CHAPTER 3

SOFTWARE IMPLEMENTATION METHODS

In this chapter

Using Oracle’s Application Implementation Method 36
Understanding Rapid Implementations 38
Understanding the Program Office 41
Understanding Phased Implementations 42
Case Studies Illustrating Implementation Techniques 44
In this section of the book, we discuss techniques for implementing the Oracle Applications. This chapter discusses general strategies and differences of various implementation methods. In other chapters in this section, you will read about the following topics:

- Factors affecting how much your project will cost and how much effort it will take
- Techniques for project management and control
- Planning project tasks and scope definitions
- Analysis techniques for your business and technical systems
- Designing ERP solutions for your business
- Techniques to enable the system
- Managing change, transitioning from old to new systems, and supporting users after your systems “go live”

**USING ORACLE’S APPLICATION IMPLEMENTATION METHOD**

Oracle Corporation formalized an Application Implementation Method (AIM) in 1994 to support its rapidly growing consulting organization and the first shipments of release 10 of the Applications. Prior to AIM, Oracle consultants used something called Application Implementation Plan (AIP) for release 9 implementation projects. Because AIP was really nothing more than a huge list of project tasks and loosely defined activities, AIM was a big improvement. AIM provided the Oracle consultant with an integrated set of templates, procedures, PowerPoint presentations, spreadsheets, and project plans for implementing the applications. AIM was such a success, Oracle created a subset of the templates, called it AIM Advantage, and made it available as a product to customers and other consulting firms. Since its initial release, AIM has been revised and improved several times with new templates and methods.

**AIM IS A SIX-PHASE METHOD**

Because the Oracle ERP Applications are software modules you buy from a vendor, you will use different implementation methods than the techniques you use for custom developed systems that you get from your information technology department. AIM has six major phases:

- **Definition phase**—During this phase, you plan the project, determine business objectives, and verify the feasibility of the project for given time, resource, and budget limits.

- **Operations Analysis phase**—Includes documents business requirements, gaps in the software (which can lead to customizations), and system architecture requirements. Results of the analysis should provide a proposal for future business processes, a technical architecture model, an application architecture model, workarounds for application gaps, performance testing models, and a transition strategy to migrate to the new systems. Another task that can begin in this phase is mapping of legacy data to Oracle
Using Oracle's Application Implementation Method

Application APIs or open interfaces—data conversion. If you know what data is going to be converted, and know what API it needs, then begin this task. The sooner you can start data conversion, the better!

- **Solution Design phase**—Used to create designs for solutions that meet future business requirements and processes. The design of your future organization comes alive during this phase as customizations and module configurations are finalized.

- **Build phase**—During this phase of AIM, coding and testing of customizations, enhancements, interfaces, and data conversions happens. In addition, one or more conference room pilots test the integrated enterprise system. The results of the build phase should be a working, tested business system solution.

- **Transition phase**—During this phase, the project team delivers the finished solution to the enterprise. End-user training and support, management of change, and data conversions are major activities of this phase.

- **Production phase**—Starts when the system goes live. Technical people work to stabilize and maintain the system under full transaction loads. Users and the implementation team begin a series of refinements to minimize unfavorable impacts and realize the business objectives identified in the definition phase.

**Note**

AIM tends to create large and robust project plans that are suitable for complex enterprises. Because large and robust also means costly, we discuss a modified and more reasonable version of implementation methods later in Part II, “Implementing the Oracle Applications,” of the book. The basic phases, activities, and tasks are suitable for most implementations.

**Defined Deliverables**

AIM has a defined set of project tasks, and each task has one or more deliverable results. Each deliverable document has a template with some form and content, and there is a standard style to give a consistent look and feel to the templates. If you do a good job of completing the deliverables, you will have a well-documented project library.

**Tip**

Many of the deliverable documents build on each other to produce a solution. For example, analysis documents drive design, build, transition, and production project activities. Try not to skip deliverables if they are required by other project activities. Also, you should retain these documents for future use. Organizations pay a lot of money to consulting firms to develop these deliverables, only to find the consultants are the ones who benefit, not the organization.
Advantages of AIM
AIM was a great thing when it was first released in 1994. Oracle consultants took to it like a drowning man grabs for a life preserver. Since that time, Oracle has improved the product several times, and thousands of consultants have learned basic packaged software implementation techniques. Consider using AIM to gain these advantages:

- The method is usually successful and reduces the risk of an ERP project.
- AIM is a good roadmap for inexperienced customers and consultants.
- Complex projects are supported.
- Communication among project team resources is improved through the project library, presentations, and integrated activities.
- The documentation is a good source of reference for future upgrades and customizations.
- The templates look professional.
- AIM is based on the familiar MS Office suite of products.

Disadvantages of AIM
It is hard to say much that is bad about AIM. On balance, the advantages outweigh the disadvantages. However, like any tool, you can misuse it and get less than great results. Consider the following before automatically selecting this method for your project:

- AIM produces a relatively high-cost project plan.
- Several activities are appropriate only for complex projects. It takes a skilled practitioner to know how to scale AIM down for the less-demanding situation.
- Like any tool, you must learn how to use AIM techniques. You can’t get great results if you try to skip the learning curve. Early in the project, you must learn how to use the method and approve the deliverables.
- Many AIM deliverables are required and not flexible because they are needed for downstream activities.
- Skipping or poor execution of a required deliverable can throw off dependent project activities.
- AIM does not help you find bugs in the software, diagnose problems with the software, or learn how to work with Oracle support.

Understanding Rapid Implementations
In the late 1990s as Y2K approached, customers demanded and consulting firms discovered faster ways to implement packaged software applications. The rapid implementation became
possible for certain types of customers. The events that converged in the late 1990s to provide faster implementations include the following:

- Many smaller companies couldn’t afford the big ERP project. If the software vendors and consulting firms were going to sell to the “middle market” companies, they had to develop more efficient methods.
- Many dot.coms needed a financial infrastructure; ERP applications filled the need, and rapid implementation methods provided the way.
- The functionality of the software improved a lot, many gaps were eliminated, and more companies could implement with fewer customizations.
- After the big, complex companies implemented their ERP systems, the typical implementation became less difficult.
- The number of skilled consultants and project managers increased significantly.
- Other software vendors started packaging preprogrammed integration points to the Oracle ERP modules.

**Note**

A rapid implementation of five basic financial applications using preconfigured modules and predefined business processes for 30–50 users can still take three or four months. When you add the manufacturing applications, rapid means six months.

**Description of Rapid Implementation Techniques**

Rapid implementations focus on delivering a predefined set of functionality. A key set of business processes is installed in a standard way to accelerate the implementation schedule. These projects benefit from the use of preconfigured modules and predefined business processes. You get to reuse the analysis and integration testing from other implementations, and you agree to ignore all gaps by modifying your business to fit the software. Typically, the enterprise will be allowed some control over key decisions such as the structure of the chart of accounts. Fixed budgets are set for training, production support, and data conversions (a limited amount of data).

**Note**

There is an important difference between the terms preconfigured module and vanilla implementation. In a vanilla implementation, you analyze and choose your configuration parameters to use any or all of the software functionality without customization. In the preconfigured method, you gain speed by accepting a basic configuration and predefined business processes.

**Tip**

Faster implementations usually mean lower costs. If you use an experienced team of consultants, the critical factor to the speed of your implementation is probably the ability of your organization to absorb change.
Chapter 3 Software Implementation Methods

Advantages
For certain customers, the rapid implementation of preconfigured modules and predefined business processes, has several advantages:

- The business can start to realize ERP benefits quickly; it's basically a turnkey approach.
- The Oracle customer will have to spend less staff time on project. This can be a big plus if the project team members can't get away from their regular job assignments.
- There might be fewer business disruptions during the implementation project.
- Decisions are made quickly.
- Rapidly implemented projects can cost less.

Disadvantages
When using this project method, the disadvantages often outweigh the advantages for some customers. If you are inexperienced with the Oracle software, you might be unable to predict the changes that a rapid implementation will have on important aspects of your business. Consider the following items:

- You will not have the time to customize or make interfaces to the ERP software to support your business processes the way you need them.
- If end users cannot or will not receive changes at the rapid implementation pace, your business processes might become unstable or uncoordinated.
- Certain aspects of your corporate strategy might have to change to fit the predefined processes. Consider the effects of the software on your customers, employees, and vendors. For example, are you prepared to change your product pricing formulas and salesman compensation programs? Can you visualize how your factory will run with the new software?
- Even when the implementation project ends, end users and technical people will not understand the full capabilities of the system, or how to work with Oracle Support.
- These techniques are only appropriate for small groups of users (20–80 users).
- The rapid implementation might require a core group of senior people to participate and to make decisions quickly.
- This project method might require higher up-front costs over the phased implementation.
- To implement effectively, you must have good understanding of your business processes and the software capability.
- The information technology department might not have the skills to support the production software, and possibly the hardware, when the system is ready to go live.
- There might be more business disruptions after the system goes live.
Understanding the Program Office

The program office method can be used in very complex situations to coordinate, control, and manage many parallel implementation projects. Using this method, the program office staff performs certain implementation tasks at an enterprise level and then issues requirements to the subprojects at each site. For example, the program office often creates a global chart of accounts and requires each business unit to translate and convert the various legacy general ledgers to the new corporate standard chart.

Typically, the program office is interested in the definition of all Key Flexfields, the definition of some of the Descriptive Flexfields, implementation of corporate policy, audit requirements, intercompany transactions, and the centralized business processes. The program office might also control the budget and master schedule for all the subprojects. Often, the program office allows local autonomy to the individual site projects for testing, end-user training, data conversions, interfaces, customizations, and configuration of system parameters that support local functions or never cross organization boundaries.

Also, the program office project team monitors the local activities and deliverables of the individual subprojects. The goal of this activity is to facilitate the flow of coding and knowledge among the subprojects. For example, if one site develops a customization, the program office knows which of the other subprojects have the same requirement and are interested in the work. The program office makes the customization available to the other sites through a centralized library of project deliverables.

Advantages

The advantages of the program office method come primarily in project efficiency and control. Often, the centralized project team can accomplish centralized tasks and give the results to the subprojects. Consider the following:

- The program office method can reduce total implementation costs by 20%–30%.
- The program office can consolidate buying power for enterprisewide software licenses.
- Enterprise configuration parameters are established in a uniform way.
- Many project tasks are performed only one time.

Note

Typically, these rapid implements require rework—or additional consulting expertise after the transition to production—possibly leading to higher project costs. Evaluate the rapid approach thoroughly to determine whether it’s appropriate for your organization. These implementations usually are inflexible and have limited functionality, access to legacy information, and user training.
CHAPTER 3  SOFTWARE IMPLEMENTATION METHODS

- Corporate policies and procedures can be consistently established and enforced by the new software configuration.
- The business rules for intercompany transactions are clearly defined and rationalized.
- Removes organization barriers and improves intercompany communications.
- Senior management can become involved in important aspects of the ERP software.
- Facilitation of project-related decisions might be made faster, as well as the resolution of issues.

DISADVANTAGES

The disadvantages of program office techniques are related to the top-down nature of the method and include the following items:

- Sometimes, the program office compromises on the lowest common denominator for a configuration decision. Many sites perceive a gap relative to their specific local requirements and the enterprise configuration. The subprojects spend time and money to customize the applications or work around the gap created by the program office.
- The implementation team members in subprojects often resent the controls and decisions of the centralized authority.
- Because each site has an active implementation subproject, staff requirements and costs are still very large.
- The analysis phase of the program office can take a very long time.
- Program office project team members must travel a lot.

Consider creating a program monitoring team consisting of executives from each sub-project, or subsidiary organization. This small team will help monitor progress of all projects, as well as keep executive management, the president or CEO, and the board of directors informed on progress. Having these people engaged will provide additional support to the projects for activities such as time and resource commitments, as well as supply advice and counsel to everyone involved with the projects.

UNDERSTANDING PHASED IMPLEMENTATIONS

Phased implementations seek to break up the work of an ERP implementation project. This technique can make the system more manageable and reduce risks, and costs in some cases, to the enterprise. In the mid-1990s, 4 or 5 was about the maximum number of application modules that could be launched into production at one time. If you bought 12 or 13 applications, there would be a financial phase that would be followed by phases for the distribution and manufacturing applications. As implementation techniques improved and Y2K pressures grew in the late 1990s, more and more companies started launching most of their applications at the same time. This method became known as the big-bang approach. Now, each company selects a phased or big-bang approach based on its individual requirements.
Another approach to phasing can be employed by companies with business units at multiple sites. With this technique, one business unit is used as a template, and all applications are completely implemented in an initial phase lasting 10–14 months. Then, other sites implement the applications in cookie-cutter fashion. The cookie-cutter phases are focused on end-user training and the differences that a site has from the prototype site. The cookie-cutter phase can be as short as 9–12 weeks, and these phases can be conducted at several sites simultaneously.

For your reference, we participated in an efficient project where 13 applications were implemented big bang–style in July at the Chicago site after about 8 months work. A site in Malaysia went live in October. The Ireland site started up in November. After a holiday break, the Atlanta business unit went live in February, and the final site in China started using the applications in April. Implementing thirteen application modules at five sites in four countries in sixteen months was pretty impressive.

**ADVANTAGES**

There are several advantages to the phased implementation over the all-at-once approach:

- The enterprise is in control of how much change it must absorb at one time.
- Each phase or organization success can be considered a win for the project team; therefore, management’s confidence in the team’s ability to succeed is high for subsequent phases.
- Typically, a phased implementation has less risk than the big-bang approach because problems can be isolated and complexity is reduced.
- Phases can be scheduled so new software is not going live during the busy season of the business.
- Lessons learned, project materials, and deliverables from early phases can be applied to later phases to improve performance.
- Because the user population and transaction volume grow at a slower rate, the technical infrastructure can be developed more precisely over a longer period.
- The core implementation team doesn’t have to do all the work when using the cookie-cutter approach. Power users from the first sites can assist with subsequent implementations. Key users at unimplemented sites can observe and learn about the production software from the already implemented locations.
- The big-bang approach takes more training and coordination during the transition phase as the new systems are about to be launched. With the big-bang method, it is hard to predict which groups of users will have problems with the new software and what affect those problems will have on the integrated system.

**DISADVANTAGES**

The phased implementation technique has a few disadvantages compared with the big-bang method:

- Phased implementations can cost more and take longer to complete.
- Returns on the investment in ERP software are slower to develop.
Oracle can release new software while the company is between phases. An upgrade might become required before a new phase can start.

The phased approach might require creation of temporary interfaces between the Oracle system and the legacy system. As the system is completed, these temporary interfaces cost money and are typically thrown away.

If any of the sites goes poorly, word will quickly spread, and the project team will have a credibility problem and a rough time when it starts work on the next site.

**Case Studies Illustrating Implementation Techniques**

Some practical examples from the real world might help to illustrate some of the principles and techniques of various software implementation methods. These case studies are composites from about 60 implementation projects we have observed during the past 9 years.

**Big Companies**

We have observed both extremely efficient and almost totally dysfunctional big company implementations of the ERP software. Each big company project is unique and has special needs. To meet those needs, the big company project manager can construct a customized project plan and implementation method to fit the situation. Phased implementations work well at big companies because it is hard to coordinate many complicated modules or handle the logistics of many sites. Sometimes the financial applications are implemented in the first phase followed by phases for distribution, manufacturing, and payroll applications. Sometimes all the applications are implemented at one site, and each site becomes a subsequent cookie-cutter implementation project.

Big companies often have a horrible time resolving issues and deciding on configuration parameters because there is so much money involved and each of many sites might want to control decisions about what it considers its critical success factors. For example, we once saw a large company argue for over two months about the chart of accounts structure, while eight consultants from two consulting firms tried to referee among the feuding operating units. Another large company labored for more than six months to unify a master customer list for a centralized receivables and decentralized order entry system.

Transition activities at large companies need special attention. Training end users can be a logistical challenge and can require considerable planning. For example, if you have 800 users to train and each user needs an average of three classes of two hours each and you have one month, how many classrooms and instructors do you need? Another example is that loading data from a legacy system can be a problem. If you have one million customers to load into Oracle receivables at the rate of 5,000/hour and the database administrator allows you to load 20 hours per day, you have a 10-day task. Can you imagine trying to load customers and train users on the same system during the week before going live? Now, combine those two nontrivial tasks with a hundred other transition activities.
Because they spend huge amounts of money on their ERP systems, many big companies try to optimize the systems and capture specific returns on the investment. However, sometimes companies can be incredibly insensitive and uncoordinated as they try to make money from their ERP software. For example, one business announced at the beginning of a project that the accounts payable department would be cut from 50–17 employees as soon as the system went live. Another company decided to centralize about 30 accounting sites into one shared service center and advised about 60 accountants that they would lose their jobs in about a year. Several of the 60 employees were offered positions on the ERP implementation team. Both of these projects encountered considerable delay and resistance by the affected users.

**Small Companies**

Small companies sometimes have a hard time staffing their project implementation teams. However, at the end of the project, the implementation team members become very qualified power users of the system. Often, the team members retain their primary job responsibilities during the project and can only work on the new system part-time. This part-time status typically has employees assigned to the project at 50%—in other words, 20 hours per week on the project and the other 20+ hours on their jobs. This situation doesn’t work, and the 50% that is neglected is the project.

I managed one project in which the organization had six developers assigned the project 50% and normal maintenance duties 50%. I asked the client to provide us with three developers at 100% of the time, with the other three remaining on legacy maintenance activity. The next issue we faced was helping the organization determine which three developers to choose.

Small companies have other problems when creating an implementation team. Occasionally, the small company tries to put clerical employees on the team and they have problems with issue resolution or some of the ERP concepts. In another case, one small company didn’t create the position of project manager. Each department worked on its own modules and ignored the integration points, testing, and requirements of other users. When Y2K deadlines forced the system startup, results were disastrous with a cost impact that doubled the cost of the entire project.

Project team members at small companies sometimes have a hard time relating to the cost of the implementation. We once worked with a company where the project manager (who was also the database administrator) advised me within the first hour of our meeting that he thought consulting charges of $3/minute were outrageous, and he couldn’t rationalize how we could possibly make such a contribution. We agreed a consultant could not contribute $3 in value each and every minute to his project. However, when I told him we would be able to save him $10,000/week and make the difference between success and failure, he realized we should get to work.

Because the small company might be relatively simple to implement and the technical staff might be inexperienced with the database and software, it is possible that the technical staff will be on the critical path of the project. If the database administrator can’t learn how to handle the production database by the time the users are ready to go live, you might need to
hire some temporary help to enable the users to keep to the schedule. In addition, we often see small companies with just a single database administrator who might be working 60 or more hours per week. They feel they can afford to have more DBAs as employees, but they don’t know how to establish the right ratio of support staff to user requirements. These companies can burn out a DBA quickly and then have to deal with the problem of replacing an important skill.

**GREEN FIELD COMPANIES**

The green field company is just starting operations. They literally start with a green field, build a factory, hire some workers, and start production. In these companies, a small core group of key managers start everything. Typically this group is very busy because they are involved in all aspects of constructing buildings, setting up production lines, hiring and training workers, and so forth. However, the green field company has many advantages over other ERP implementations, and these factors can easily result in an implementation that is 25% more efficient. Even if you are an established company, consider the following to learn some of the dynamics behind how an implementation project progresses:

- Everyone is new to the company. Politics and departmental boundaries are nonexistent. Everyone’s agenda is aligned toward getting the systems into production.
- Tolerance for change is high. Everything changes every week, and it is exciting, positive change. When things change, there is a sense of accomplishment. Employees expect problems, deal with them, and move on.
- Users are easy to train. They have no habits to unlearn, and they are eager to understand the systems of their new employer. User expectations of the new systems are low.
- There is no legacy system, and that means there are few data conversions and almost no interfaces. There are no spreadsheets, personal databases, and other isolated islands of computer processing on user desktops. In fact, the only other systems in the whole business might be a remote financial system at a parent company and some shop floor control systems.
- There are no favorite reports that have to be replicated in the new systems.

**PUBLIC SECTOR**

The public sector implementation project can have some unusual dynamics. These projects can be large and complex with significant numbers of users, transaction volumes, interfaces to control agencies, and requirements for server performance. Business processes are different in the public sector. Oracle provides the capability to configure the software and documentation for these customers, and some consultants specialize in the unique needs of the public sector.

In addition, the public sector project can have a different perspective from the commercially driven implementation. These projects are funded and therefore take on the characteristics of a fixed-price project and typically require an agreement not to exceed a proposed dollar amount. Also, change management techniques are important to the public sector project. Procedures and business processes can be difficult to change, and there are many issues to be resolved. Issues and configuration decisions might be addressed differently by the public
sector project team. Communication is important as everyone tries to figure out how the new relationships and processes will work. Usually many barriers exist within public sector organizations, primarily due to the silo nature of their operating units.

Many public sector organizations experience difficulty in hiring qualified resources. Public sector implementations are very different from commercial implementations because you deal with very different organization cultures and structures. Specifically within HR and Payroll, many configurations of resource structures and payment schedules exist.

Some public sector organizations have set work hours, such as 7 a.m.–3 p.m. five days per week. Many traveling consultants work 4–10 hours per day. This time in the evening when clients are not around can become very productive for a consultant. It can also become a problem because clients tend to leave at 3 p.m. whether you’re in a meeting or they are sitting at their desks. On some projects, we have had clients get up and walk out of meetings at 3 p.m., with no questions asked or next steps discussed.

Often, the legacy system is in terrible shape. The data conversions and interfaces to and from the legacy systems might be difficult or involve significant programming. We were once asked to convert a 50,000-record file that held both vendor and customer information in the same record layout because it was really just names and addresses. About 70% of the required fields by Oracle Payables and Oracle Receivables weren’t in the file. In another situation, the agency had no funding for laser printers and expected the ERP system to print reports on old daisy wheel printers. We calculated that some daily reports would not finish printing in less than 8 hours.

THE ASP
The ASP module is not for every organization and seems to have dwindled some in popularity the past year, although Gartner Group indicates the worldwide ASP market will reach $25 billion by the year 2005. There are also predictions of many, many failures in the ASP market. This can be a very costly alternative to hosting your own applications.

Giving management and control to an ASP can be a double-edged sword. Basically, you trust the ASP to follow good security, data backup, and data recovery procedures, upgrades, maintenance, and support (service-level agreements). These are all the things organizations have tried to achieve over the years—so make sure the ASP has these and more in place.

The ASP approach is a quick and easy way to get your organization up and running with Oracle Applications. The approach is a simple concept and one with which most of us are familiar. This approach is not applicable to all organizations, but it does have its benefits. Research this option well to determine whether this is attractive to your organization.

The ASP approach has its advantages, such as decreased customer maintenance, decreased hardware costs, and an overall implementation that is typically much less expensive. The opportunity to secure the Oracle Applications while forgoing significant infrastructure, software, and personnel can be very attractive to organizations. There is certainly a lower total cost of ownership that we’ve seen estimated at between 30% and 70%.
Some disadvantages are speed and security of the applications, use of preconfigured application modules, uptime guarantees, and future transition of hosting your own applications. There are many things to consider when selecting an ASP, and we can’t possibly cover them all in this chapter.

When considering an ASP approach, here are some other things to consider: service level agreements, documented disaster recovery procedures, statements of data security, scalability scenarios, and proven technical maintenance and support. Also, equally important is the financial stability of the ASP you are selecting.

**SUMMARY**

There are different techniques that you can use to implement your Oracle ERP Applications. The Application Implementation Method developed by Oracle is a good place to start when you evaluate the methods you will use. However, each technique has its strengths and weaknesses and might be more or less appropriate for your enterprise. Your best solution might be a hybrid approach based on your unique needs.

In the remainder of this section of the book, we will discuss the principal activities and deliverables associated with each major phase of an implementation project.