

Agility by Design: Finished But Not Done

an eprentise white paper



tel: 407.591.4950 | toll-free: 1.888.943.5363 | web: www.eprentise.com

Author: Helene Abrams
www.eprentise.com

© 2021 eprentise, LLC. All rights reserved.

eprentise, FlexField Express and FlexField are registered trademarks of eprentise, LLC.

Oracle is a registered trademark of Oracle Corporation.

All other company or product names are used for identification only and may be trademarks of their respective owners.

Part I of this series addressed the requisites of a stable design for enterprise systems that will allow them to remain agile over both the near- and longer-term. It first identified the core building blocks of systems, the metadata, data, and business processes and explained the relationships of those building blocks and the need for complete, consistent, and correct data and standardization of business practices necessary to achieve business agility. This installment focuses on the high level steps involved in implementing an agile overall enterprise applications ecosystem.

Part 2: Finished But Not Done

Enterprise applications must continually change in order to support the ongoing business changes of the enterprise. Enabling and facilitating those changes so that they can occur with a minimum of cost and business disruption is the essence of designing for agility. The backbone of potential design agility is most often the enterprise resource planning (ERP) application, whether it is the Oracle® Business Suite, SAP, or another similar system. In larger firms, of course, the ERP systems do not run in isolation – the ERP systems are to differing degrees supplemented, complemented, and duplicated by other applications. The average Fortune 1000 company has approximately 300 different “enterprise” systems. Nevertheless, since the ERP systems contain the most central data and business processes of the company, their implementation is a focal point for design agility. Companies do not recognize that a successful implementation of their keystone enterprise application is really much more ambitious than what was envisioned when the implementation was started. Many reach a stage at which the enterprise application is up and running, the old data has been imported and new data can be entered, and they can treat the implementation as finished. What they have attained at that point is what might be termed “functional capability.” They have **finished** a vital stage of implementation, but they are not **done**.

Having functional capability brings the company to the point at which the application is configured and companies safely can move existing data and processes onto the new system and conduct business as usual. The systems work adequately, but very often the new applications mimic the legacy systems that they were meant to replace. A system that has only reached functional capability does not have common global data standards, common accounting practices, or shared and coordinated business processes. The merely functional applications support silos of the business rather than enable the enterprise as a whole to be agile and respond to changes. Knowing that the raison d’être of implementing an enterprise application is to obtain efficiency gains in a whole host of ways, the merely functional system is an incomplete success. The functional system will capture some gains that inhere in improved systems that generate economies of scale. Those same systems will also capture some gains that flow from increased efficiency in processing data and reduced expenditures for maintaining systems. The merely functional system, however, leaves the most difficult to attain and richest rewards on the table because it is incapable of providing business agility. In its unfinished state, a merely functional ERP system will not sustain a total enterprise information ecosystem that permits information to flow from its point of entry into the company’s data stream to all of the places where it will be used without cumbersome intermediate steps. In the e-world of instantaneous customer-company and company-supplier interaction via the web, immediate access to correct, complete, and consistent information is the strategy for greatly enhanced profitability and agility to make changes as the business requirements change.

Almost all of the large enterprise systems are built on relational databases. These databases are extraordinarily complex, often with thousands of tables and millions of columns that may be related to each other. As the last article in this series mentions, there are 4 types of data: seed data, configuration

data, master data, and transaction data. Because of all the relationships in an enterprise application, changes in the seed data or the configuration data impact all of the other data in the system, meaning that one small change in a configuration parameter could result in hundreds of thousands of changes in the system. When implementing an enterprise system, companies set up the parameters of the system. Initially, the system may allow changes to the setup, and users often go through several conference room pilots (tests) until the application is configured the way that they want to do business. After the system is configured and the user has begun to enter transactions, however, most of the configuration data can't be changed because it is difficult to permeate the changes throughout the rest of the database. This leaves the business in a state where the applications support the current business, but not the changes that all companies go through.

When I consulted, implementing large ERP systems for multi-national corporations with hundreds of sites, we would implement at the first site and go into production successfully. We would do the same at the second site, but by the time we finished at the third site, the company had undergone a major business change like a merger or acquisition, a divestiture, it had to comply with new statutory or regulatory requirements, or it had gone to the Euro as a functional currency. We had to go back to the first site and reimplement in order to change the configuration parameters. The reimplementation often took as long and cost as much as the initial implementation, and often, by the time we were finished again, the business had changed again. Thus, while the enterprise implementation was considered "finished," most emphatically, it was not "done" because the company was not able to recognize the benefits of their enterprise application.

As described more fully below, there is a method for moving enterprise applications from "finished" to "done." The method operates at two levels, the individual application level and what might be called the enterprise ecosystem level. Together, the method permits the company to maintain an interrelated series of mature applications that provide complete, consistent, and correct data that is efficiently delivered and maintained. The applications and system are mature. If a change is required, it should only need to be changed in a single place and from there the change would flow through the entire applications ecosystem. That maturity of each application and a similar maturing of the set of applications to efficiently support the company's business processes, in turn, permits adaptive modification that responds quickly and easily to business change – the desired level of agility.

Even a mature system has an on-going life cycle necessitated by the simple fact that the business environment is never static, which is what prompted the need for agility in the first place. For example, adding a new product line or differently regulated market segment will require adjustment of existing applications and may call for the addition of new applications. Each change of any moment triggers a cyclical process. As new applications enter the lifecycle, the organizational ecosystem may take a temporary step back in its level of maturity until the new application reaches first the "finished" stage, but is not yet "done" enough to fluidly support on-going business changes.

Method of Getting Enterprise Applications from Finished to Done

Document the Application Portfolio – Create an inventory of each of the existing applications identifying the users, major functions or business processes that the application supports, data usage in the application (including create, read, update, and delete), the business unit(s) supported by the application, costs related to support and maintenance of the application, and expected life of the

application.

Determine the Initial Configuration of Each New Application – As each application is implemented, the business rules dictate the basic parameters or configuration of that system. Those business rules might include supplier terms, customer credit limits, aging buckets, min and max reorder points for inventory, and the life of an asset. This is often the stage when companies say they are finished. The new application is ready to use to conduct day-to-day operations.

Stabilize Each Application – Define global standards for the enterprise and adjust or change all of the data of each existing application to conform to those standards. These standards might include a global chart of accounts, naming standards, pricing standards, catalog standards, compliance standards, and security standards. Standard business processes are all defined and each application is evaluated to determine the level of adherence to each of these standards. Each individual application becomes stable, and the applications taken together become consistent across the enterprise with all duplicates identified across all systems, combined and removed for all configuration data and master data.

Synthesize the Applications within the Enterprise – No enterprise application operates in a vacuum. During this stage, the applications of the enterprise work together with the information flowing seamlessly from one system to another, the data shared among applications, and redundant applications consolidated so that each business process is only supported by one application. During this stage, a master data management program is put into place so that there is a single source of data, and that data is created, updated, and deleted in one place. It is during the synthesis process that both the data and the processes become complete and correct and there is a single source of truth, a single way of doing business throughout the enterprise.

Connect the Applications to External Sources and Destinations – In order to be effective, both the enterprise's data and processes need to interact with events from outside the enterprise, whether it is accepting an order automatically from a customer, purchasing from a supplier, or complying with external statutory and reporting requirements. If there is a single source of truth for each data type within the enterprise, it is easy to communicate that information to the outside world. For example, if there is a standard product catalog, then customers can order from that catalog and be assured of product consistency whether they order from a storefront or order on-line. If inventory information is consolidated, then orders can go to suppliers on a planned demand schedule without the fear of over or under supply.

As these methods are deployed throughout the enterprise, the applications can be changed to reflect business changes. Because the applications portfolio has been rationalized and optimized, the way to effectuate most business changes (credit limits, new compliance requirements, altered currency operations, etc.) is easily identified and does not require substantial effort. The enterprise applications take on a new value because they now can be used to develop a strategy to undertake major changes like a merger or acquisition, or a divestiture. The enterprise applications, instead of being the barrier to change that they were in a "finished, but not done" stage, literally contribute to instituting strategic changes because they provide a global view of the organization and the decision makers have access to complete, consistent, and correct information. If a company is thinking about acquiring another company, it is easy to determine whether to keep the applications because they add to data and processes that don't already exist in the company's portfolio. If a decision is made to sell part of the company, if all the information is in one place, the company is able to determine the impact of the change by modeling scenarios with and without that part of the company. As an enterprise is able to use its applications to

enable business changes, those applications move from “Finished” to “Done,” a mature state that allows them to respond to business change with agility.

Curious?

For more information, please call **eprentise** at **1.888.943.5363** or visit **www.eprentise.com**.



About eprentise

eprentise provides transformation software products that allow growing companies to make their Oracle® E-Business Suite (EBS) systems agile enough to support changing business requirements, avoid a reimplementation and lower the total cost of ownership of enterprise resource planning (ERP). While enabling real-time access to complete, consistent and correct data across the enterprise, **eprentise** software is able to consolidate multiple production instances, change existing configurations such as charts of accounts and calendars, and merge, split or move sets of books, operating units, legal entities, business groups and inventory organizations.