Oracle® Retail Enterprise

Implementation Best Practices

August 2010

Revision History

Date	Author	Revision
August 2010	Implementation Tools	Version 1.0

Note: The following is intended to outline our general product direction. It is intended for information purposes only, and may not be incorporated into any contract. It is not a commitment to deliver any material, code, or functionality, and should not be relied upon in making purchasing decisions. The development, release, and timing of any features or functionality described for Oracle's products remains at the sole discretion of Oracle.

Contents

Introduction **1**

Pre-implementation Phase **2**

Executive Sponsorship 2

Establish Measurable Business Goals 3

Staffing 4

Selecting an Implementation Partner 5

Minimize Customization 6

Rollout (Go-Live) Schedule 6

Communication / Change Management 7

Training 8

Technical Skills 8

Project Phase **10**

General Implementation Management 10

Staff Readiness 10

Hardware/Environment Planning 11

Solution Architecture 12

Requirements Gathering 12

Design 13

Code Management 14

Database 14

Installations / Environment Setup 14

Development / Custom Build 15

Development – Conversion 15

Testing 16

Test Planning 16

Conversion Testing 17

Performance Testing 17

Performance Tuning 18

Rollout 18

Post-implementation Phase **19**

Post-Rollout Support 19

Appendix: Oracle Resources **20**
Oracle Retail User Group 20
Oracle University 20
Oracle Retail Reference Model 20
Oracle Retail Documentation and Supplemental Documents 20
Oracle Retail Custom Development 21
Send Us Your Comments **22**

Introduction

This document is meant to provide clients with critical success factors from successful Oracle Retail implementations. These critical success factors capture executive-level action items to consider, as well as actual implementation-level items.

This document is intended for clients who are in the early stages of their implementation planning.

Pre-implementation Phase

Executive Sponsorship

Executive-level sponsorship is critical to a successful implementation. A large-scale software implementation requires strong and decisive leaders who are empowered to make decisions. These processes are not democratic decisions. Lack of timely decision-making does not allow progress.

Achieved	Goal
	Establish executive alignment and the structure for executive support. Clearly defining the executive team and the executive support structure streamlines decision making and the escalation process.
	Structure an effective steering committee. Retailers have found it beneficial to have membership from external parties who have experience doing similar types of projects. Bringing in resources with experience with retail and expertise with large transformation projects provides valuable input on potential issues and risks.
	Empower the core team in decision making. Allowing the core team to make decisions allows a more agile project phase. Decisions can be made more quickly by the people with direct knowledge of the situation.
	Establish a strict escalation process. Having a well-established escalation process allows issues that arise during the project phase to be resolved quickly. The core team needs a defined process to escalate to the executive sponsor and steering committee when input is required for difficult issues.
	Establish a project charter.
	Establish project charter enforcement.
	Provide effective governance.
	Select the core team carefully and judiciously. The core team should include representatives from the business and IT teams. Selecting an effective and correctly sized core team helps ensure a successful implementation. A team that is too large can lengthen decision-making time, while a team that is too small can limit the quality of decisions. Here is an example of effective team membership: One successful retailer core team structure included two teams. One team was composed of members from multiple countries to give insight into operations across all regions. This team was responsible for defining common capabilities, processes, and solutions. The other team was a group of external advisors (primarily Oracle Retail subject matter experts) who helped guide the creation of the capabilities.
	Provide timely decision-making. When the executive team is called on to make critical project decisions, it is important to provide feedback in a timely manner.

Establish Measurable Business Goals

A large-scale implementation effort should be treated as a business program, not strictly an IT-driven program. The retailer should take the project as an opportunity to review the business while renewing the applications. Installing a new information system only to support existing processes is a missed opportunity to evolve and improve.

Sponsors must define the fundamental business objectives and success measures of an effort. To track and prove success and gain momentum for the project, measurable goals need to be defined early on.

Achieved	Goal
	Define your strategic vision. Develop a clear strategic vision, and use the project as an enabling factor.
	Define your business benefits. Define business benefits to be attained for each project phase to start realizing results early in the implementation and justify cost.
	Define your business metrics. The retailer should consider business metrics (GMROI, GMROF, GMROL, markup, and others) around organization (formats, stores) or product hierarchies (fashion, sports, home) in the implementation phasing plan.
	Define and deliver an end-to-end business process. Focus on defining end-to-end business process needs, not only on detailed deliverables. This is important during contractual negotiation, because contracts tend to detail exactly the work to be delivered.
	Define your business sponsors. Processes should be owned by business sponsors (sign-off and deliverable responsibility).

Staffing

Choose some of the best people from the business to be involved in the project. These resources should be leaders on their teams who are able to influence others and sell the new solution to the teams. A core team from the retailer (five to ten users or business experts), dedicated full-time to the project, is critical to the success of the implementation. The business resources should take ownership of areas that leverage their core business competencies. For example, the core team should own new business process design, create user acceptance testing scenarios, and be the "evangelists" of the new information system in the company.

Achieved	Goal
	Establish experienced program management. Program management for the implementation should be owned by experienced project managers with similar experience in size and scope, whether a point solution or a business transformation. Enterprise implementations, especially phased implementations, are complex programs. Retailers need someone who can understand and manage critical-path tasks and provide proven strategies for managing risks and staying on track. It may be beneficial to bring in an independent third-party resource for this, if a retailer does not have resources with this type of experience in-house.
	Define roles and responsibilities between the program management, IT architect, and business analyst teams. These key resources will be involved in many cross-team project milestones and deliverables. Clear definitions of resource dependencies and critical paths for the program are needed to ensure that resources are allocated to key activities.
	Align skills to the project phase. Make sure that your implementation team has the right skills to support your team and integrator to make sure that you have sufficient coverage and experience in all critical areas of the project. Involve subject matter experts.
	Maintain staff continuity. Another important goal of implementation staffing is to maintain the team through the project. Changing key team members loses continuity and experience gained during the implementation phase and requires additional ramp-up time.
	Establish exit criteria for system integration resources. Ensure that knowledge transfer has occurred prior to the departures of system integration resources at the end of a project phase. This is critical to your ability to be self-sustaining.

Selecting an Implementation Partner

The integration partner selected is an important factor in overall implementation success. Leverage the expertise of the software vendor and implementation partner to weigh in on key implementation decisions. Projects suffer greatly when there is a lack of trust between the retailer and the implementation partners. This can lead to significant delays in decision-making and achieving project milestones.

Achieved	Goal
	Select a proven implementation partner. Select an implementation partner based on a successful implementation track record, rather than implementation cost.
	Involve subject matter experts who have specific product knowledge. If the implementation partner seems to lack key product knowledge, encourage partners to bring in subject matter experts from the software vendor to support them.
	Review the resumes of integrators to ensure the right level of experience and expertise across the team. System integrators sell projects based on collective experience. Not every resource needs to be a product expert; however, the right level of experience on the overall project team is critical. Skills and strengths of the team must align to areas of strategic importance and business benefit for the overall program.
	Verify that the implementation team has the correct experience. If the retailer does not have experience delivering large software implementations internally, bring in an external partner to manage the effort. Not all systems integrators have the experience to play this role.
	Add independent contractor subject matter experts who represent the retailer's interests as needed. An independent contractor representing the retailer's interests can be effective working alongside a systems integration partner. This should be considered for key roles such as the solution architect and technical architect.
	Utilize your implementation consultants effectively. Maximize the return on your implementation consultants by letting them focus on areas in which the retail business does not have expertise.

Minimize Customization

You should have a good understanding of your business processes and how these contribute individually to your competitive advantage. This allows the implementation project to challenge the processes that do not clearly differentiate you from your competitors. If a process is not a differentiating factor, it is better to leverage the standard functionality that Oracle Retail delivers. One of the best reasons to select a software package is that it supports leading industry practices. Take advantage of these leading practices for business processes that so not give you a competitive advantage.

Achieved	Goal
	Use the base application whenever possible. For areas that are not differentiators, use the base application. This saves time and money on the initial implementation, ongoing support and management, and long-term upgradability.
	Define business processes and use cases for any existing tasks or processes that are strategic differentiators. Clearly understand and define business processes, and use cases for any existing tasks or processes that are strategic differentiators. These may need to be different from what is available in a standard software application.
	Leverage Oracle Retail subject matter experts and best practices. Plan to use Oracle Retail functional and technical experts to supplement the implementation team. These resources are familiar with the supported applications and business processes, and they can be helpful in identifying workarounds or options that eliminate the need to customize the application.
	Avoid last-minute customization or quick fixes. Avoid customization towards the end of the project as a quick fix for issues encountered. Changes introduced late in the project cycle allow less time for testing and could have adverse effects after your go-live date.

Rollout (Go-Live) Schedule

Phasing the implementation provides many benefits. Each phase should be targeted to a specific objective and produce meaningful business results in a reasonable amount of time. Lessons learned from earlier phases can be leveraged by later phases.

Achieved	Goal
	Set the priorities of implementation stages according to the business objectives, components, and tools, as well as availability of resources.
	Schedule your rollout during a less busy time of the year. Your go-live dates should be scheduled to avoid major shopping periods. To mitigate risks of failure, schedule your rollout for a low sales period.

Achieved	Goal
	Define a strategy for your rollout. Many retailers base phasing on their business formats or stores, based on the business needs and risks of the business unit. Less-profitable business units may go live first, to reduce the risk of an initial go-live failure. There are example strategies: ▪ Phase by function—This strategy can allow benefits to be gained earlier in the process, because all work does not need to be completed before go-live. This can be more expensive over the course of the project, however, because some temporary interfaces may be required. ▪ Phase by legacy IT environment—This strategy can lower overall costs, because less integration is required. This can allow you to replace or sunset some applications ▪ Phase by department—This strategy has low risk but high cost, because all work needs to be done from a work perspective and phased from an impact to the business ▪ Phase by location—This strategy is a possibility for store and warehouse implementations
	Factor for Patching/Upgrades Depending on the overall timeframe and the number of project phases, consider leaving time in the schedule to bring the production environment up to the current patch level or upgrade to a newer release of the application as part of the phasing process.

Communication / Change Management

Achieved	Goal
	Create a communication plan. Good communication is critical to a successful implementation project. Leadership needs to articulate a clear view of the big picture across project teams and to the entire company. A communication plan should be defined and included in change management.
	Make sure that the business takes ownership of the implementation. The most effective change management comes from programs in which retailers dedicate resources to the project. These resources are involved in process designs, design decisions, defining reporting needs, and identifying test cases. They are able to communicate effectively back to the business what was coming, how people's jobs would change, and what benefits they could expect.
	Set up weekly meetings. Weekly meetings of business analysts with selective end business users (a mix of resources from Logistics, Purchasing, Product/Brand, Marketing, Finance) facilitate change management, acceptance, and faith in the new application among a wider audience. They include a prospective resource consisting of user acceptance testers, help desk resources, and end user trainers.

Achieved	Goal
	Define and communicate implementation targets and milestones. Targets that cannot be moved must be defined early and communicated across all projects. Examples of business changes requiring significant planning lead time might include the following: changing accounting method by the beginning of the next financial year, opening a new central data center within 12 months, going live with a new supplier portal, or a new finance system. Understanding these types of milestones is helpful to the decision-making and scope management processes as well.

Training

Achieved	Goal
	Provide training environments for weeks or months before users are live on the system. The bulk of user training should occur before go-live, rather than early in the project. A quick initial training is helpful to obtain buy-in, but users forget details before the actual implementation. More in-depth training is needed later on.
	Leverage vendor-led training for initial core team training. Training should not be entirely focused on the system; it should allow users to see how they will use new business processes to achieve improved long-term results. Training should not be limited to live vendor classes.
	Utilize a train the trainer approach for end user training. Ideally, the business personnel assigned to the team will become the lead trainers and superusers of the applications. This typically involves a core team of business analysts who attend early training and later lead the training for end users. This is often called the "train the trainer" approach.

Technical Skills

Achieved	Goal
	Plan to supplement your existing IT teams with Oracle Retail technical experience. Although your database administrators may have general Oracle database experience, they cannot perform effectively in an Oracle Retail project without significant ramp-up.
	Plan an appropriate amount of time for the technical support team to get up to speed on your Oracle Retail applications. It is best to plan on 6 to 12 months for the learning curve for the technical support team. During this time, you need to budget to supplement the team with Oracle Retail consulting and system integration skills.
	Establish joint system integrator and client partnership on the team. The team learns environment setup and maintenance by doing it. For long-term project support success, involve client team members from the start with support from skilled partners.

Achieved	Goal
	Select qualified, experienced technical team leads. As with the core business team, select technical team leads who are engaged in learning a new technology footprint.
	Supplement the team through critical phases Plan for assistance through a stabilization period after go-live. This time will help ensure that you have mature support processes in place for assessing and assigning priorities to issues, applying and testing software patches, promoting changes into the production environment, and so on.

Project Phase

General Implementation Management

All phases of the implementation project should be managed using a generally accepted project management approach or software lifecycle implementation plan or approach. Program management for the implementation should be owned by experienced project managers, rather than mainstream business managers.

A clear project scope should be identified from the beginning, with processes in place to control changes to scope, costs, and maintainability. While there will be changes to scope as the project progresses, there must be a clear process to justify and evaluate all potential changes.

Achieved	Goal
	Create checklists for each project phase to encompass all activities and deliverables required for the phase.
	Implement and enforce a project-wide issue tracking and escalation solution.
	Measure, monitor, and track the implementation. Resource time should be tracked against project tasks to allow management to maintain an accurate view on deliverable progress and timelines. This is a program management best practice.
	Provide regular checkpoints in the process to assess and manage implementation risks.
	Communicate about the risks, and build mitigation and contingency plans. Plan for disaster recovery plans across multiple systems (both new and legacy applications).
	Create and implement a contingency resource plan to counter attrition of key resources.
	Define your go-live and cutover processes. Go-live and cutover processes should be documented and agreed to long before go-live is scheduled.

Staff Readiness

Maximize employee preparedness for the system by leveraging a complete training plan.

Achieved	Goal
	Provide a comprehensive orientation and training guide to new resources on the project to get them productive quickly.
	Provide resources such as team knowledge repositories and wikis to facilitate knowledge transfer within the team.
	Leverage informal sessions to initiate team discussions for core changes to business processes, and new functionality and technology.

Hardware/Environment Planning

Oracle Retail Enterprise solutions are complex and generally require a complex hardware solution as well. Hardware requirements and vision should be addressed early in any project.

Achieved	Goal
	Define environment management strategy early on.
	Define a patch and hot fix uptake strategy.
	Establish good processes for managing code (version control) and promoting code between environments.
	Budget an appropriate number of environments for your project. Clients need to budget about five to seven environments through all courses of the project. Needs shift during different phases of the project, but there will likely be a continuous need for this many environments.
	Define your implementation environments. A number of dedicated environments are typically needed for an Oracle Retail implementation. In addition to separate needs for activities such as testing and training, multiple environments are commonly needed for development efforts. For example, it is not feasible to develop conversion programs in an environment that is being used for other development, because of dropped tables and the likelihood of converted data corrupting other development efforts. The different environments commonly needed for an implementation include the following: ▪ Gold/base product ▪ Development—conversion ▪ Development—other builds ▪ Training ▪ Testing ▪ Performance ▪ Production
	Plan to set up and maintain a testing environment indefinitely. Maintain development and test environments on an ongoing basis to ensure the ability to support patch management and upgrade processes over the life of the software.
	Plan for a copy of your production environment. After rollout, a copy of your production environment is critical for research, testing code fixes, re-creating issues, and estimating approximate run times.

Solution Architecture

Achieved	Goal
	Current footprint The solution architecture of the current footprint should be documented.
	Align the solution future state to business processes. A phased visualization of the solution from start to finish is critical to getting buy-in from all teams. For the future vision, the business processes mentioned in the document must align to the solution architecture.

Requirements Gathering

Achieved	Goal
	Hold requirements workshops. Hold requirement workshops on individual functional areas (planning, merchandising, pricing, inventory management, point of sale, finance, reporting, and identity management), as well as other areas such as security, performance, sizing, archiving, user interface, and authentication.
	Map business processes to identify levels of detail and functional activities. Perform as-is and to-be process analysis, and identify relevant process gaps to be filled by the implementation. Receive approval from all business teams to support the gaps and relevant changes in processes.
	Identify reporting needs early in the implementation. Especially at an executive level, Business Intelligence (BI) is the system. Some BI should be delivered with each phase of the project. Do not rely on an ability to use canned reports.
	Create relevant business use cases. Identify and relate all business use cases to relevant business process levels and requirements. For a business scenario, identify actors, series of steps, prerequisites, post conditions, related business requirements and use cases. Business use cases are referenced by system designs, system integration, and user acceptance testing scripts.
	Maintain a requirements traceability matrix. Maintain a traceability matrix to relate requirements to business use cases, designs, test cases, and other project deliverables. The document helps in tracing and relating project deliverables to business requirements.
	Perform a required proof of concept to support proposed business processes and use cases.
	Perform secondary vendor proofs of concept. Proofs of concept should be performed during identification of secondary vendors for supportive business solutions (electronic funds transfer point-of-sale, point-of-sale peripherals, handhelds for inventory management, authentication cards, and so on). This helps to minimize risks of solution incompatibility or failure during design and development phases.

Achieved	Goal
	Perform updates to business requirements before approval. Updates to business requirements should not be performed after approval.
	Implement a strict change requirement process. Additional new requirements or changes in existing requirements during the implementation should be phased separately or added through a strict change requirement process to minimize disruption to design, build, and test phases, and to minimize scope creep.
	Requirements Sign off Timely sign-off on requirements needs to be enforced from the start. Delays in requirements sign-off prevent progress to the next phase. This causes bottlenecks and possible impact to the project schedule.

Design

Achieved	Goal
	Define and adopt common design processes and standards.
	Define a central repository for all design documents. Design documents should be kept in a central repository available to all. This is especially important throughout the project lifecycle, because issues are found and revisions are required.
	Clearly identify owners for all designs.
	Include stakeholders in the design development process. Design walkthroughs and review processes should include all stakeholders.
	Design and use an approval process when updating designs. If a relevant design update is required, the design should go through an update process with relevant approvals. This helps to limit updates to relevant changes, and it lowers unforeseen deviations of designs from other project documentation.
	Keep designs up to date. Keep designs current with updates up to and even after go-live, because the designs will be a valuable reference to subsequent implementation phases or support.
	Hold system configuration workshops. Hold system configuration workshops to capture approvals on system parameters.
	Perform required proofs of concept to support designs.
	Maintain requirement traceability to ensure that you capture all business requirements in designs.

Code Management

Achieved	Goal
	Use a version control product. Evaluate available version control options and select a product that meets your needs. Some products used by clients include PVCS Source Control, PVCS Dimensions, and Microsoft Version Manager.

Database

Achieved	Goal
	Define your database backup strategy.
	Set up your database analysis. Ensure that database analysis is set up and addressed on an ongoing basis. As new functionality is brought live on the system, additional tables will be populated and the analyzer needs to be run.
	Assess your partitioning strategy. After installation, assess the partitioning strategy based on anticipated volumes and hardware. **Note:** The installer will install a generic partitioning strategy not suitable for production systems, if partitioning has not been completed as specified by the installation guide.

Installations / Environment Setup

Achieved	Goal
	Use the product documentation. Take advantage of the installation guide (including appendixes) and My Oracle Support documents. **Note:** Any text that is in brackets [] in the installer fields is required input. The installers do not always validate that a value has been entered.
	Familiarize yourself with the installation order before you start. Installing applications in the correct order is essential to a successful implementation.
	Ensure that all prerequisites have been addressed before you start. All installation guides contain information on your hardware and software prerequisites.
	Perform a backup before you start.
	Take your time during setup to get it right. If you doubt that something is set up correctly, it is easier to start over and follow the steps carefully than to continue the installation and troubleshoot later. **Note:** You can continue a database installation or patch if it fails. The installer tracks which scripts have already been run and skips them when the process is rerun.

Achieved	Goal
	Obtain support from subject matter experts. Consider obtaining support from Oracle Retail subject matter experts to support the installation process. Oracle Retail also provides preconfigured offerings for some solution areas.

Development / Custom Build

When customizations are absolutely required for strategic differentiation, the following practices are recommended.

Achieved	Goal
	Design custom modifications with long-term maintenance in mind, to ease upgrades and patch application. ▪ Use a "bolt-on" noninvasive strategy. ▪ Use strict naming conventions to identify custom code.
	Create customizations that conform as closely as possible to Oracle Retail design and development standards. Conforming to Oracle Retail design and development standards makes patching and upgrading easier. Conforming to Oracle Retail design and development standards means that the user interface should use a similar pattern, consistent error messaging, and so on.
	Create customizations that follow the design process.
	Document all modules affected by customizations, for cross-reference when analyzing or applying patches.
	Select custom build developers who are familiar with the application. To modify existing functionality or add new functionality to the product, custom build developers should have experience with the application .

Development – Conversion

Achieved	Goal
	Create a data conversion strategy. Data conversion needs to be addressed early in the project and should not be underestimated.
	Leverage accelerators and prebuilt tools.
	Develop a plan that includes an inventory of every table in the database, with a method identified to populate the data in each.
	Identify an owner for each required data set.

Achieved	Goal
	Create a design document for each data set required. The design documents should include the source of data, mapping of data elements, all transformations required, and any dependencies.
	Set up a dedicated environment to capture master static data. The online screens of the application must be available to this environment, because smaller tables can be keyed online. Online validation ensures higher data integrity; loading the master data exclusively through scripts introduces more risk to the data. ▪ Each table should be populated with data that is not likely to change between the beginning of the conversion effort and the go-live. ▪ The identified data owner should validate and approve the data entered, whether the data is keyed or loaded. ▪ After the data is approved, no one should have access to this data. There should be an owner of this master data who makes and approves changes to this data. ▪ Row counts should be recorded and tracked for all data in this environment.

Testing

You must prepare to fully test all critical business scenarios. Do not rely on the software provider to have covered all client-specific business scenarios in testing. Often clients leverage specific enhancements in ways that are unique to their business processes.

Test Planning

Achieved	Goal
	Create an independent test team. The test team should be independent, using business process flows and detailed business use cases and procedures as input for test planning. Emphasis should be on testing business scenarios or the future state of the business process.
	Allow extra time in your timeline for setbacks and fixes. Ensure that the testing timeline allows time to deal with setbacks, and for software fixes from the vendor.
	Follow review and approval processes for test scripts. Follow review and approval processes for test scripts, and involve business analysts in the test planning and review effort.
	Create separate test scripts for individual functionalities and end-to-end processes. It is always recommended to have separate test scripts for individual functionalities and end-to-end processes. Individual functionality test scripts should be highly detailed to test each and every aspect of the functionality.
	Create controlled end-to-end testing cycles.

Conversion Testing

Achieved	Goal
	Perform a full functional test before introducing conversion or interfaced data. Testing before you introduce conversion or interfaced data ensures that all testing can be done successfully with data that originated in the application (that was created from the online forms). After this testing, repeat all tests with conversion or interface data. This allows differentiation between conversion or interface errors and other application errors.
	Test using converted data early in the overall testing schedule. After the functional test is complete, start introducing tests with converted data to start uncovering issues.
	Data Validation Especially for data-intensive applications, retailers need to build data validation before delivering the data. This is especially critical when recurrent weekly data is required. A clear data validation plan should be in place to mitigate late discovery of data issues that can cause project delays.
	Plan for testing and validation of converted and integration data throughout its lifecycle. Converted data should be tested and validated through its entire lifecycle. For example, a converted supplier should be: 1. Added to an item 2. Used on a cost change 3. Used on a return to vendor 4. Used on a purchase order 5. Received on a purchase order 6. Invoiced and matched in ReIM This should be done even if these processes are not part of the implementation phase. It is possible to detect conversion issues that affect future functionality, and you can correct these defects before go-live.

Performance Testing

Achieved	Goal
	Performance-test the applications based on expected business volumes. Perform performance tests on the applications based on your expected business volumes, regardless of whether a benchmark exists. Benchmarks do not address client-specific hardware constraints and integration scenarios.
	Plan to performance-test any custom modifications.

Achieved	Goal
	Plan performance testing after testing of fully converted data. Performance testing is an important part of conversion testing and batch testing. It is not possible to successfully performance-test until full volumes of data are available; therefore you should make sure that the timing of performance testing is planned after heavy testing of fully converted data by the users. Performance testing requires about three months. You do not want untested converted data for performance testing, because you can introduce conversion issues into performance testing.

Performance Tuning

Achieved	Goal
	Define and implement a performance tuning strategy. Performance tuning is always needed at the client site to make the system perform optimally. Performance tuning should be ongoing as new functionality is enabled.

Rollout

Achieved	Goal
	Prepare your organization for go-live. Attempt to minimize the volume of questions at go-live with preparation such as user training, FAQ documents, and knowledge repositories. Plan to have business resources available to answer questions on go-live. Retailers should be prepared for a flood of questions from users, no matter how small the functional impact.
	Create an implementation plan. When creating the implementation plan, think about the impact on the resources involved. For instance, try to convert all departments for a resource at the same time, so that they do not have to split time between systems. Asking a person to live in both the old and new worlds for a period of time constantly reinforces what they do not like about change, and it makes user adoption more difficult.
	Create a back-out plan. Plan for contingencies like a no-go decision, and have a back-out plan in place.
	Prepare all affected locations for go-live. When functions impact users in multiple locations, make sure that business users are represented across multiple locations. Try to involve users from future locations, so that they can start to think about how the change will affect their location. A rollout team should include people who worked on the last site's implementation, some from the current site, and some from a future site.

Post-implementation Phase

Post-Rollout Support

Achieved	Goal
	Plan your post-rollout support.
	Create a support help desk and appropriate Service Level Agreements (SLAs) to resolve business and technical issues.
	Plan to engage support personnel early. Support staff should be included in the earliest part of the development process, to develop more experience with the applications. This allows the support staff to be self-sufficient and prepared when the integrator leaves.
	Plan for business analysts to be part of your help desk. Business analysts at your help desk can support end user business issues and concerns in production.

Appendix: Oracle Resources

Oracle Retail User Group

The Oracle Retail User Group holds monthly calls to discuss best practices and implementation strategies with other retailers, and annual meetings to discuss and vote on enhancement requests and to preview and influence the Oracle Retail product roadmap.

Oracle University

Oracle University offers superior training tailored for Oracle Retail products, focused on enhancing the adoption of Oracle technologies.

Oracle Retail Reference Model

The Oracle Retail Industry Process Model was developed based on input and experience with a broad range of Oracle Retail customers.

- Aligned with Oracle applications
- Built on business process and industry standards
- Developed with Oracle Business Process Architect
- Available to customers on My Oracle Support

Oracle Retail Documentation and Supplemental Documents

Oracle Retail provides a complete library of generally available (GA) documents, as well as many supplemental support documents, that provide valuable information for Oracle Retail implementations.

The documentation for each product is packaged with the application code and is also available on the Oracle Technology Network at the following URL:

http://www.oracle.com/technetwork/documentation/oracle-retail-100266.html

Supplemental documents are available to customers at My Oracle Support:

https://support.oracle.com

The following supplemental documents at My Oracle Support are guides to find and use Oracle Retail documentation:

- Quick Start Guide to Oracle Retail Documentation (Article ID 1145078.1)

  This article describes how to find both GA and supplemental documents, including Data Model documents and documentation about required application infrastructure and compatibility.

- Index of Noteworthy Oracle Retail Knowledge Base Articles (Article ID 989300.1)

  This article is a compilation of some of the noteworthy supplemental documents that can provide additional help with your implementation. Examples are:

  – Data conversion planning guides
  – Installation test cases
  – High availability case studies

Oracle Retail Custom Development

Oracle Retail also offers customization development services.

Send Us Your Comments

Oracle welcomes customers' comments and suggestions on the quality and usefulness of this document.

Your feedback is important and helps us to best meet your needs as a user of our products. For example:

- Are the implementation steps correct and complete?
- Did you understand the context of the information?
- Did you find any errors in the information?
- Does the structure of the information help you with your tasks?
- Do you need different information or graphics? If so, where, and in what format?
- Are the examples correct? Do you need more examples?

This document was created by the Implementation Tools team. The Implementation Tools team is a solution unit within the Oracle Retail Research & Development organization focused on reducing Total Cost of Ownership and Total Cost of Implementation across the Oracle Retail Enterprise Suite. The Implementation Tools team fosters ongoing partnership and collaboration across Oracle Retail, integration partners, and clients to create tools/utilities/processes to improve the ease of Oracle Retail implementations.

If you have feedback/suggestions on how to improve this document, please contact the Implementation Tools team: RGBU-Implementation-Tools@oracle.com.

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 © 2010, 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.