 22 percent of the original net software license, depending on the level of support
- 15 percent p.a. of the initial implementation cost³

Putting It All Together

The table below represents an example of the financial summary for build versus buy analysis.

	Internal Build		Buy		Savings	
	man-days	\$	man-days	\$	man-days	\$
Initial Development	5,200	\$4,160,000	-	\$0	5,200	\$4,160,000
<i>Net Software License*</i>	-	\$350,000	-	\$2,500,000	-	(\$2,150,000)
<i>Configuration and Data Load</i>	500	\$400,000	1,650	\$1,980,000	(1,150)	(\$1,580,000)
Integration	600	\$480,000	400	\$480,000	-	-
Application Maintenance (5 yr)	5,670	\$5,420,300	1,538	\$3,355,000	4,133	\$2,065,300
Labor	5,670	\$5,122,800	1,538	\$1,230,000	4,133	\$3,892,800
Software (5 yrs maintenance)	-	\$297,500	-	\$2,125,000	-	(\$1,827,500)
Total	11,970	\$10,810,300	3,588	\$8,315,000	8,183	\$2,495,300

* Net software licenses, configuration, and data load estimates for a packaged solution will vary a lot depending on the vendor selected, the scope of the implementation, and so on.

² “New Wave Development: Resizing the SDLC,” Gartner Group, October 2001.

³ “Application Buyers Will Seek Vendor Independence,” Forrester Research.

Financial analysis represents projections based on a set of assumptions, and it is important that companies have sufficient confidence in the assumptions made. For example, if the internal IT department claims that they can build a CDI solution with a team of five people within six months, the company should ask itself what is the probability of this project being completed within the time frame promised, within budget, and with the key performance indicators (KPIs) needed. It would be useful, for example, to look back at its own track record of delivering custom-built solutions and analyze the plan versus the actual effort for past projects of similar scope. The same analysis should be performed for implementing a packaged solution.

In most cases a buy strategy will deliver significant financial benefits over a build strategy.

OTHER FACTORS

Companies should also include in their financial analysis other factors besides the “hard” financial projections for time and expense.

Project Risk

With every large-scale IT project companies are assuming substantial risk; that is, there is a plausible scenario under which the company may spend millions of dollars and not achieve its goals. A company that is embarking on a CDI project should try to estimate its probability of success for both build and buy strategies, and adjust the financial projections accordingly. In most cases a build strategy will carry more substantial risk—most IT departments would not have built a CDI solution before, so there will be mistakes on the way. With a buy strategy, companies can profit from the expertise of the vendor in building scalable and performant software, as well as from the expertise of a systems integrator that has created such solutions for other companies before.

Time to Market

Time to market can be another substantial consideration. The design and development cycle for an internally built solution can be expected to be yearlong, and a company that embarks on such a strategy must ask itself if it can wait that long. Even if the difference in dollar amount between putting in place a custom-built solution and a packaged one is not overwhelmingly in favor of the latter, a buy strategy typically results in a much faster time to market.

Opportunity Cost

Building a CDI solution internally will occupy a significant percentage of the resources of any IT department, and custom building a CDI solution simply means that a lot of other projects will not get done. A company that is pondering that strategy must assess its tolerance for delaying other IT projects.

EXPERIENCE FROM THE FIELD

There is nothing more powerful than looking at the experience of other enterprises in deploying CDI solutions. Below are some examples of market-leading companies that abandoned custom-built strategies after projecting the expected time and effort involved, or after investing significant amounts of money and resources.

Large Automotive Manufacturer

This automotive manufacturer developed a CDI solution internally at a cost of more than US\$90 million, only to discover that the solution did not perform as expected. The solution was abandoned and never put in production. The company is currently assessing packaged software solutions and completely outsourcing the task to a service bureau.

Leading High-Tech Company

This high-tech company scoped out an internal project to build a customer master solution. After developing the project plan, the company realized that the task would require a team of about 40 people working either part time or full time for a whole year. The total effort was estimated at 12,760 man-days. Even when 40 percent of the work could be done offshore at significantly discounted rates, the project still amounted to a multimillion-dollar figure. The company ended up deciding for a packaged customer mastering solution.

Diversified High-Tech Manufacturer

This diversified high-tech manufacturer has so far spent more than \$80 million building, deploying, and maintaining its own customer master, only to be evaluating other options now.



Customer Data Integration: Buy vs. Build
October 2004
Author: Bogomil Balkansky

Oracle Corporation
World Headquarters
500 Oracle Parkway
Redwood Shores, CA 94065
U.S.A.

Worldwide Inquiries:
Phone: +1.650.506.7000
Fax: +1.650.506.7200
oracle.com

Copyright © 2004, 2006, Oracle. All rights reserved.

This document is provided for information purposes only and the contents hereof are subject to change without notice.

This document is not warranted to be error-free, nor subject to any other warranties or conditions, whether expressed orally or implied in law, including implied warranties and conditions of merchantability or fitness for a particular purpose. We specifically disclaim any liability with respect to this document and no contractual obligations are formed either directly or indirectly by this document. This document may not be reproduced or transmitted in any form or by any means, electronic or mechanical, for any purpose, without our prior written permission. Oracle, JD Edwards, PeopleSoft, and Siebel are registered trademarks of Oracle Corporation and/or its affiliates. Other names may be trademarks of their respective owners.