Q8-4 How Do CRM, ERP, and EAI Support Enterprise Processes?

Enterprise systems like the one in Figure 8-7 were not feasible until network, data communication, and database technologies reached a sufficient level of capability and maturity in the late 1980s and early 1990s. At that point, many organizations began to develop enterprise systems.

The Need for Business Process Engineering

As they did so, organizations realized that their existing business processes needed to change. In part, they needed to change to use the shared databases and to use new computer-based forms and reports. However, an even more important reason for changing business processes was that integrated data and enterprise systems offered the potential of substantial improvements in process quality. It became possible to do things that had been impossible before. Using Porter's language (Lesson 2), enterprise systems enabled the creation of stronger, faster, more effective *linkages* among value chains.

For example, when the hospital used a paper-based system, the kitchen would prepare meals for everyone who was a patient at the hospital as of midnight the night before. It was not possible to obtain data about discharges until the next midnight. Consequently, considerable food was wasted at substantial cost.

With the enterprise system, the kitchen can be notified about patient discharges as they occur throughout the day, resulting in substantial reductions in wasted food. But when should the kitchen be notified? Immediately? And what if the discharge is cancelled before completion? Notify the kitchen of the cancelled discharge? Many possibilities and alternatives exist. So, to design its new enterprise system, the hospital needed to determine how best to change its processes to take advantage of the new capability. Such projects came to be known as business process reengineering, which is the activity of altering existing and designing new business processes to take advantage of new information systems.

Unfortunately, business process reengineering is difficult, slow, and exceedingly expensive. Business analysts need to interview key personnel throughout the organization to determine how best to use the new technology. Because of the complexity involved, such projects require high-level, expensive skills and considerable time. Many early projects stalled when the enormity of the project became apparent. This left some organizations with partially implemented systems, which had disastrous consequences. Personnel didn't know if they were using the new system, the old system, or some hacked-up version of both.

The stage was set for the emergence of enterprise application solutions, which we discuss next.

Emergence of Enterprise Application Solutions

When the process quality benefits of enterprise-wide systems became apparent, most organizations were still developing their applications in-house. At the time, organizations perceived their needs as being “too unique” to be satisfied by off-the-shelf or altered applications. However, as applications became more and more complex, in-house development costs became infeasible. As stated in Lesson 4, systems built in-house are expensive not only because of their high initial development costs, but also because of the continuing need to adapt those systems to changing requirements.

In the early 1990s, as the costs of business process reengineering were coupled to the costs of in-house development, organizations began to look more favorably on the idea of licensing preexisting applications. “Maybe we’re not so unique, after all.”

Some of the vendors who took advantage of this change in attitude were PeopleSoft, which licensed payroll and limited-capability human resources systems; Siebel, which licensed a sales lead tracking and management system; and SAP, which licensed something new, a system called *enterprise resource management*.

These three companies, and ultimately dozens of others like them, offered not just software and database designs. They also offered standardized business processes. These inherent processes, which are predesigned procedures for using the software products, saved organizations from the expense, delays, and risks of business process reengineering. Instead, organizations could license the software and obtain, as part of the deal, prebuilt processes that the vendors assured them were based on “industry best practices.”

See the Career Guide to learn more about careers in managing large-scale systems.

Some parts of that deal were too good to be true because, as you’ll learn in Q8-5, inherent processes are almost never a perfect fit. But the offer was too much for many organizations to resist. Over time, three categories of enterprise applications emerged: customer relationship management, enterprise resource planning, and enterprise application integration. Consider each.

Customer Relationship Management (CRM)

A customer relationship management (CRM) is a suite of applications, a database, and a set of inherent processes for managing all the interactions with the customer, from lead generation to customer service. Every contact and transaction with the customer is recorded in the CRM database. Vendors of CRM systems claim that using their products makes the organization *customer-centric.* Though that term reeks of sales hyperbole, it does indicate the nature and

intent of CRM packages.

Figure 8-8 shows four phases of the customer life cycle: marketing, customer acquisition, relationship management, and loss/churn. Marketing sends messages to the target market to attract customer prospects. When prospects order, they become customers who need to be supported. Additionally, relationship management processes increase the value of existing customers by selling them more product. Inevitably, over time the organization loses customers. When this occurs, win-back processes categorize customers according to value and attempt to win back high-value customers.

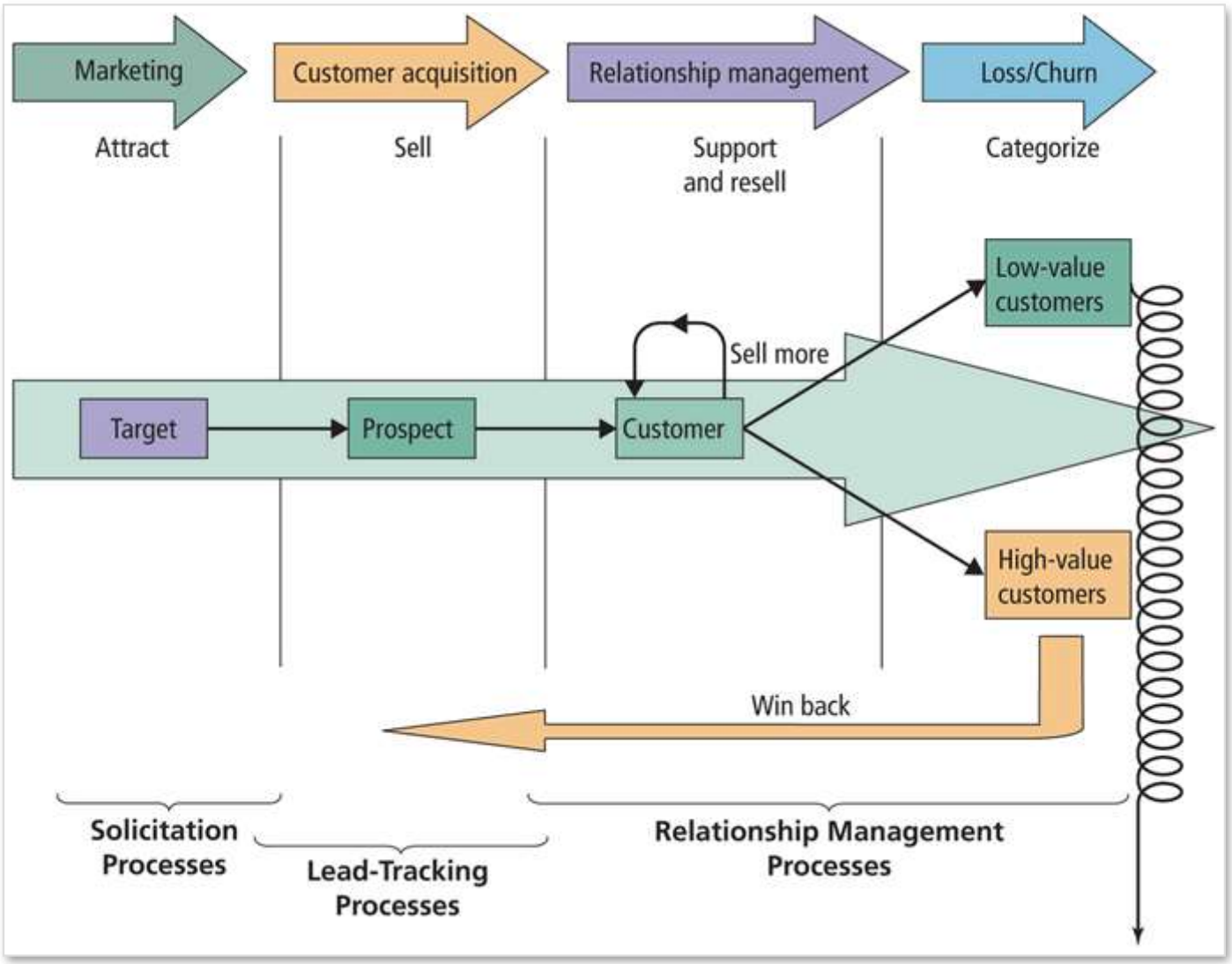

Figure 8-8: The Customer Life Cycle

Source: The Customer Life Cycle. Used with permission from Professor Douglas MacLachlan, Foster School of Business, University of Washington.

Figure 8-9 illustrates the major components of a CRM application. Notice that components exist for each stage of the customer life cycle. As shown, all applications process a common customer database. This design eliminates duplicated customer data and removes the possibility of inconsistent data. It also means that each department knows what has been happening with the customer at other departments. Customer support, for example, will know not to provide $1,000 worth of support labor to a customer that has generated $300 worth of business over time. However, it will know to bend over backward for customers that have

generated hundreds of thousands of dollars of business. The result to the customers is that they feel like they are dealing with one entity, not many.

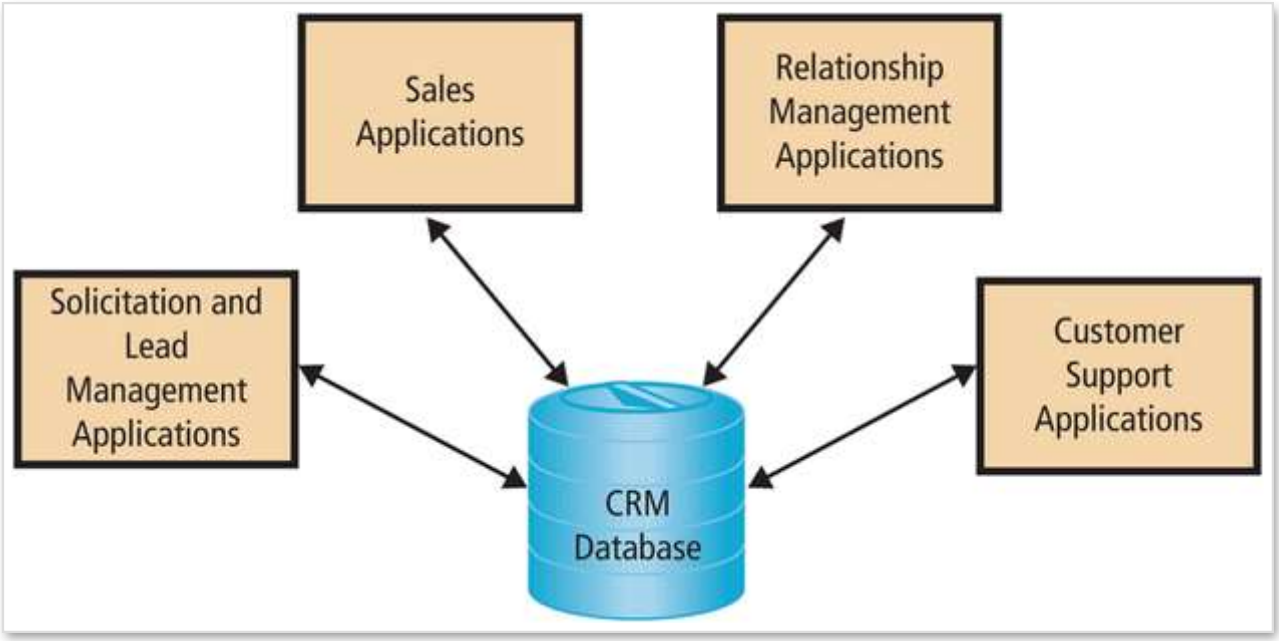

Figure 8-9: CRM Applications

CRM systems vary in the degree of functionality they provide. One of the primary tasks when selecting a CRM package is to determine the features you need and to find a package that meets that set of needs. You might be involved in just such a project during your career.

Enterprise Resource Planning (ERP)

Enterprise resource planning (ERP) is a suite of applications called modules, a database, and a set of inherent processes for consolidating business operations into a single, consistent, computing platform. An ERP system is an information system based on ERP technology. As shown in Figure 8-10, ERP systems include the functions of CRM systems but also incorporate accounting, manufacturing, inventory, and human resources applications.

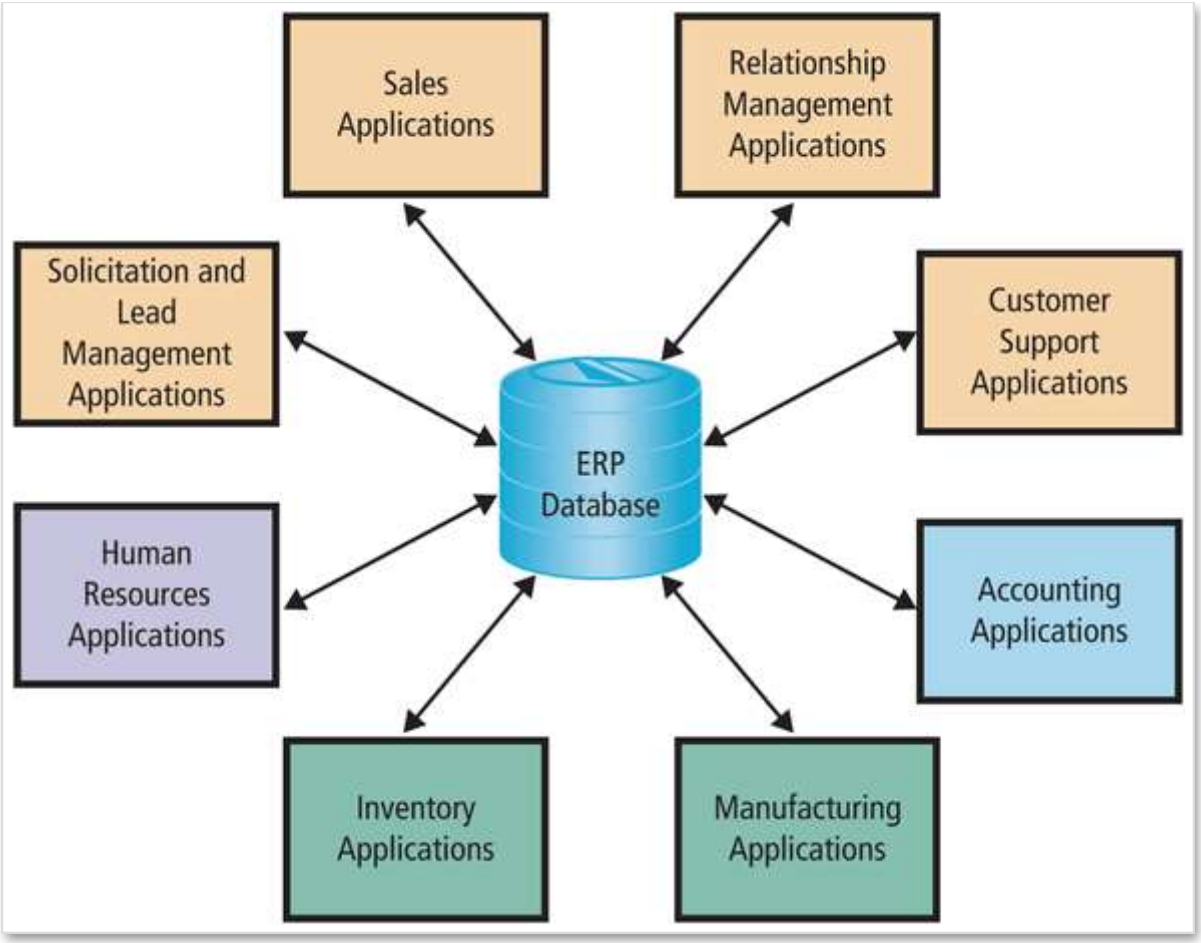

Figure 8-10: ERP Applications

Large centralized ERP systems can be attractive targets. For more information, see the Security Guide.

The primary purpose of an ERP system is integration; an ERP system allows the left hand of the organization to know what the right hand is doing. This integration allows real-time updates globally, whenever and wherever a transaction takes place. Critical business decisions can then be made on a timely basis using the latest data.

To understand the utility of this integration, consider the pre-ERP systems shown in Figure 8-11. This diagram represents the processes used by a bicycle manufacturer. It includes five different databases, one each for vendors, raw materials, finished goods, manufacturing plan, and CRM. Consider the problems that appear with such separated data when the Sales department closes a large order, say, for 1,000 bicycles.

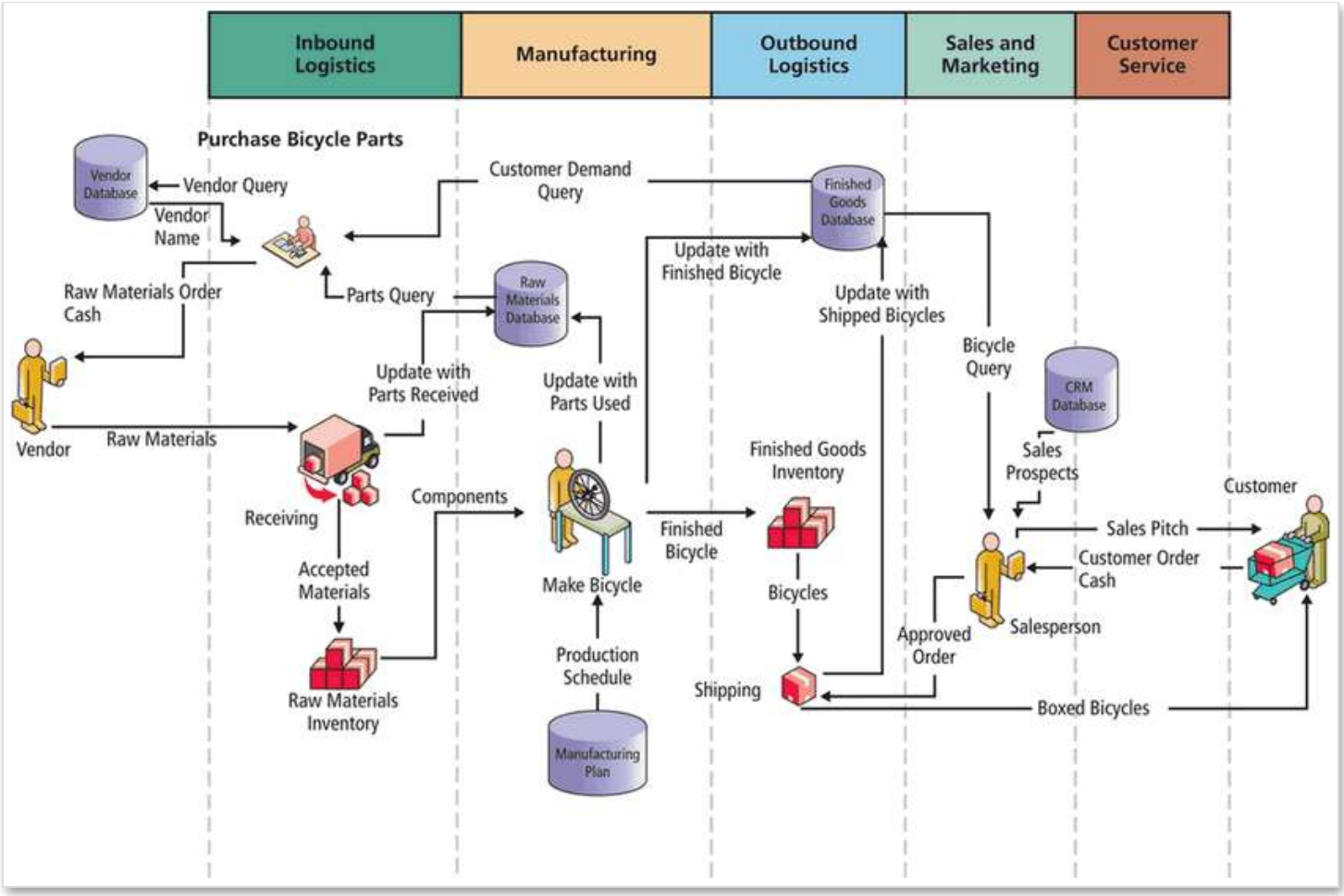

Figure 8-11: Pre-ERP Information Systems

First, should the company take the order? Can it meet the schedule requirements for such a large order? Suppose one of the primary parts vendors recently lost capacity due to an earthquake, and the manufacturer cannot obtain parts for the order in time. If so, the order schedule ought not to be approved. However, with such separated systems this situation is unknown.

Even if parts can be obtained, until the order is entered into the finished goods database, purchasing is unaware of the need to buy new parts. The same comment applies to manufacturing. Until the new order is entered into the manufacturing plan, the Production department doesn't know that it needs to increase manufacturing. And, as with parts, does the company have sufficient machine and floor capacity to fill the order on a timely basis? Does it have sufficient personnel with the correct skill sets? Should it be hiring? Can production meet the order schedule? No one knows before the order is approved.

Figure 8-11 does not show accounting. We can assume, however, that the company has a separate accounting system that is similarly isolated. Eventually, records of business activity find their way to the Accounting department and will be posted into the general ledger. With such a pre-ERP system, financial statements are always outdated, available several weeks after the close of the quarter or other accounting period.

Contrast this situation with the ERP system in Figure 8-12. Here, all activity is processed by ERP application programs (called *modules*), and consolidated data are stored in a centralized

ERP database. When Sales is confronted with the opportunity to sell 1,000 bicycles, the information it needs to confirm that the order, schedule, and terms are possible can be obtained from the ERP system immediately. Once the order is accepted, all departments, including purchasing, manufacturing, human resources, and accounting, are notified. Further, transactions are posted to the ERP database as they occur; the result is that financial statements are available quickly. In most cases, correct financial statements can be produced in real time. With such integration, ERP systems can display the current status of critical business factors to managers and executives, as shown in the sales dashboard in Figure 8-13.

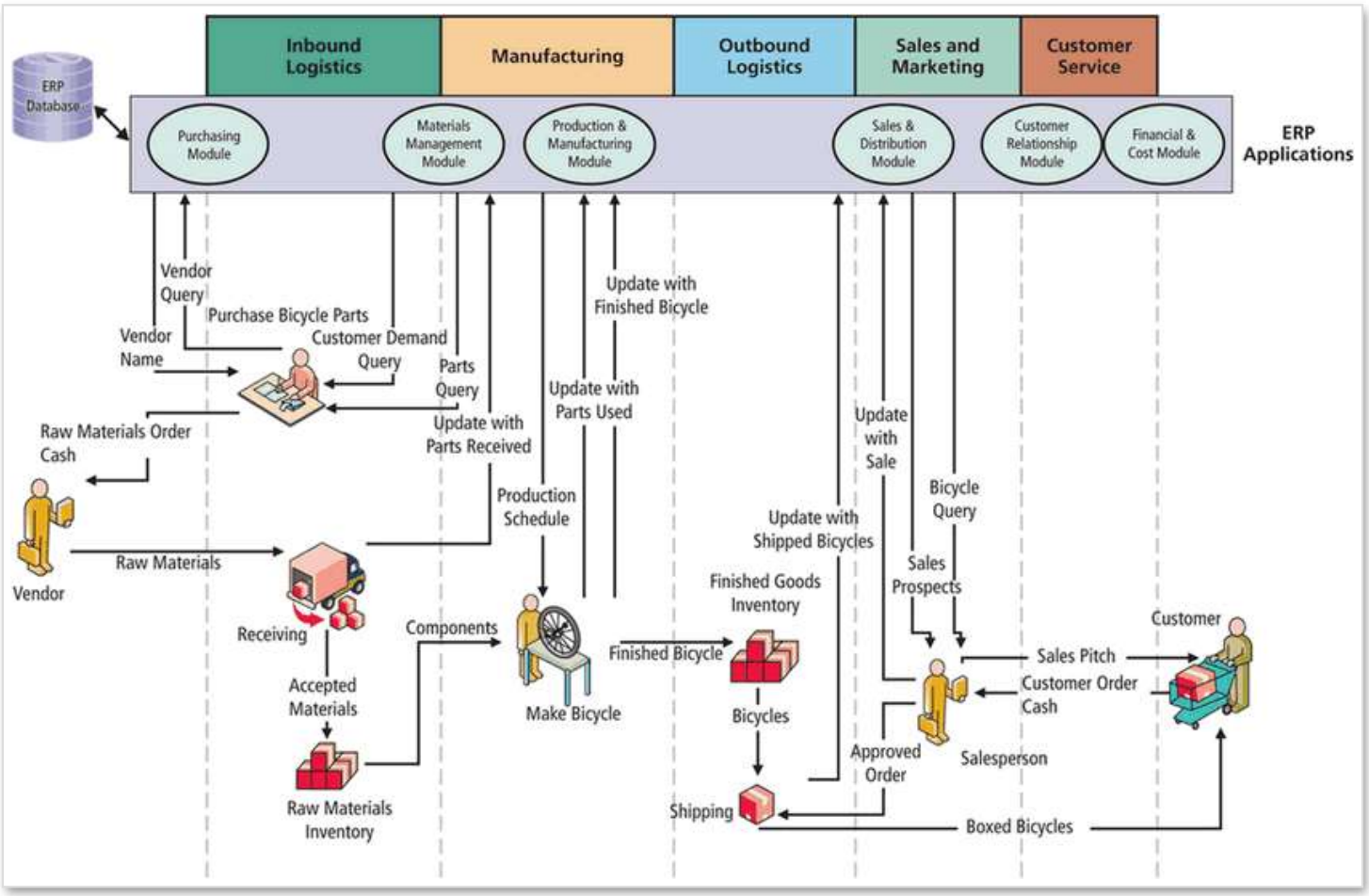

Figure 8-12: ERP Information Systems

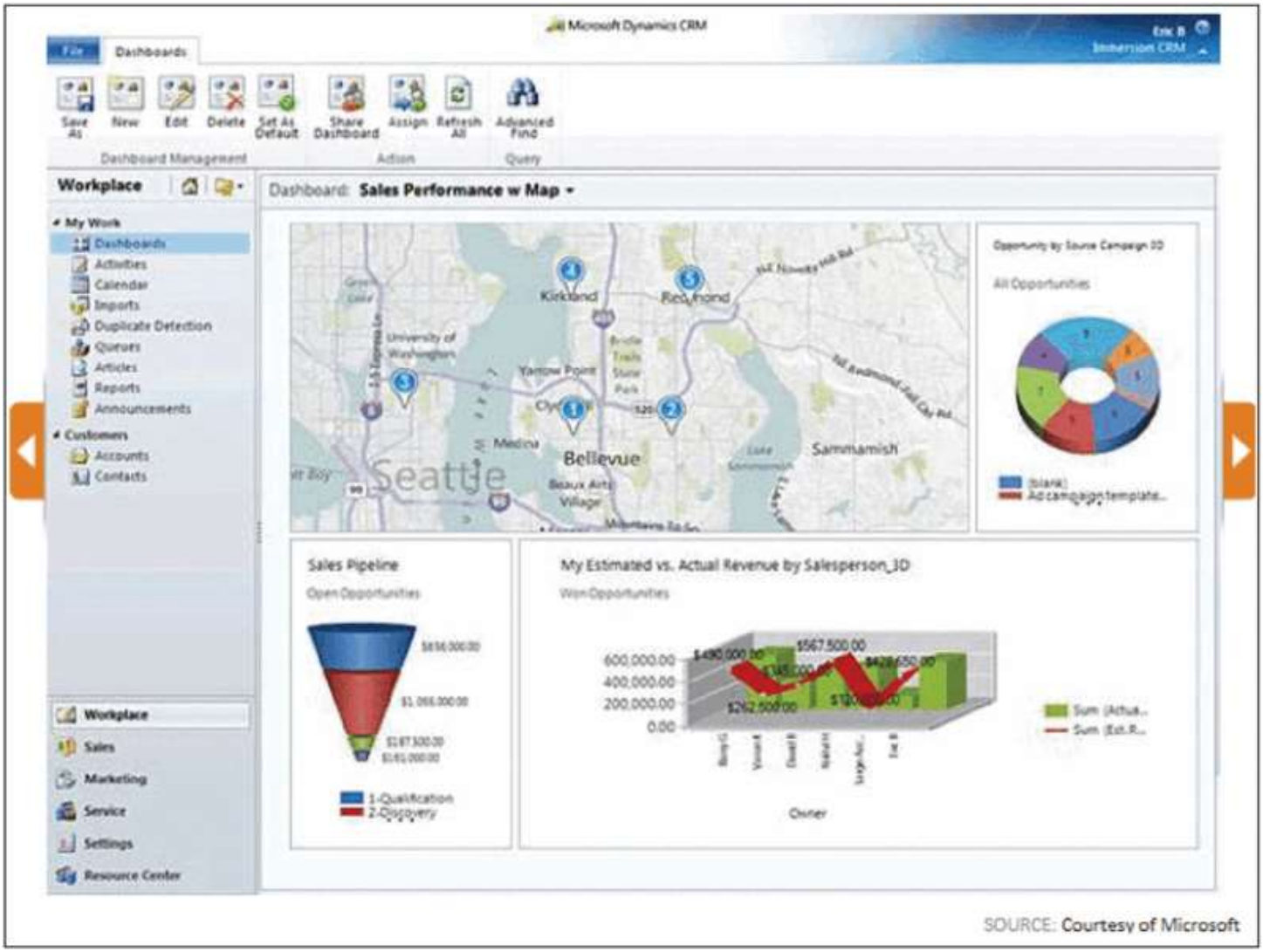

Figure 8-13: Sales Dashboard
Source: Windows 10, Microsoft Corporation.

Of course, the devil is in the details. It's one thing to draw a rectangle on a chart, label it "ERP Applications," and assume that data integration takes all the problems away. It is far more difficult to write those application programs and to design the database to store that integrated data. Even more problematic, what procedures should employees and others use to process those application programs? Specifically, for example, what actions should salespeople take before they approve a large order? Here are some of the questions that need to be answered or resolved:

- How does the Sales department determine that an order is considered large? By dollars? By volume?
- Who approves customer credit (and how)?
- Who approves production capacity (and how)?
- Who approves schedule and terms (and how)?
- What actions need to be taken if the customer modifies the order?
- How does management obtain oversight on sales activity?

As you can imagine, many other questions must be answered as well. Because of its importance to organizations today, we will discuss ERP in further detail in Q8-5. Before we do so, however, consider the third type of enterprise system: EAI.

Enterprise Application Integration (EAI)

ERP systems are not for every organization. For example, some nonmanufacturing companies find the manufacturing orientation of ERP inappropriate. Even for manufacturing companies, some find the process of converting from their current system to an ERP system too daunting. Others are quite satisfied with their manufacturing application systems and do not wish to change them.

Companies for which ERP is inappropriate still have the problems associated with information silos, however, and some choose to use enterprise application integration (EAI) to solve those problems. EAI is a suite of software applications that integrates existing systems by providing layers of software that connect applications together. EAI does the following:

- It connects system “islands” via a new layer of software/system.
- It enables existing applications to communicate and share data.
- It provides integrated information.
- It leverages existing systems—leaving functional applications as is but providing an integration layer over the top.
- It enables a gradual move to ERP.

The layers of EAI software shown in Figure 8-14 enable existing applications to communicate with each other and to share data. For example, EAI software can be configured to automatically carry out the data conversion required to make data compatible among different systems. When the CRM applications send data to the manufacturing application system, for example, the CRM system sends its data to an EAI software program. That EAI program makes the conversion and then sends the converted data to the ERP system. The reverse action is taken to send data back from the ERP to the CRM.

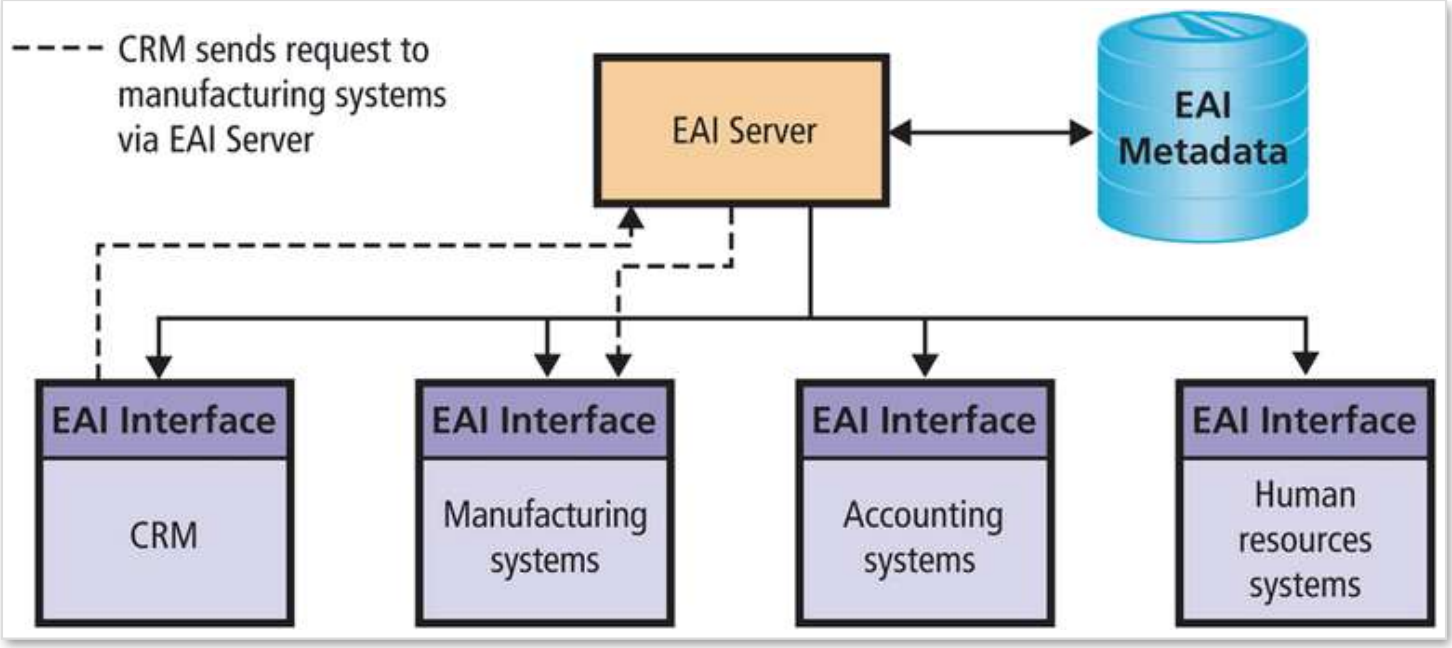

Figure 8-14: Design and Implementation for the Five Components

Although there is no centralized EAI database, the EAI software keeps files of metadata that describe data formats and locations. Users can access the EAI system to find the data they need. In some cases, the EAI system provides services that provide a “virtual integrated database” for the user to process.

The major benefit of EAI is that it enables organizations to use existing applications while eliminating many of the serious problems of isolated systems. Converting to an EAI system is not nearly as disruptive as converting to an ERP system, and it provides many of the benefits of ERP. Some organizations develop EAI applications as a stepping-stone to complete ERP systems. Today, many EAI systems use Web services standards to define the interactions among EAI components. Some or all of the processing for those components can be moved to the cloud as well.

Knowledge Check