



Solution Guide for
Replatforming Line-of-Business
Applications from UNIX to Windows

SAP • PeopleSoft • Siebel

Version 1.0

The information in this document and any document referenced herein is provided for informational purposes only, is provided AS IS AND WITH ALL FAULTS and cannot be understood as substituting for customized service and information that might be developed by Microsoft Corporation for a particular user based upon that user's particular environment. RELIANCE UPON THIS DOCUMENT AND ANY DOCUMENT REFERENCED HEREIN IS AT THE USER'S OWN RISK.

MICROSOFT CORPORATION PROVIDES NO WARRANTIES, EXPRESS, IMPLIED OR STATUTORY, AS TO THE INFORMATION CONTAINED IN THIS DOCUMENT AND ANY DOCUMENT REFERENCED HEREIN. Microsoft Corporation provides no warranty and makes no representation that the information provided is in this document or any document referenced herein is suitable or appropriate for any situation, and Microsoft Corporation cannot be held liable for any claim or damage of any kind that users of this document or any document referenced herein may suffer. Your retention of and/or use of this document and/or any document referenced herein constitutes your acceptance of these terms and conditions. If you do not accept these terms and conditions, Microsoft Corporation does not provide you with any right to use any part of this document or any document referenced herein.

Complying with the applicable copyright laws is the responsibility of the user. Without limiting the rights under copyright, no part of this document may be reproduced, stored in or introduced into a retrieval system, or transmitted in any form or by any means (electronic, mechanical, photocopying, recording or otherwise), or for any purpose, without the express written permission of Microsoft Corporation.

Microsoft may have patents, patent applications, trademarks, copyrights or other intellectual property rights covering subject matter within this document. Except as provided in any separate written license agreement from Microsoft, the furnishing of this document does not give you, the user, any license to these patents, trademarks, copyrights or other intellectual property.

© 2004 Microsoft Corporation. All rights reserved.

Microsoft, Active Directory, BizTalk, Windows, Windows Server, and Windows Server System are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

The names of actual companies and products mentioned herein may be the trademarks of their respective owners.

Table of Contents

Preface	5
Overview	5
Intended Audience	5
Knowledge Prerequisites for Understanding This Guide	5
How to Use This Solution Guide	6
Organization by Chapter	7
Contributors	7
Chapter 1: Envisioning Your Replatform Solution	9
Introduction and Goals	9
Creating the Vision	10
Understanding the Business Problem/Opportunity	10
Defining Your Business Goals	12
Defining High-Level Requirements	12
Defining the User Profiles	16
Setting Up the Replatform Project Team	17
The Change Control Board	19
Engaging Partners or Consultants	20
Assessing the Current Situation	21
Creating a Baseline Inventory	21
Performing the Initial Assessment and Analysis	23
Creating the Solution Concept	28
The Solution Concept and LOB System Layers	29
The Timing Approach	31
Project Success Criteria	31
Defining Scope	32
Defining the Replatforming Project Structure	33
Project Communications	33
Documentation Standards	34
Project Change Control	34
Assessing and Managing Risk	34
Closing the Envisioning Phase	36
Key Deliverables for the Envisioning Phase	36
Major Milestone: Vision/Scope Approved	36
Chapter 2: Planning Your Replatform Project	37
Introduction and Goals	37
Developing the Solution Design and Architecture	38
Conceptual Design	38
Logical Design	39
Physical Design	43
Validating the Technology	45
Creating the Functional Specification for the Solution	46
Creating the Master Project Plan	47
The Budget Plan	47
The Development Plan	47
The Test Plan	48
The Deployment Plan	49
The Training Plan	53
The Operations and Support Plan	54
Project Schedule	55
Setting Up the Development and Test Environments	56
Hardware	57
Software and Tools	57
Testing the Environments	58
Closing the Planning Phase	58
Key Deliverables Checklist for the Planning Phase	58
Major Milestone: Project Plans Approved	59
Chapter 3: Developing Your Replatform Solution	61

Introduction and Goals.....	61
Starting the Developing Phase	62
Addressing Risk in the Developing Phase	62
Building a Proof of Concept	63
Developing the Solution Components and Building the Solution	64
Database Layer Replatform Examples	65
Developing a PeopleSoft DB Replatform Solution	65
Building an SAP DB Replatform	69
Testing the Replatform Solution.....	72
Closing the Developing Phase	73
Key Deliverables Checklist for the Developing Phase	73
Major Milestone: Scope Complete	73
Chapter 4: Stabilizing Your Replatform Solution	75
Introduction and Goals.....	75
Testing the Replatform Solution.....	76
Stress/Load Testing.....	76
Regression Testing.....	76
User Acceptance Testing.....	77
Testing Criteria	77
Reporting the Results	78
Lifting Code Freeze — A Major Decision	78
Conducting the Pilot Program	79
Following Best Practices.....	79
Deciding Next Steps	79
Creating Pilot Test Reports.....	80
Release Readiness Checklist	80
Closing the Stabilizing Phase	80
Key Deliverables from the Stabilizing Phase	81
Major Milestone: Release Readiness Approved	81
Chapter 5: Deploying Your Replatform Solution.....	83
Introduction and Goals.....	83
Deployment Contingencies	84
Deployment Weekend.....	84
SAP Replatform Deployment Weekend.....	84
PeopleSoft Replatform Deployment Weekend	84
Siebel Replatform Deployment Weekend	85
Closing the Deploying Phase.....	85
The Warranty Period.....	85
Project Review.....	85
Key Deployment Deliverables.....	86
Major Milestone: Deployment Complete	86
Chapter 6: Operating and Maintaining the New Platform.....	87
Introduction	87
Monitoring Performance.....	87
SAP Performance Tuning	87
PeopleSoft Performance Tuning.....	87
Siebel Performance Tuning	87
Managing Operations on Windows	88
Microsoft Updates and Service Packs	88
Summary	89
Glossary	91
Acronyms	93
References	94
White Papers, Case Studies, Benchmarks and Links	95
Examples of Other Replatform Case Studies	96
Appendices	97
Appendix A: Windows Platform Capabilities	97
Appendix B: Templates.....	100

Preface

Overview

Welcome to the Microsoft Solution Guide for Replatforming Line-of-Business Applications from UNIX to Windows: SAP • PeopleSoft • Siebel. This guide provides a roadmap for how to plan and implement a replatform of line-of-business (LOB) applications from UNIX to Microsoft® Windows®. Replatforming an LOB application means changing the platform on which the application runs — the servers, network, storage area network, database and operating system — to another type of technology. Organizations may undertake this type of IT project to reduce the costs, improve the performance, or increase the quality of these systems.

An LOB application is a critical computer application for areas such as accounting, finances, customer relationship management, supply chain management, or resource planning application, that is vital to running an organization. Corporate Finance, Customer Care, and Human Resources are examples of three business departments that commonly depend upon LOB applications. LOB software systems such as customer relationship management (CRM), enterprise resource planning (ERP) and financial support systems (FSS) are used to manage the business processes of these departments.

In this market space, Systeme, Anwendungen, Produkte (SAP), PeopleSoft, and Siebel each produce top-tier LOB applications that are referred to by the company names. This guide specifically addresses replatforming for SAP, PeopleSoft, and Siebel LOB applications on Windows. It is based on the experience of consultants and organizations that have successfully completed replatform projects.

Important Every organization's implementation of these applications will vary and have special customization issues. It is highly recommended that you contact your SAP, PeopleSoft, or Siebel Certified Partner, and Microsoft Consulting Services, and use this guide in concert with their recommendations for your replatforming project.

Intended Audience

This document provides guidance to business and IT decision makers, as well as implementation managers and technical members of the team, who want to replatform LOB applications (SAP, PeopleSoft, or Siebel) from UNIX to the Microsoft Windows environment. The audience for this guide also might include certified partners and consultants who are helping an organization with a replatform project.

Regardless of job title, the team member who is responsible for managing the replatform project should read the entire guide. In contrast, individuals whose project responsibilities are limited to specific roles such as Development, Test, or Release Management, might want to read just the sections that pertain to their specific focus areas.

Knowledge Prerequisites for Understanding This Guide

Business and technology decision makers should be familiar with IT project management methodologies. Implementation managers should have prior experience implementing an SAP, PeopleSoft, or Siebel application for their organization and be familiar with its

workings. Development, test, and operations personnel should have a general understanding of the Windows and UNIX platforms, as well as their specific LOB implementations.

How to Use This Solution Guide

The organization of this guide is based on the industry-proven need to manage IT projects according to a disciplined process that improves the odds of project success. It is designed to be used with a companion guide, the *UNIX Migration Project Guide* (UMPG). Both guides follow the structure of the Microsoft Solutions Framework (MSF) process model, which defines five distinct phases for IT projects: Envisioning, Planning, Developing, Stabilizing, and Deploying that are separated by milestones. While this guide focuses on the solution-specific information, the UMPG focuses on the organization of processes and people, using the MSF Process Model and the MSF Team Model as basic models.

Note Although the two guides are designed to be used together, it is not necessary to follow the MSF processes and team guidance described in the UMPG if your organization has an alternative project methodology in place. In that case, the UMPG would be used merely to map the MSF phases and team structure to the elements of your chosen methodology. Readers who will use this guide to implement a project should read at least the overview of MSF in the UMPG to familiarize themselves with the MSF Process Model, the MSF Team Model, and MSF terminology.

In-depth information about MSF is available to interested readers on the Microsoft Solutions Frameworks Web site at <http://www.microsoft.com/msf>.

Because organizational personnel and project team members tend to have different levels of involvement during different phases of the project, the division of content according to project phase also supports the ability to focus on the material that is most relevant to a particular responsibility. The following table identifies likely readers by phase:

Table 0.1 Readership by Organizational Responsibility or Project Team Roles

Guide section or project phase	Organizational responsibilities of likely audiences	Project team roles of likely audiences
Preface	All readers	All readers
Envisioning	Business decision makers, technical decision makers, architects	Product Management, Program Management, all team leads
Planning	Business and technical decision makers, architects	Product Management, Program Management, all team leads
Developing	Development team, certified consultants	Development, all team leads
Stabilizing	Test and quality assurance	Test, Development, Release Management, all team leads
Deploying	Operations and production support	Release Management, all team leads
Operations	Operations	Release Management

Organization by Chapter

This guide is organized into a preface, six chapters, a glossary, acronym list, reference section, and appendices.

- Chapter 1 — Envisioning Your Replatform Solution. This chapter describes how the business and IT reach agreement on the high-level requirements and goals of the project. It covers forming a project team, creating a vision/scope document, assessing the current state of the IT environment, and assessing project risk. This chapter also provides a link to a TCO tool for evaluating your organization's TCO.
- Chapter 2 — Planning Your Replatform Project. This chapter describes the detailed planning activities for a replatform project. It defines what will be built and deployed, as well as how and when it will be built. The concepts of code freeze and cutover are discussed in the deployment plan.
- Chapter 3 — Developing Your Replatform Solution. This chapter provides a high-level overview of the development steps necessary to replatform. It also provides examples of a DB-layer replatform project for SAP and PeopleSoft applications. The actual development must be done working with a certified partner.
- Chapter 4 — Stabilizing Your Replatform Solution. This chapter provides information about how to stabilize your builds of the replatform project. This chapter also provides information about how to complete testing of the replatform.
- Chapter 5 — Deploying Your Replatform Solution. This chapter discusses executing the cutover to the new platform, and weighs the pros and cons of three cutover approaches.
- Chapter 6 — Operating and Maintaining the New Platform. This chapter provides links to information about the operation and maintenance of the new platform.

Contributors

Authors

Arshia Tayyab, Computer Consultants International, Inc.

Mark Prazak, Microsoft Corporation

Program Management

Dhilip Gopalakrishnan

Pablo Barcenas Ibinarriaga

Arshia Tayyab

Reviewers

Thomas J. Junk, CUNA Mutual

Noel Thurlow, WMC Resources Ltd.

C Shekhar Khanijow, Global Outsourcing Advisory

Nathaniel D. Holcomb, Computer Consultants International, Inc.

Lubna Noman, Computer Consultants International, Inc.

Murrell Ray White, Junior, Computer Consultants International, Inc.

William S. Holderby, Computer Consultants International, Inc.

Ashish Baveja, Symantec Corporation

Siva Kumar, ASK-IT Corporation

1

Envisioning Your Replatform Solution

Introduction and Goals

This chapter provides the background information required to complete the first phase of a replatform project, which represents an early form of planning. The purpose of this initial work is to get the project started by achieving a basic agreement between the business and IT on the goals for the project as well as its constraints.

The guide makes the following basic assumptions about the state of the organization at the beginning of Envisioning.

- A decision to investigate the project has been made by an IT manager who has sufficient budgetary authority to fund the investigation (Envisioning and Planning phases), and who may or may not have sufficient budgetary authority to approve the entire project.
- Rough estimates indicate that project benefits will exceed its costs, but detailed Return on Investment (ROI) calculations cannot be made until the project scope is defined.
- Whether the team that undertakes the project is made up entirely of in-house staff or it includes consultants, the work of Envisioning and Planning remains the same.

During this phase the core project team is assembled and a vision/scope document is created. The team also provides a rough estimate of the project timeframe and duration, and makes an initial assessment of the risks the project will need to assume. The phase ends with a major milestone when the vision/scope document is reviewed and approved by customers, key stakeholders, and the project team. This usually happens at a formal Vision/Scope Approved Milestone meeting.

The major tasks and deliverables for the Envisioning Phase are listed in Table 1.1.

Table 1.1 Envisioning Phase Major Tasks and Deliverables

Major Tasks	Deliverables
Setting up the project team	
Creating the vision	Vision statement
Defining the business goals	List of business goals that incorporates the goals of all stakeholders for the project
Defining high-level requirements	List of high-level requirements that the

Major Tasks	Deliverables
	solution must meet
Assessing the current situation	Baseline inventory and current state assessment
Defining the scope of the project	
Incorporating the vision statement, business goals, a brief summary of the current situation with respect to the business problem, high-level requirements, solution concept, and solution scope into a vision/scope document	A vision/scope document
Creating a project structure document	Project structure document
Assessing project risk	The initial risk assessment document
Closing the Envisioning Phase at the Vision/Scope Approved Milestone	Sign-off by customer, stakeholders, and project team that milestone has been met

Refer to the UMPG for more detailed discussions of how these tasks and deliverables can be approached and responsibility assigned for them.

Creating the Vision

The vision statement expresses the vision in a few words as an easily understandable concept that can serve as a general guide to project decisions, which need to align with it. The vision statement is unique to each organization. Some sample vision statements that might guide a replatform project are:

- All of our databases run on a single platform.
- IT operating expenses are 15% higher than last year with no increase in service levels.
- All systems are interconnected through one technology and platform.
- Our entire sales field staff can access our LOB applications remotely.

A vision for the solution

- Defines the business opportunities or problems that the solution is expected to address, including major business and user requirements.
- Describes the current state of the customer's environment (at a high level) with respect to the technology being considered.
- Describes the desired future, "to-be" state of the organization's environment from an unbounded viewpoint.

Understanding the Business Problem/Opportunity

Creating a vision for the solution that is shared by both replatform project team members and stakeholders begins with a clear articulation of the business problem or opportunity that is the reason or justification for the project. Various stakeholders may advocate for a number of known problems and recognize different opportunities that they would like the replatform to address. To enable the team, business sponsors, and stakeholders to stay focused on solving the same problem, it is important to explore the range of problems and agree on some priorities.

Some sample business problems and opportunities that might drive a replatform project are:

- We must cut operations costs significantly in order to remain competitive.
- We are losing customers because they are dissatisfied with the level of service we are providing.
- Our competitor was just forced to close a product line, giving us the opportunity to grab significant market share if we can give our sales force all of the support they need in our CRM system.
- We want to launch a new product and our SAP database is not configured to handle the high concurrent user load we anticipate.
- New regulations were just passed to go into effect next year and our LOB system does not meet the new standards.

Consider the following questions to pinpoint your organization's most important problems or opportunities that could be addressed by replatforming.

- Does the application and surrounding infrastructure (servers, network, storage area network, and operating system) cost more than the original projections suggest it should? Is there a problem with excessive costs or an opportunity for cost savings?
- When is the application scheduled for an upgrade or a renewal of its license and how much will that cost?
- Does the application experience performance problems on the current platform?
- As your organization grows and your business needs change, is your LOB software system sufficiently flexible and tolerant to change to accommodate these changes?
- Likewise, as business and regulatory environments change, will your LOB software system be capable of adapting to meet possible new demands?
- How easy or difficult is it to find knowledgeable, trained personnel for the application and current platform?

If, in answering these questions, you find that the performance and cost factors of the current applications and platform are not optimal, replatforming might provide an efficient, cost-effective method of improving and maintaining the LOB applications. You might also conclude that your current system satisfactorily meets your business needs or the replatform would cost more than can be returned on your investment. In these cases, replatforming might not be advisable.

Sometimes an organization's current IT environment supports a change. For example, if an organization is already using the Windows platform and technology for some services and trained staff is available to handle that technology, then moving LOB applications to Windows server-based applications becomes even more feasible. SAP, PeopleSoft, and Siebel applications have been fully tested and have run very effectively on the Windows platform for many years.

Note It can be valuable in the project's Envisioning Phase to look at case studies, ask for customer references, compare platform benchmarks, and examine proof of concepts for customer situations that are similar to yours. This may give you unexpected insights and enable you to make more educated decisions.

Defining Your Business Goals

The specific business problems and opportunities that you defined will help you to decide upon the business goals that you want to pursue. It is very likely that you began Envisioning with specific business goals in mind, but it is worthwhile to define them in concrete terms and prioritize them. This will help you to frame a vision statement that captures the important ones. It will also help you determine the feasibility of replatforming as a way to achieve them. The process of articulating your business goals is likely to lead to some insights about why you have not been able to meet them with your current LOB systems. Following are some sample business goals that correspond to the business problems and opportunities that were previously listed.

- Cut IT operations costs 20 percent over the next year.
- Increase our customer retention rate 5 percent through raising service standards for all channels (e-mail, fax, phone, Internet, and mail)
- Win a 28 percent market share in the heart medication category in pharmaceuticals by providing superior support to the sales force.
- Achieve \$22,000,000 in sales in the first year of our new product launch while maintaining the service reliability that our existing customers count on.
- Stay in business by enabling our LOB system to meet new security regulations before they go into effect.

The need to cut IT operations costs is likely to be the primary business driver in a replatform project. Even if it is not, cost will certainly be a factor that the organization will want to consider. The need to increase the performance or quality of your LOB system can also be a business driver. In fact, your organization may have a number of related business goals that it wants to achieve. Documenting all of them in an explicit list will help you articulate a vision that expresses an ideal situation in which the primary ones are realized.

Defining High-Level Requirements

High-level requirements include requirements in many areas, such as business requirements, operational requirements, system requirements, and user requirements. High-level requirements to meet business goals can be defined in terms of three broad categories that reflect common ways of thinking about a system. The requirement categories are as follows:

- **LOB System Costs.** The components of IT costs might include hardware, licenses, and labor costs. If the business goal is to lower costs, these are likely areas in which an LOB replatform can do so. Labor costs might be lowered not only because of a more available resource pool with the needed technology skills, but also through higher labor productivity after the replatform.
- **LOB System Performance.** Performance components might include scalability and flexibility, and interoperability. Increasing performance for an LOB application refers to increasing its speed or processing power. For example, an application that originally served 500 local users with a five-second response time for the most frequently used screens might, because of changing business requirements, now need to serve 5000 worldwide users with a two-second response time for the same screens. LOB software system attributes that affect performance include scalability and flexibility, integration, and interoperability.
- **LOB System Quality.** LOB system quality might include the reliability, security, and manageability attributes of the LOB system. The quality of an LOB application refers to its capability to meet standards for information storage, information security, and use of information. In addition to internal quality standards, your

business processes are likely to be subject to government regulations such as those imposed by the U.S. Food and Drug Administration (FDA) and Health Insurance Portability and Accountability Act/Health Care Financing Administration (HIPPA/HCFA) in the healthcare industry, which contain their own standards. The reliability, security, and manageability attributes of both the LOB software system and platform affect the capability to meet such standards.

These quality standards must be included in the LOB application design. An LOB application version upgrade might even be necessary as a part of your replatform effort because the new version of the application meets the organizational and industry quality standards and has the quality features required to run the organization, which the older version of the application no longer meets.

Business goals often require improvements in more than one of the three categories, and replatform projects are likely to add value in more than one. Many decision-makers feel that if significant improvements in one of these three areas are gained without degrading the other two, then replatforming is worth considering. Different stakeholders are likely to champion different goals and requirements. Inevitably, conflicts between the goals and requirements will arise later in the project even if they are not immediately apparent, so it is important to achieve agreement on the priorities as a basis for decision-making.

The following sections discuss requirements categories in more detail. Use this information to determine and document the requirements that your solution must meet, based on your goals.

LOB System Cost Requirements

The cost to an organization of an LOB application should be measured not only in terms of its initial acquisition cost but also in terms of the cost of running it, which includes the platform it runs on.

Total Cost of Ownership (TCO)

A valuable concept for calculating and comparing costs is the total cost of ownership (TCO). In addition to the up-front costs of hardware and software for both the application and platform, TCO includes the later costs of:

- Platform and database licenses
- Installation
- Customization
- Personnel training and certification programs
- Technical support
- Upgrades
- Repairs

The maintenance and support costs in this list incorporate the effects of a related factor, operations productivity. Enhanced operations productivity lowers support overhead by enabling users to do more without administrative intervention. Likewise, the costs of technical support and personnel training reflect, at least indirectly, the availability of the resource pool for personnel available to administer a given platform.

The most significant opportunities for lowering TCO in a UNIX to Windows LOB application replatform are usually in the areas of hardware and licenses for the database and platform.

Estimating TCO

An easy way to compare TCO for LOB applications is to use CIOview Player, a preliminary sizing/TCO tool for SAP, PeopleSoft, Siebel and other LOB applications. A Microsoft Consulting Services representative can conduct this estimate for you, or you can purchase the tool and conduct the estimate.

The tool is provided by the company CIOview. For more information about the company and product, visit <http://www.cioview.com>.

For download instructions and a download link, registration information, and licensing information, visit <http://www.cioview.com/pdf/InstallingfromDownload.pdf>.

You can download CIOview Player from <http://www.cioview.com/download/content/Playersetup.exe>.

For more information about TCO studies and Microsoft SQL Server, visit <http://www.microsoft.com/sql/evaluation/compare/tco.asp>.

LOB System Performance Requirements

The need for a specific level of performance is closely tied to the services that the LOB application provides to your organization. An increase in the need for services will often lead to the need for increased performance. The following is a short list of types of business changes that can result in generating new needs. Use this example list to begin the discussion of your organization's increased performance needs:

- Expansion of product line and services your organization offers
- Growth in customer base
- Increase in number of employees who use the LOB application
- Geographical expansion of business
- Mergers and acquisitions
- Changes in the organization of business units
- Changes in the competitive environment
- Changes in customer preferences

The following sections discuss the system attributes that affect performance.

Server Performance

LOB applications typically must run on high-performance server systems. Examine your system to determine if it has the capability to accommodate the performance levels necessary to meet your business goals. The following capabilities can enhance the ability of a server system to meet server performance requirements for speed and power:

- A boost in system startup speed. Increased speeds can eliminate many unnecessary delays that can occur when logging onto a network.
- Network load-balancing and load-sharing services (clustering).
- Improved memory tuning for more efficient use of larger memory maps.
- Processor affinity tuning that enables IT managers to specify the optimum processor configuration.

Scalability and Flexibility

If growth is planned or expected in the future of your organization, then scalability and flexibility are important attributes to consider. Specifically, factors such as the projected number of concurrent users, the agreed-upon reasonable response times, and variations

in the types and size of data captured in databases affect a system's need for scalability and flexibility.

Achieving scalability and flexibility tends to be a function of the system design. Can capacity be increased by adding a single server? Can components be replaced easily because they are "plug and play?"

Integration

Integration involves the capability of two applications to work together effectively. The main integration features of an LOB software system that affect performance are:

- Synchronization services, which enable different system elements such as the LOB application and databases to exchange data files.
- Identity management, which authenticates user profiles. Logon performance is enhanced when this capability is automated across a number of systems.
- Application integration, which is the ability to open an application from inside another application and access its capabilities. Siebel, for example, can send e-mail messages from inside Siebel saving time for the user.

Interoperability

Interoperability involves the capability to exchange information between different applications and other parts of systems. Interoperability requirements as they affect performance are likely to accompany events such as mergers, acquisitions, and other reorganizations of business structures that necessitate changes in the way your LOB software system interacts with other systems. In this situation it is common for legacy information systems to co-exist with existing infrastructure. The replatformed system must still communicate with other applications and use online and batch interfaces with the applications. These online and batch interfaces must work on the new platform. Consider what other systems and applications, both within and outside your organization, will need to communicate with the replatformed system and applications. Additionally, the replatform event itself is frequently accompanied by other hardware and software upgrades that raise interoperability issues.

LOB System Quality Requirements

The following sections define quality in terms of the system attributes that are normally associated with it. Reviewing these discussions should help you describe your requirements for quality.

Reliability

As an organization becomes increasingly dependent on LOB applications, reliability becomes increasingly more critical. Any unplanned downtime can reduce productivity and revenue. Integrated systems are often expected to be up, running, and available at all times, making information available to users and customers on demand. A common way of expressing this availability is system uptime of 99.999 percent, or "five nines." You need to evaluate the level of reliability that is required to meet your business goals. The replatform development sub-team will convert this information into system design requirements.

The following methods can be used to increase reliability:

- Clustering servers to reduce single-point failures.
- Mirroring servers and using Redundant Array Independent Disc (RAID) technology.
- Building redundancy to provide reliability for 24X7 operations.

Security

Because increasing volumes of sensitive data such as customer and financial information are stored on LOB systems, organizations must take every possible measure to maximize security. Managing a secure environment, however, presents an increasing challenge as the complexity of your computer system grows. Organizations now have extended data and infrastructure relationships that link suppliers and customers together, and managing security can be overwhelming and costly without the right infrastructure. Identify your security requirements with respect to the following areas:

- Identity management
- Data integrity
- Protection from intentional attack

Manageability

Manageability refers to the level of intervention required for your IT personnel to maintain, change, tune, and oversee the operation of your servers, networks, and applications. Enhanced software-restriction policies can improve system integrity and manageability, requiring less intervention. You might have dozens of IT personnel working on your IT infrastructure and want their tools to be both state-of-the-art and extensible across your entire organization. Identify your manageability requirements with respect to the following factors:

- CPU utilization percentage
- Concurrent users
- Total number of users
- Response times

Vendor Support

When changing platforms and technologies, consider whether the new technology and the vendor that provides the technology is long-lasting, and will be available in the future to support your organization.

Defining the User Profiles

Creating user profiles will help you to define user requirements. To capture a clear description of each user, the User Experience Role leads the effort to create profiles of each user class (for example, manager, sales and so on). In LOB applications the most important user profiles are usually those of the administrators (both IT and business) and the end users. However it is highly recommended that you come up with a list of all the user classes as well as the security and access levels that are needed for each user class.

Include the following types of information in the user profiles:

- A list of all tasks that each user class performs. This will define the screens they need to access.
- Essential expectations of the end user from the application interface that must be met in the new environment.
- The language localization needs for the end users.
- User documentation, online help, and FAQ lists that are needed.
- The physical location of the end-users: Factors such as locations, number of users at each location, the bandwidth and usage of network links between the sites should all be understood and documented.

- Windows GUI Interface to add users in the database.
- Training on the SAP, PeopleSoft and Siebel as a business administrator to add business users to the correct profiles in the application.

Setting Up the Replatform Project Team

During the Envisioning Phase you will assemble your core project team, which should be in place by the end of the phase.

Note Although setting up the team is the logical first step for a project, many of the early discussions about the business goals and vision for the project are likely to occur between just one or two members of the team and business decision makers who are outside the team. This is the reason for delaying the team discussion until this point in the guide. All core team members will participate in the decisions as they join the project.

This guide defines team members in terms of six role clusters (usually shortened to "roles") that are each responsible for a specific goal that is vital to overall project success. Each role may be performed by one or many persons, depending upon the complexity and size of the replatform project.

Refer to the "Setting up a Team" section of the *UNIX Migration Project Guide* for a description of the team roles, the functional areas that each one covers, and a list of their typical responsibilities during a project. For an in-depth discussion of the MSF team model, see the MSF team model white paper v3.1 at <http://www.microsoft.com/msf>.

The use of roles that are independent of any one organization's job titles provides a consistent way to refer to project responsibilities throughout this guide. You may want to map the roles to a different type of team structure.

Team members can be drawn from both inside and outside your organization. In order to assure continued product support, SAP, PeopleSoft, and Siebel require that one of their certified partners or certified consultants participate in the project. The extent of the certified partner's involvement varies; you may want them to take ownership of the entire project, act only in an advisory capacity, or participate at a level between these extremes. If you have a lot of in-house technical expertise, your need for them is likely to be smaller.

The challenge to in-house staff who is involved on the team is to make efficient use of the special expertise of the certified partner. When choosing and hiring the partner or consultant, define clearly the scope and services provided by their role to ensure that you get the proper skill sets and an optimal staffing level. (See the end of this section for partner contact information.).

Table 1.2 lists each role cluster and, where relevant, specific skills and knowledge required for that role in a replatform project beyond those normally required for the role. The range of required expertise for the Development Role is particularly wide, making it clear that it needs to be filled by several persons. In fact, the Development and Test roles are often staffed by entire sub teams.

Table 1.2 Replatforming Project Role Clusters

Role cluster and goal	Required skills and knowledge that are specific to LOB replatform projects
Product Management— satisfy customers*	

Role cluster and goal	Required skills and knowledge that are specific to LOB replatform projects
Program Management — deliver the solution within project constraints	<ul style="list-style-type: none"> — Experience in the installation of your organization's SAP, PeopleSoft, or Siebel applications — Sufficient understanding of technology to ensure that technical activities in the project plan are adequately addressed — Experience in replatform projects (This role is likely to be staffed by the certified partner.)
Development — build to specification	<ul style="list-style-type: none"> — LOB operations technologies. Sufficient expertise in SAP, PeopleSoft, or Siebel technologies to enable the installation, tuning, and monitoring of the application (likely to be supplied by the PeopleSoft or Siebel certified partner or the SAP certified consultant). — Microsoft technologies. Expertise in Windows platform and SQL Server (could be provided by a Microsoft Consulting Services (MCS) expert experienced in managing and controlling the SAP, PeopleSoft or Siebel application (likely to be supplied by your organization's current application administrator) — LOB application development. Experience developing with your SAP, PeopleSoft, or Siebel applications, including modification of application components — Database administration. Sufficient experience in database administration to ensure that the database layer is properly configured for its intended purpose, database tables are mapped correctly, and data is correctly moved from one system to another (likely to be supplied by the current DB administrator) — Systems administration. Experience with systems administration, including the proper installation, configuration, and management of UNIX and Windows systems
User Experience — enhance user effectiveness	<ul style="list-style-type: none"> — Ability to recruit volunteers and coordinate activities of a "Super User Group"
Test — approve solution for release only after identifying and addressing all quality issues	<ul style="list-style-type: none"> — Sufficient understanding of the SAP, PeopleSoft, or Siebel application functionality and its interfaces to create and execute a test plan and test cases — Familiarity with your organization's individualized implementation of SAP, PeopleSoft, or Siebel (This role is often filled by your organization's current test team and is likely to require several persons.)
Release Management — smooth deployment and ongoing operations	<ul style="list-style-type: none"> — Understanding of how to run systems in parallel (This role is often filled from your organization's current operations staff.)

*Customer is defined as the individual or organizational entity that expects to gain business value from the solution (also known as the business sponsor). For replatform projects that are driven by the need to save IT operations costs, the customer might be

the CIO. For replatform projects that are driven by a desire for increases in quality or performance, customers would include the business units that are asking for those increases.

Important Best practices show that staffing the replatform project roles with people from the organization's current development and operations staff enables the replatform project to run more smoothly. The reason for this is that the current staff is familiar with all of the application customizations, operating system tweaks, DB optimizations, and middleware integration. In this way, in-house knowledge is most effectively used in conjunction with the certified partner.

The trade-off is that the replatform project might take longer to accomplish if these staff members cannot be freed from their ordinary responsibilities of maintaining the production system.

Some common organizational titles of persons who are likely to fill the team roles, particularly the development sub-team, are as follows:

- SAP, PeopleSoft, or Siebel technologies expert (certified partner)
- Microsoft technologies expert (certified partner)
- SAP, PeopleSoft, or Siebel application administrator
- DB administrator (Oracle, SQL Server)
- System administrator (UNIX and Windows)
- Business analyst (SAP, PeopleSoft, and Siebel)

The Change Control Board

Another important group in a replatform project is the change control board (CCB), a group made up of representatives from each of the following areas:

- A business stakeholder for each business unit that is affected by the replatform
- Operations
- Production control
- Development
- Testing and quality assurance (QA)
- Program Management

The CCB handles change control procedures for the project. Some organizations have a standing CCB that is part of IT operations, which must carefully control any changes to the production environment. If your organization does not have a CCB, the Release Management Role should form one for the project after the Vision/Scope Approved Milestone.

The Super User Group

Another important contributor to the project can be a special group drawn from different areas of the business. In an enterprise-wide LOB environment, the application users are the principal end users of a replatform solution. The User Experience Role often recruits a "Super User Group" from the various departments or functions which use the LOB application modules being migrated (for example, Invoicing, Purchasing, Sales, Service, HR). These individuals should be the most knowledgeable about actual business processes in these areas; they are often used as trainers or mentors for new employees and are usually considered as "gurus" by their respective departments. User Experience coordinates the work of this group of expert users.

This group can contribute in the following ways:

- Help the User Experience Role to develop use cases
- Supply user requirements and preferences
- Provide feedback on what does and does not work in the context of lab "walkthroughs," usability tests, user acceptance tests, and pilots
- Function as platform evangelists to other users in the organization, helping to overcome resistance to change by explaining reasons for the replatform and extolling new features that the new system provides
- Learn the new system early and provide informal training to other users on the new system
- Reduce the burden on technical support staff by being available to other users for simple troubleshooting

To gain the maximum value from a Super User Group, keep the group informed and up-to-date with weekly sessions on what the system will look like when it goes live. Listen to input from your user group and make changes as appropriate to the project for the highest level of success. Effective use of a Super User Group tends to lead to a higher user satisfaction rate after adoption of the new system. Also, this higher level of acceptance makes IT management of the new system easier due to better understanding and fewer conflicts between administrators and users.

Engaging Partners or Consultants

To find a certified partner for your LOB replatform project visit the SAP, PeopleSoft and Siebel Web sites for partner information:

SAP Certified Consultants

SAP customer service provides a list of Certified BASIS consultants in your area.

SAP Partners

Refer to <http://www.sap.com/partners/index.aspx>

PeopleSoft Partners

<http://www.peoplesoft.com/corp/en/partners/index.jsp?site=corp>

Siebel Partners

<http://www.siebel.com/partners/find-partner.shtml>

Microsoft Key Alliance Partners

Links to Microsoft partners are:

For SAP: <http://www.microsoft-sap.com/resources.asp>

For Siebel: <http://www.siebelonmicrosoft.com/mssiebel/partners.aspx>

Microsoft Partner Community

The Microsoft Partner Community is comprised of software and service partners who provide complementary technologies and support for Microsoft products. It is a global network of Systems Integrators (SIs) and Independent Software Vendors (ISVs).

- For a complete listing of Microsoft partners who provide replatforming services or tools, visit <http://www.microsoft.com/windowsserver2003/partners/default.msp>.

- For a list of partners who provide services and tools specifically related to Microsoft Windows Services for UNIX, visit <http://www.microsoft.com/windows/sfu/partners/default.asp>.

Assessing the Current Situation

You must identify the gaps between the current and desired states of a business to create a solution that effectively addresses your organization's business goals. Performing a gap analysis between your "as-is" system and your "to-be" system helps to identify a path to the desired state of a business need (your vision statement). Assessment involves gathering information (taking a baseline inventory of your current "as-is" system) and then analyzing the implications of the information in terms of how it will affect the replatform. Different types of business and technology issues that you need to assess might include business policies, business operations, existing systems, their load capacities, and other infrastructure components that will be affected by the solution.

Important Assessing your organization's current situation for a replatform project is extremely complex. Your certified partner has the knowledge necessary to guide you through this process. They typically meet with you in a Joint Application Development (JAD) session or a strategic planning session and by asking you a thorough set of questions about your business goals and current situation, they help you determine what you need to achieve your goals.

Creating the baseline inventory prior to meeting with your certified partner will save time and expense by enabling you to provide ready answers to your partner's questions.

Creating a Baseline Inventory

To perform and validate your current system inventory, make sure that the inventory takers are people with all the necessary understanding and information about the original application and system architecture and any customizations that have been made, as well as an understanding of the new application versions and system architecture. This work can be performed (or at least managed) by some members of the project team while others focus on defining the desired future state.

Before you start, do some planning, including:

- Decide on the scope for the inventory and the analysis
The inventory and analysis follow the law of diminishing returns; that is, at some point you gain very little more information relative to the effort required to get it. Therefore size these activities relative to the scale of the replatform.
- Identify data sources
Some organizations maintain a system inventory, which is a good starting point. Next, look to current staff members who worked with your initial LOB application implementation partner and certified consultants to arrive at your current implementation. They know the specifics of the application, the operating system, the database, and the middleware that was configured in the initial installation and production deployment.
- Arrange interviews and workshops with the key personnel needed to assist the partner, if the inventory and analysis is performed by your certified partner

It is useful to think in terms of platform layers — the operating system, database, application, and middleware layers — for both the inventory and the subsequent analysis.

The following categories reflect these layers. You will collect information on server configurations, network, desktop software, and all relevant hardware:

Application

For each SAP, PeopleSoft, and Siebel application, record:

- The number of modules installed and their respective versions
- Any new module requirements you anticipate for the near future
- How the application was customized
- Whether anything was hard coded
- Different types of user profiles (such as system admin, manager, sales agent, and so on)
- The current number of concurrent users for each module
- Do not count users more than once, and categorize them according to usage profiles: heavy, medium, light, and self-service.
- The application security that has been implemented

Database

Include the following information in your inventory:

- The current size of the database
- The version of the database
- The number of active users of the database
- The number of total users of the database
- The volume of batch, reporting, and query users

Specify how much this activity overlaps with normal online transaction processing (OLTP), and if there are specific timing requirements for completing certain batch processing operations.

- The libraries that are currently used
- How the database is configured to communicate to the application

Operating System and Supporting Infrastructure

Collect information on all of the devices that the original application uses. These devices might have their own dependencies; so include all related devices as well. For each device, record the following information:

- Device name, type, and purpose
- Software versions that the device runs
- The version of the OS and the patch number that you are using
- Mechanisms that the application uses to access the device
- Dependencies on other devices and software packages
- Whether the application will still use the device on the new platform
- Number of environments that the OS is installed in (development, test, staging, conversion, production)

The information that this inventory provides will help you identify potential replatforming issues, for example, whether Windows can support each device or whether you should seek an alternative device.

Middleware

LOB applications typically interact with other applications and software packages that are essential for the application to run with all its intended features and functions. These applications and software packages can include database management systems, messaging and groupware applications, content management and Web server software, accounting software, resource management software, and sales automation software. Although it is a large undertaking, take a complete inventory of all the interfaces that the LOB applications use. For example, you might find a little file that passes to another system and has never been inventoried as a batch or online interface. Sometimes the inventory process finds small, highly reliable systems that were developed in-house and run in the background and never generate an error, so few if any of the IT staff are aware of the existence of the files or systems.

To understand how each package will affect the replatform, identify it and document:

- Any specific configurations in the middleware for interacting with the online or batch interfaces.
- Any specific settings or parameters in the middleware affecting how it communicates with other systems.

Replatforming one application can affect a number of interoperability factors with this external software, such as:

- How the applications communicate.
- How the users access the applications, for example, through a single interface or portal, or through multiple interfaces.
- How security (authentication and access controls) is managed between applications.

External applications might have dependencies that affect the replatform. For example, a dependent software package already running on Windows might require an older version of Windows than the one intended for the replatform. The team must then decide whether to upgrade the external software package to a more recent version or use two servers that each run a different version of Windows in parallel. Each option offers trade-offs that are best explored through a cost-benefit analysis.

Overall System Architecture

Document the following:

- The number of application servers
- The number of disks, or other hardware and devices used
- The current allocated memory

Performing the Initial Assessment and Analysis

Whereas the baseline inventory provides an "as-is" snapshot of your current system and defines the starting point for any changes you make, the assessment has a related but different purpose. It will help you evaluate how well your current system meets your business goals—both your long term strategic goals and your shorter term tactical objectives. The resulting gap analysis summarizes the differences between the as-is and "to-be" states of your system. It will help you work out solution alternatives that meet both business and technical requirements, bringing you closer to your visualized "to-be" system.

Organizing Your Assessment by LOB Software System Layers

Organize the assessment, like the baseline inventory, on the basis of the four layers of your LOB software system (OS, DB, application, and middleware layers). These four layers will become the focus for developing the solution concept for your replatform project, discussed later in this chapter.

The assessment questions listed in Table 1.3 are ones that you should always consider in a replatform project no matter what your business goals are. Many of them are open-ended and may lead to lengthy discussions. Others may require a lot of research to answer (your certified partner can be a significant resource here). Still others are specific, but the answers will lead to further questions that need to be explored.

Table 1.3 Assessment Questions

OS Layer
Do the operating system licenses need to be renewed?
How many licenses need to be renewed?
Are the licenses per site, or per processor?
Are any version upgrades needed or required in the near future?
Can you run the version of software on your current infrastructure and operating system or is the operating system outdated?
Are there other operating systems with better application support?
Are there other operating systems with lower or more cost-effective licensing fees?
DB Layer
Is the database providing the necessary functionality?
Is the database optimized for high performance?
Where does your data reside?
Does it reside in multiple data stores, and if so, do you have effective processes for data cleansing, management, and end-to-end data access?
What is the anticipated growth of the database over the next two to three years?
Are you planning to move from your current database to a different database?
Are there other databases that might provide functionality, including performance capability, that would better suit your business goals?
Application Layer
Is the application server running on a different operating system from the database server?
Will the application provide more functionality on a different platform?
Will the application provide easier maintenance on a different platform?
What is the level of integration between applications?
Is it low enough to make it difficult to respond to changing market forces?
What are the application's performance metrics or benchmarks?
Do you expect to install additional modules in the near future?
Do you need to upgrade the application in the near future?
Middleware Layer
Is there an enterprise-level standard for middleware?
Was the middleware developed in-house?
Are you using a combination of different products that are available in the market?
Does the current middleware layer have any problems?

Do you want to standardize your middleware solution?

Does your middleware meet your business needs and requirements for interoperability?

Miscellaneous

What is the current repair and maintenance situation, and what are the costs?

Connecting LOB Software System Layers and Requirements

To get a picture of how each layer overall as well as specific facts about the layer affect the high-level requirement categories for the replatformed LOB system (cost, performance, and quality), use Table 1.4. Three additional columns have been added to Table 1.3 that indicate relationships between the requirements and the questions. You may wish to assess your current situation more granularly, focusing on the components of each of the categories.

Table 1.4 Assessment-Requirements Relationships

Assessment Questions	Related Requirements		
	Cost	Performance	Quality
OS Layer			
Do the operating system licenses need to be renewed?	X		
How many licenses need to be renewed?	X		
Are the licenses per site, or per processor?	X		
Are any version upgrades needed or required in the near future?		X	
Can you run the version of software on your current infrastructure and operating system or is the operating system outdated?	X		X
Are there other operating systems with better application support?		X	X
Are there other operating systems with lower or more cost-effective licensing fees?	X		
DB Layer			
Is the database providing the necessary functionality?		X	
Is the database optimized for high performance?		X	
Where does your data reside? Does it reside in multiple data stores, and if so, do you have effective processes for data cleansing, management, and end-to-end data access?		X	X
What is the anticipated growth of the database over the next two to three years?	X	X	
Are you planning to move from your current database to another database?		X	
Are there other databases that might provide functionality, including performance capability, that would better suit your business goals?	X		
Application Layer			
Is the application server running on a different operating system from the database server?		X	

Assessment Questions	Related Requirements		
	Cost	Performance	Quality
Will the application provide more functionality on a different platform?		X	X
Will the application provide easier maintenance on a different platform?	X		
What is the level of integration between applications?		X	X
Is it low enough to make it difficult to respond to changing market forces?	X		
What are the application's performance metrics or benchmarks?		X	
Do you expect to install additional modules in the near future?	X		
Do you need to upgrade the application in the near future?	X	X	X
Middleware Layer			
Is there an enterprise-level standard for middleware?		X	X
Was the middleware developed in-house?		X	
Are you using a combination of different products that are available in the market?		X	
Does the current middleware layer have any problems?			X
Do you want to standardize your middleware solution?	X	X	
Does your middleware meet your business needs and requirements for interoperability?			X
Miscellaneous			
What is the current repair and maintenance situation, and what are the costs?	X		X

Using Decision Trees

Using a decision tree to follow up on the questions of most interest to you enables you to drill down from your initial assessment questions to a level of detail that will help you determine if your current system is adequate to meet these goals or if changes must be made. Figure 1.1 illustrates this process. Although the process is logical, it also requires considerable knowledge of LOB software systems to do well. The person asking the questions must understand the implications of the answers well enough to know which follow-up questions to ask. The SAP, PeopleSoft, or Siebel certified partner can add significant value in pointing to the initial questions that merit follow-up and verbalizing the follow-up questions.

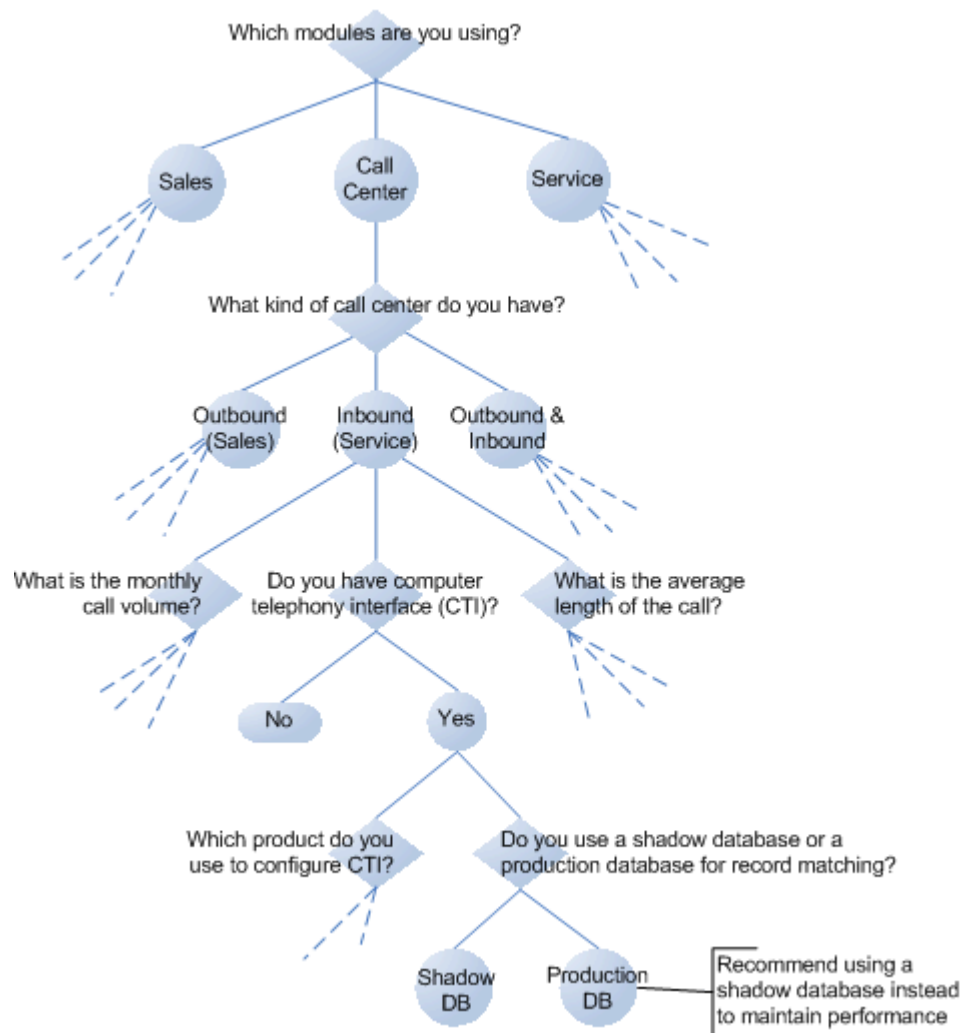


Figure 1.1 Assessment Decision Tree

Figure 1.1 shows an example of the type of decision tree that will help you to assess your current system. In this illustration, the Call Center branch is explored until you arrive at answers at each sub-branch that indicate whether your current system is sufficient for your business goals, or if you need to make changes to meet those goals. For example:

- Depending on your answer to the "monthly call volume" question, you might need to increase the number of servers in your system.
- The "average length of call" question helps you to measure the impact call lengths have on your system performance and your organization's productivity.
- Following the branch of questions related to the use of computer telephony interface (CTI) helps you to determine whether you need to implement a shadow database.

Best Practices for Assessment and Analysis

The results of the analysis will be used by decision-makers, managers, and the replatform team.

Some general guidelines for decision makers include:

- Focus on the application in the organization that makes the most sense to replatform first.

- Look at both the tactical and strategic implications of the replatform.

Some general guidelines for managers and team members include:

- Ensure that you understand the end-to-end LOB system and how it fits into the overall enterprise architecture of the organization. Changes to any part of the LOB system could trigger significant changes to the organization and to the organization's IT architecture.
- Understand and make clear distinctions between the original, out-of-the-box (OOB) architecture and the customizations. The customizations were introduced to help solve the organization's business problems and needs. These needs must continue to be addressed during the replatform project.

Creating the Solution Concept

The solution concept outlines the approach the team takes to meet project goals and requirements. The gap analysis that culminated goal setting and assessment activities leads directly to its creation. At this stage, the objective is not to conduct a detailed analysis of the different elements involved in the replatform, but rather to understand them at a high level and make the preliminary decisions on how to handle the replatform.

A solution concept:

- Describes at a high level how the solution will solve the business problem in terms of the approaches it will take to build and deliver the solution.
- Documents business goals, design goals and the required functionality of the solution,
- Calls out assumptions and constraints that apply to the solution
- Defines the success criteria for the project.

You will use the solution concept to provide the project team with limited but sufficient detail to prove the solution to be complete and correct; to perform several types of analyses including feasibility studies, risk analysis, usability studies, and performance analysis; and to communicate the proposed solution to the customer and other key stakeholders. The solution might be simple or complex, depending on the business goals that are driving the project and the requirements that must be respected while achieving the goals. Examples of simple solution concepts are:

- Change the operating system because licenses are expiring and offer the opportunity to achieve a significant savings in TCO, with no changes to the other layers.
- Change the database because performance and cost considerations call for a change. The DB technology change means that the OS must also change. Make no change in other layers, but test them rigorously for functionality.

A more complex solution would be required to address a situation in which you want to make changes to applications and business functions that require changes to all four layers of the system. The reasons might be that your current applications are so old that they are no longer supported by your software provider; your OS licenses are all due for renewal; and your user base has grown and continues to increase. Because of all these factors your system can no longer support your business functions and growth, and your organization must undergo a major replatform.

When developing the solution concept, concentrate more on how to do the replatform process than on the specific technologies themselves.

The Solution Concept and LOB System Layers

LOB applications can be divided into four areas for purposes of defining at a high level how a replatform project will accomplish the agreed-upon business goals while meeting business and design requirements.

- **OS layer.** The OS layer is the operating system that the LOB application runs on. For a replatform the OS layer may move from UNIX to Windows.
- **Database layer.** The database layer is the database that the LOB application uses to store all the information for an organization. For a replatform the database moves from a UNIX-compliant database such as Oracle to the Windows-compliant SQL Server database.
- **Application layer.** The application layer is all of the software along the path from the LOB application through to the user interface. This includes the LOB application itself, web servers, web browsers, gateway servers, custom client applications, and so on. The most current versions of LOB applications typically use a Web browser to provide the user interface, but this is by no means universal. Older versions of some LOB applications relied more strongly on client/server architectures to present the user interface.
- **Middleware layer.** Middleware is custom-built software (or an SAP, PeopleSoft, or Siebel program) that provides services to batch and online interfaces to other systems in the organization.

Your solution concept should define which layers you plan to replatform, and the order in which you will do this. You will need to explore the pros and cons of each option by taking into account your business goals and the combined set of requirements that you defined.

You could replatform:

- Any one of the four layers.
- All four layers
- A combination of layers, for example:
 - Application and database
 - OS and application
 - Database and middleware

Consider the strengths and vulnerabilities of your current IT infrastructure when working out your solution concept. Consult with your certified partners and industry experts for advice and information about their experience with replatforms that are similar to the type your organization requires. Take into account the business goals you have defined, the high-level business, user, operations, and system requirements you have identified, and the impact on both the architecture and the organization, including operations and maintenance.

Special Considerations for the Application Layer

The application layer has additional considerations to keep in mind when defining your solution concept.

Version Upgrades

If you plan to replatform your LOB applications and perform a version upgrade in the same timeframe, it is recommended that you upgrade the applications first, and then initiate the replatform project. For example, an upgrade from Siebel version 99 to version 99.5 should be performed in two distinct steps; first upgrade from 99 to 99.5, then replatform to Windows. This results in a cleaner implementation after the replatform,

because any leftovers from previous versions will not have been transitioned to the new platform. (See the "Defining Scope" section in this chapter for a related upgrade discussion.)

Application Source Code

For the application layer, remember to address any issues concerning the actual application source code replatform, and, if applicable, the replatform of associated test programs and/or scripts. For example, a portion of PeopleSoft source code is still in COBOL, and the PeopleSoft installation process includes steps to compile that code in the Windows environment. Document the issues and your resolution of them.

Installation Issues

Find out if there are any differences between installing the application layer on the source system and the target system. Most SAP, PeopleSoft, and Siebel LOB applications are available as Web-based systems. The LOB products are moving away from the client-server model to a Web-based architecture where there is no code at the client level. However, there might be differences in how the browser accesses a database from a Windows platform compared to a UNIX system that the team must evaluate.

Application History

An LOB application implementation might have a long history; it might not have even begun as a UNIX application. Determining information about the application's history (for example, whether it was already ported to various UNIX environments) might provide useful insights into how the application should be replatformed to Windows. This would be an important consideration for replatforming SAP, PeopleSoft, or Siebel from one software version to another, for example, replatforming Siebel and upgrading from 6.3 to 7.x, or upgrading PeopleSoft from 7.x to 8.x. In these cases the LOB versions are also moving from a client-server application design to a Web-based, thin-client architecture.

Special Considerations for the Middleware Layer

Another layer that is very important yet a pain point for most of the organizations in the enterprise space is the middleware.

Interfaces with Other Systems

Whether it is SAP and Net Weaver, PeopleSoft and App Connect, or Siebel and UAN, evaluate whether the replatform will affect the online and batch systems accessed by the middleware. The replatform process must take care not to break the interfaces with other systems. Be sure that the baseline inventory of your current system includes all the interfaces with which these systems communicate.

Middleware Product Configurations and Dependents

Investigate the most commonly used middleware products such as TIBCO, Web Method, and Microsoft BizTalk™ Server and/or IBM MQ series to see how they are configured and what applications or systems are dependent on these middleware products.

Caution Replatform the middleware layer last.

It is not recommended that the middleware layer be moved during the initial replatform. Instead, take the inventory, do a phased-approach replatform, and handle the middleware layer last. Make sure that the replatform of other layers works and that the interfaces are working first. Then revisit the middleware layer as a separate enterprise-wide replatform decision rather than simultaneously trying to replatform the middleware for each LOB application.

Following are links to the guidance provided by SAP, PeopleSoft, and Siebel for their middleware:

SAP and Net Weaver

- <http://www.sap.com/solutions/netweaver/index.aspx>
- <http://www.sap.com/solutions/netweaver/businessbenefits/index.aspx>

PeopleSoft and App Connect

- <http://h71028.www7.hp.com/enterprise/cache/3909-0-0-225-121.aspx?MWServerName=mxMbmlART%2fatKsH7HY3smFP0jS72weahup+pF4vkTENCs3vJuoHilA%3d%3d&Mode=external>
- <http://www.serverworldmagazine.com/monthly/2002/11/peoplesoft.shtml>
- <http://www.notesnet.com/print/16468042d81e7ce985256c230051fb8a>

Siebel and Universal Application Network (UAN)

- <http://www.siebelonmicrosoft.com/mssiebel/Technology/BizTalkServer.aspx>
- <http://www.siebelonmicrosoft.com/mssiebel/Technology/BizTalkServer.aspx>
- http://www.siebelonmicrosoft.com/mssiebel/Downloads/DOC/Understanding_BizTalk_Server_2004.doc

The Timing Approach

Your solution concept should also define the timing approach you will take to replatform each layer. There are basically two possible approaches:

- **All at once (sometimes known as a big-bang approach)**
Replatforming all systems means that everything will move to one technology and one platform. This can streamline operations, maintenance, and troubleshooting. The disadvantage to this approach is that the entire system must be down for a longer period of time to handle the replatform.
- **A phased approach**
A phased approach will maintain dual platforms while you replatform layers in phases. This approach has the advantage of involving less downtime for the system. The disadvantage is that it might increase operations, maintenance, and troubleshooting management.

The phased approach is a necessity when your organization needs to maintain the UNIX environment even as it moves some layers of the LOB software system to the Windows environment. While some replatforms result in the removal of the UNIX environment, other environments need Windows to integrate (and coexist) with UNIX, and to run the same replatform application software that is currently running on the UNIX platform. This need for coexistence is an important factor in replatform projects. A replatform strategy that takes into account cross-platform software portability allows Windows to be accepted in organizations that require the application to remain on the existing UNIX platform.

Project Success Criteria

As part of your solution concept, you should define your project success criteria clearly. For example, will you measure your project's success by lowered operating costs, increased functionality, or some combination of those criteria? The following examples suggest a way of looking at defining success.

- **Implementation success factors.** A replatform project is, at its most basic level, meant to duplicate your current environment on a new platform. Although at the

platform level there is a fundamental change in how your system works, at the business level, ideally you would see no change in your operations. A minimum definition of a successful replatform implementation includes the following:

- There is no disruption of your current business operations.
- There is no loss of functionality (operations continue at the same level).
- There is a reduction in TCO.
- **Project management success factors.** As noted in the listing of the goals for team roles, delivering the solution within project constraints is an important success criterion for projects. Project constraints, simply stated, are "on time" and "within budget." Meeting these criteria is determined by how well the project is managed. You want to control cost and schedule overruns on the project. One way to manage this is to establish a margin (such as 5 – 10 percent) by which the project should not exceed limits. Also remember to account for capitalization, depreciation, and expenditure factors.

Important Make a clear distinction between improved functionality and new functionality. A replatform project often offers improvements such as greater speed and power to your current functionality. New functionality represents a change in both your business and system model.

If you are considering adding new functionality to your system (such as new applications or new modules), it is usually better to implement the new functionality in a separate project. There are times when it might make sense to combine new functionality with a replatform project, but there are many trade-offs in terms of time, cost, staffing, and other resources that you must consider to make this choice.

Defining Scope

After assessing the organization's current environment and defining the solution concept, you determine the scope of the replatform project. Defining scope represents the opportunity to put limits on both the solution and the project, and address the unbounded nature of the vision. *Solution* scope clearly delineates what stakeholders expect the solution to do, making it a basis for defining project scope and for performing many types of project and operations planning. *Project* scope applies the constraints imposed on the project by resources, time, and other limiting factors to the overall solution vision expressed in the vision statement. You will need to distinguish the scope of the solution from the scope of the project in cases where the solution will be implemented in stages or will require discrete projects to be run in parallel.

Scope decisions are often made in parallel with solution concept decisions. If the solution concept is generally to replatform the LOB system, different layers can be replatformed at different times, and project constraints may argue in favor of limiting the scope of any particular project.

Here is an example of what it means to limit the scope of a solution: The solution is to replatform the DB layer of an LOB application that is used around the world by people who speak five different languages. The database has five instances, each of which supports users who speak one of the five languages. One way to limit the project scope would be to replatform just one instance of the database (for example, the Spanish version or English version but not the Japanese version or French version).

Although the team should try to decide upon the solution concept before deciding upon scope, these decisions are not entirely independent and may need to be made in conjunction with each other. You may choose a more modest solution if it enables you

increase the scope of the project and achieve more elements of it within a given budget or timeframe. Alternatively a solution that calls for a phased approach might necessitate dividing the work into a number of serial projects. See the UMPG for a fuller discussion of scope and how to define it.

Following are some examples of issues that can affect the scope of your project:

- If you are going to use SQL Server you must replatform the DB server to the Windows platform, even if other applications are still running on a different platform.
- Your budget: If a complete replatform of all four layers costs more than you can currently afford, you might choose to limit the replatform to areas identified as crucial and revisit replatforming other layers at a later stage.
- Software upgrade issues: If a new version of the software is expected in the near future, you can either (1) replatform now and upgrade later or (2) delay the replatform long enough to upgrade first and then include the upgrade in the replatform. You need to investigate the impact of the upgrade on performance, database access, installed modules, server configuration, and user access.

One reason not to upgrade is:

- The upgrade might necessitate moving to a new application architecture (from client server to Web-based), and this is likely to represent extra complications and extra work.

Reasons for upgrading include:

- It might fix some of your current business problems that were drivers for the replatform project.
- It has great potential for higher customer satisfaction. Between an upgrade and a replatform to Windows, the users will likely be pleased with the combination of increased functionality of the upgrade along with additional flexibility and empowerment from the new platform.
- Customization issues. Your business goals might require additional customization of the application beyond the customizations you completed in your old system. For example, you might want the added functionality of the Siebel Fulfillment Center added to your system for servicing customers. This functionality is available OOB, but was not customized in your original system. It integrates well with Windows products, and you want to use it now. However, making this customization will mean adding more communication servers and other infrastructure changes.

Defining the Replatforming Project Structure

The project structure establishes how the team manages and executes the project and describes the administrative structure for the project team going into the subsequent project phases. It describes the specific responsibilities of each team role and also clarifies the chain of accountability to the customer and designates points of contact that the project team has with the customer. These points of contact can vary depending on the circumstances of the project.

The main function of the project structure is to define the standards the team uses during the project. These standards are discussed in the following sections.

Project Communications

The project structure defines the standards by which team members communicate with one another. Some of these communication standards should include:

- A definition of the reporting structure under which team members operate.
- Procedures to escalate project issues.
- Regular mandated status meetings.

The communication plan should also include contact information for the project teams, including e-mail names, aliases, telephone lists, mail addresses, server share names, directory structures, and other information critical to team organization. Consider establishing a team collaboration environment where communication can occur and progress be monitored and updated as necessary.

Documentation Standards

Clarify the documentation standards for the project by specifying what kinds of documentation it will have, what it will cover, and what formats will be used. You may cover these standards by stating that they are the same as your current regular process standards.

Project Change Control

The definition of change control standards is an important part of structure. These standards can apply to two distinct areas: the solution content and the project documents that describe the content, such as the functional specification. For solution content, change control standards can include prescribed daily builds and the use of source control and versioning software that controls access and solution components and provides the capability to roll back changes as needed.

Change control also needs to be exercised over any changes that could affect the scope of the project. This control is enforced by applying change control to project documents. This method of change control helps ensure that the team is more capable of completing the project on time and within budget.

Replatform projects have an additional source of change that is not obvious, but must be managed: the initial state of the technology being replatformed. This could be source code for applications, or hardware or software configuration and versioning for infrastructure or the middleware. Although you can baseline the initial state when the replatform project is started, operational needs may require changes to that configuration while the replatform is underway. These operational changes are typically subject to configuration management standards imposed by an operations team.

The Program Manager and Release Manager should work with operations personnel responsible for the existing technology to minimize and accommodate these changes.

Instituting change control early on helps the team baseline documentation early in the project and iterate through the Envisioning and Planning Phases to complete the design. Controlling change throughout the process of stabilizing the solution helps in the process of capturing metrics and controlling bugs and issues as the solution nears completion. If your company has a change control process in place, use the same one for the replatform project.

Assessing and Managing Risk

Before proceeding to the development of concrete project plans, a final step in the Envisioning Phase is to identify and compile a list of all risks that team members can anticipate for the replatform project. The team identifies risks by calling on past experience, looking at the current assessment to discover any required changes or

design issues that might cause problems, and thinking through the solution concept to get insights into "big picture" risks.

They create an initial risk assessment document using the process described in the *UNIX Migration Project Guide*, prioritize the risks, and then focus on managing the highest priority risks. For a full discussion of risk, including a detailed explanation of a six-step risk management process, see the MSF risk management white paper, available at <http://www.microsoft.com/msf> in the MSF resource library.

Some common risks for replatform projects include:

- The organization lacks the necessary in-house skills to complete the project.
- Staffing the replatform team with the same people who manage and maintain the current production system can result in the inability to meet both project needs and everyday production needs. This is particularly critical in the Developing Phase.
- A project grows in size and new tasks are added, requiring so much time to complete that the reasons for the replatform become moot. This is known as "scope creep."
- The replatform application fails to meet one or more functional requirements.
- The replatform infrastructure component fails to scale to the required degree.
- The different components in the datacenter and systems in the target environment do not interoperate well after the replatform.
- The system fails during deployment and the team does not have a contingency plan in place.
- There is an unforeseen disaster and the team has forgotten to revise the disaster recovery plan for the new system.
- The strategic implications of the commitment to a different platform are not completely understood, and the business finds that it has made an investment that conflicts with important organizational goals.

The team can significantly improve the odds for project success by reassessing and managing project risk continuously throughout the project. Each team member is responsible for keeping the other team members aware of new risks and of any change in the likelihood of known risks. As the project proceeds from one phase to another, it is likely that some risks can be retired but new ones will be identified. Each team member is also responsible to mitigate and prevent risks where possible. The Program Manager reviews the list frequently and drives modifications of the list.

SAP, PeopleSoft, and Siebel Certified Partners will work with the replatform team in the initial risk assessment process. Risk document templates are available for download on the following Web sites.

- <http://www.microsoft-sap.com>
- <http://www.peoplesoft.com>
- <http://www.siebelonmicrosoft.com>

Templates are also available from Microsoft by e-mailing Plat-LOB@microsoft.com.

Surveying and validating the target environment helps to manage and control risk. For example, the last risk in the list of common risks requires a survey that measures the impact of changing datacenter operations on the components and system relationships in the target environment. The components installed should be compared with the component requirements to validate the target environment. Much of this validation takes place during the Planning Phase, when you plan your physical design and validate your technology.

Closing the Envisioning Phase

Closing the Envisioning Phase requires completing a milestone approval process. The team must document the results of the different tasks it has performed in this phase to enable the project team, customer, and key project stakeholders to commit to the vision and scope of the project.

Key Deliverables for the Envisioning Phase

The following is a checklist for the deliverables and their contents that the team creates during the Envisioning Phase:

- Vision/scope document
 - A vision statement and scope definition.
 - Problem statements and business objectives
 - High-level requirements
 - The solution concept that outlines the approach the team will take achieve the business goals
- Baseline inventory
- Current state assessment
- A project structure document
- A risk assessment document

Major Milestone: Vision/Scope Approved

Reaching the Vision/Scope Approved Milestone concludes the Envisioning Phase. At this point, the project team, the customer, and key stakeholders have agreed on the overall direction for the project, including which features the solution will and will not include and a general timetable for delivery.

2

Planning Your Replatform Project

Introduction and Goals

The Planning Phase of a project is the time when the replatform team translates the initial vision into practical plans for how to achieve it. The team defines the solution in detail:

- What to build
- How to build it
- When it will be built

Most of the Planning Phase activity involves making decisions and documenting them. The full project team, including the certified partner, should be engaged in these activities. The team's goal is to document the solution to a degree that the team can produce and deploy it in a timely and cost-effective manner. The documents are considered living documents and are updated continuously throughout the Planning Phase and beyond, but are subject to change control.

The team works through the design process to create the solution architecture and design; writes the functional specification; updates the risk document, and prepares work plans, cost estimates, and schedules for the various deliverables. Program Management coordinates building a master project plan, which includes plans prepared by team members for areas such as development, test, and deployment, as well as a master schedule based on the plan.

Diligent work in the Planning Phase, which often involves several iterations of plans and schedules, should mitigate risks and increase chances for success. The team continues to identify new risks as they are revealed in the planning process, and it addresses new risks as they emerge. The team must also set up the development and test environments before development can begin.

The Planning Phase culminates in the Project Plans Approved Milestone, when the replatform project team, customer, and key project stakeholders agree upon and formally sign off on the details of the plans.

The major tasks and deliverables for the Planning Phase are listed in Table 2.1

Table 2.1 Planning Phase Major Tasks and Deliverables

Major Tasks	Deliverables
Developing the solution design	— Conceptual design

Major Tasks	Deliverables
	<ul style="list-style-type: none"> — Logical design — Physical design
Completing the technology validation	
Baselining the functional specification	The functional specification, which includes the three views of the solution design
Baselining the master project plan	The master project plan, which includes: <ul style="list-style-type: none"> — Budget plan — Development plan — Test plan — Deployment plan — Training plan — Operations and support plan
Baselining the master project schedule	The master project schedule
Updating the risk assessment document	Risk assessment document that includes risk management plans
Setting up the development and test environments	Development, test, and if needed, staging and conversion environments
Closing the Planning Phase	Sign-off by customer, stakeholders, and project team that Project Plan Approved Milestone has been met

Work closely with your LOB certified partners during the Planning Phase. Their experience in replatform projects will help to produce the deliverables for this phase.

Developing the Solution Design and Architecture

Developing the solution design and architecture begins with a design process that looks at the solution from three perspectives: conceptual, logical, and physical. They are roughly linear, beginning with the conceptual design, but they may overlap,

Conceptual Design

The conceptual design describes the solution in terms that nontechnical people can understand. It captures business needs and user requirements, focusing on what is needed rather than how it will be provided. It clarifies the functionality desired by users within various usage scenarios, including both users and administrators. For example, conceptual designs illustrate each of the user interface elements that must be included in the solution.

The team should consider the needs of all user groups when designing. The business goals and high-level requirements described in the vision/scope document is a good starting point, but the team might need to drill down with users for more details.

The conceptual design shows how the solution will meet the following types of requirements in terms of descriptions that eventually become part of the functional specification:

- Business requirements. This information provides a framework of the business strategic and tactical goals.
- User requirements. This information provides a framework for application GUI design. For example, how many screens are needed, how fast they appear, and what information they contain.

- Usage scenarios. This information provides a framework for the business flow of the application interface and what order the screens appear in.
- Operational requirements. This information provides a framework for how the system is used, for example, whether the system is needed on a 24X7 basis, how fast it should operate, and so on. It would include service level agreements (SLAs).
- System requirements. This information provides a framework for how the system is configured, for example, how many servers, how much memory, and so on.

A replatform project for SAP, PeopleSoft, and Siebel applications is unique because these applications all come with out-of-the-box (OOB) functionality. You can choose or have already chosen in the first implementation the functionality you want to implement based on your needs and what the application provides. For example, the Siebel Sales and Service modules provide "price quotes" functionality, which you would implement if your organization provides price quotes.

To determine your conceptual design, you start by matching your business goals with the prepackaged functionality that the LOB application provides, plus the customizations that currently exist in production. Consider this in hand with your solution concept and also keep in mind the layers (OS, DB, application, or middleware) that you are replatforming. Following are some examples to consider:

- If a business goal is to increase the number of people using Web browser technology for the application interface and to eliminate client-server interfaces, then your conceptual design will include a provision for any changes needed to move to a Web-based application, including such things as adding more Web servers to your system when you move the platform.
- If a business goal is to streamline your sales lead management process, and you want to use the "Leads" functionality provided by the application, include a provision for this in your conceptual design.
- If the OOB functionality provided by the application does not completely meet your business goals, then you must consider customizations. Choosing whether or not to include customizations in the scope of your replatform project has an effect on the design and must be included in the conceptual design.

If your goal for the replatform is solely to reduce TCO, your current production environment can be the basis for the conceptual design.

Logical Design

Logical design describes the solution in terms of the organization, structure, and interaction of its parts. It takes each piece of conceptual design and assigns it to a specific logical role within the enterprise architecture. For this reason, it will reveal any errors and inconsistencies in the conceptual design. Because this is a replatform project, the team should start with the existing logical design if it is available and if it is not, document it in order to be able to use it. Then the team should document the logical design of the replatform system, emphasizing the areas of change. It might be necessary to show how other components outside the scope of the project interact with the subject of the replatform. The logical design provides a foundation for the physical design.

Out-of-the-Box Architecture

The building blocks of SAP, PeopleSoft and Siebel applications begin with their OOB architecture. Your replatform team needs to understand the customizations that were made when you first implemented the LOB application as well as the changes that have been made to maintain the LOB system. Cumulatively, these represent the delta between the LOB architecture and the current state.

SAP, PeopleSoft, and Siebel provide OOB architectures that can be used as the basis for your logical design. Your challenge is to map your original implementation of the LOB applications to the new system.

SAP Heterogeneous OOB Architecture

Figure 2.1 illustrates the building blocks of a SAP System. This is the SAP Web Application Server. Figure 2.2 shows a SAP system with multiple application instances (Web Application Servers) and one database instance. This is the same for every SAP product such as R/3, Business Warehouse (BW), Materials Management (MM), Supply Chain Management (SCM), and Enterprise Portals. One or multiple application server instances plus one database instance make a SAP system.

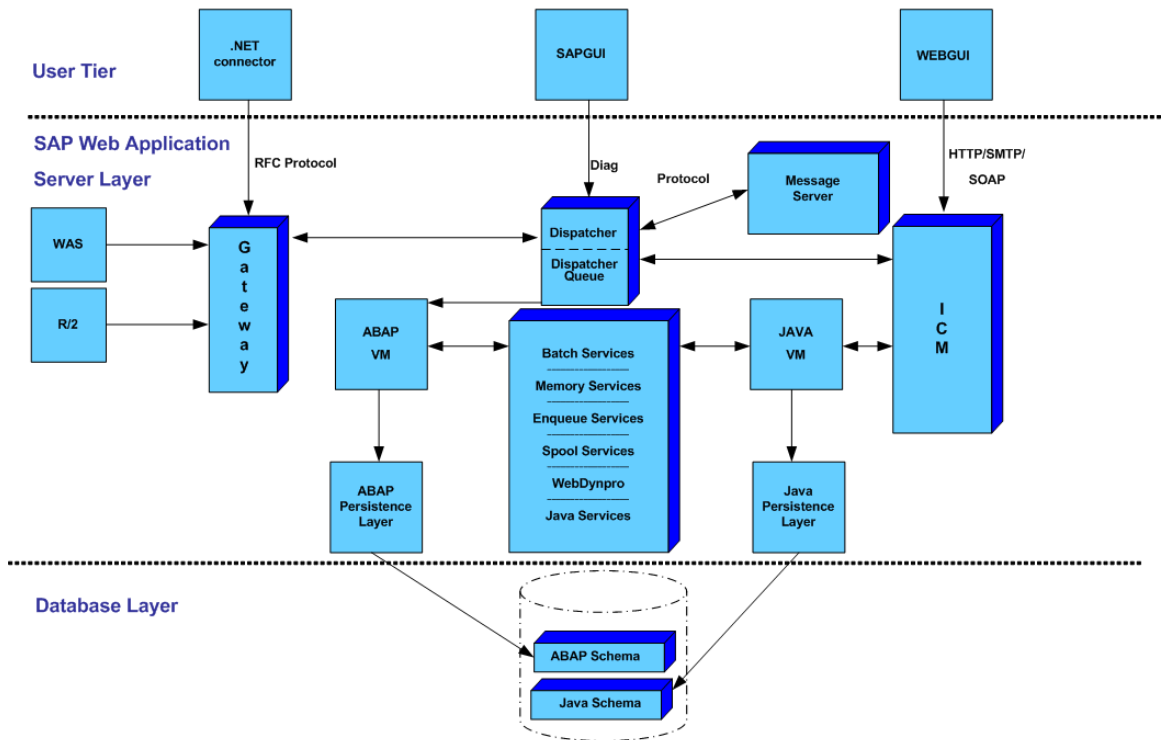


Figure 2.1: Architecture of SAP Web Application Server

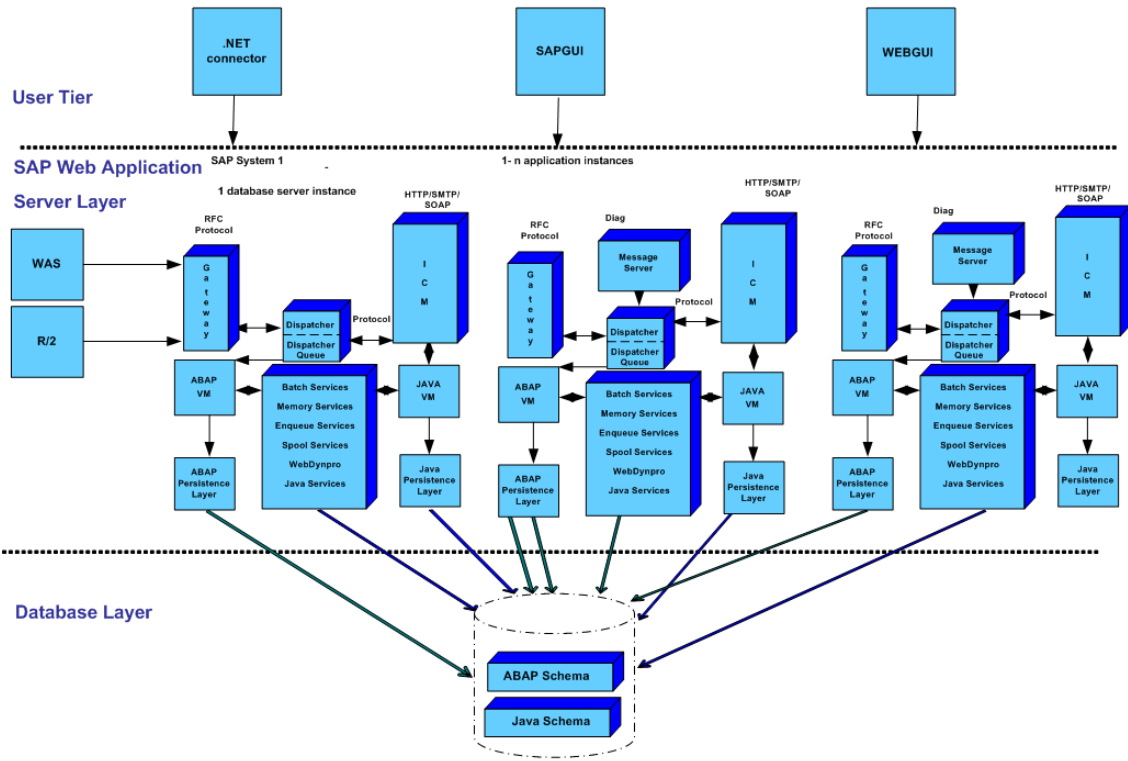


Figure 2.2: SAP system with multiple Web Application Servers and one database instance

SAP supports a number of OOB databases and provides the SAP R/3 Load/Unload tool to move data from one database to another. The vendor Web sites for databases are listed in Table 2.2.

Table 2.2 SAP-Supported OOB Databases

Database Product	Web site
Microsoft SQL Server	http://www.microsoft.com
DB2	http://www.ibm.com
Informix	http://www.informix.com
Oracle	http://www.oracle.com

PeopleSoft OOB Architecture

The PeopleSoft Pure Internet Architecture is a cross-platform, server-based architecture that enables users to work with enterprise applications by using a standard Web browser. This architecture does not rely on client-side code, and is thus able to grant access to PeopleSoft enterprise applications to employees, partners, suppliers, and customers, both inside and outside an organization. This model enables all users to participate in real-time business processes.

The PeopleSoft standard issue architecture is shown in Figure 2.3.

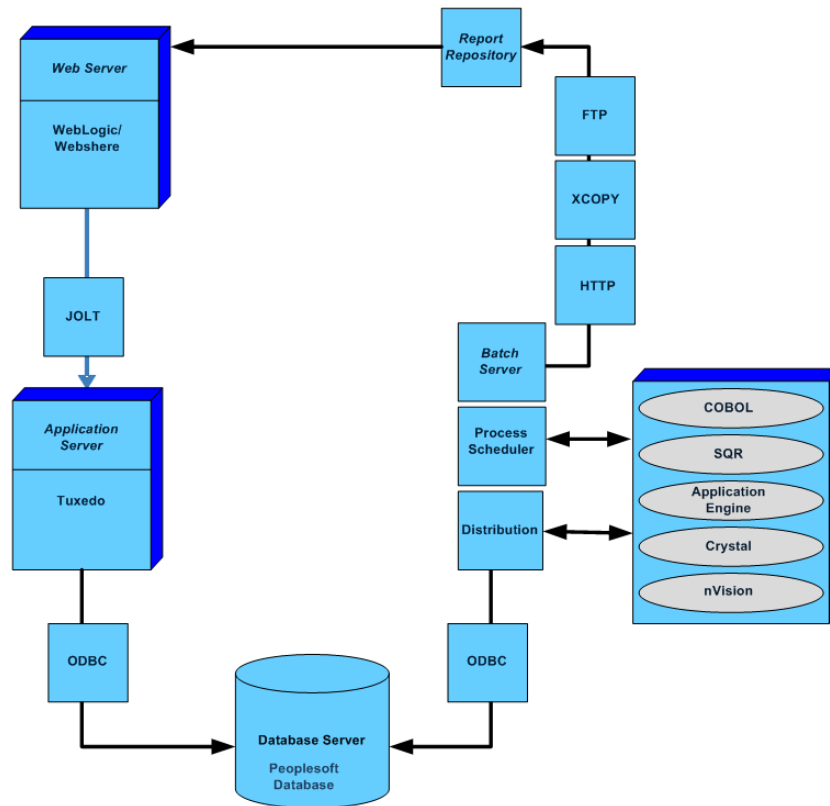


Figure 2.3: PeopleSoft standard architecture

PeopleSoft supports a number of OOB databases. The vendor Web sites for databases are listed in Table 2.3.

Table 2.3 PeopleSoft-Supported OOB Databases

Database Product	Web site
Microsoft SQL Server	http://www.microsoft.com
DB2	http://www.ibm.com
Informix	http://www.informix.com
Oracle	http://www.oracle.com
Sybase	http://www.sybase.com

Siebel OOB Architecture

Figure 2.4 illustrates a Siebel 7 deployment for large implementations showing multiple servers within an enterprise. These servers are usually either load balanced or clustered for high availability.

A more complete description of the Siebel standard product architecture is available at <http://www.siebel.com>

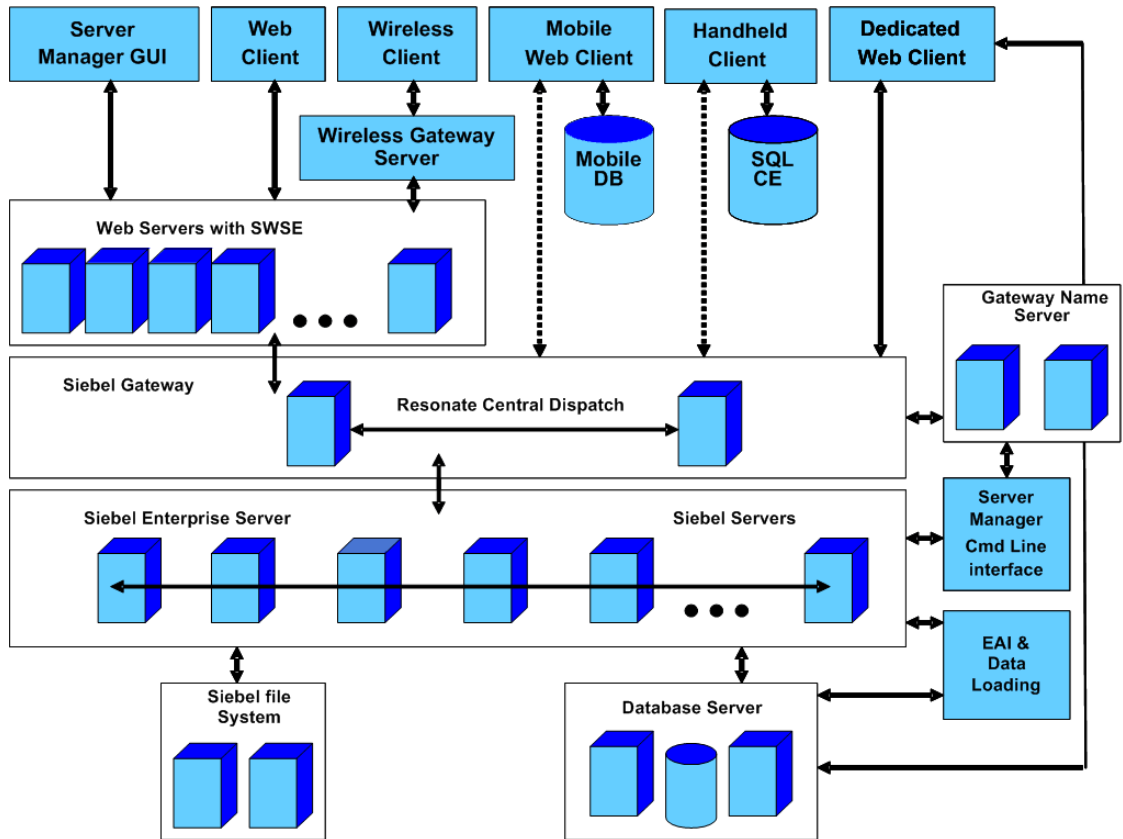


Figure 2.4: Siebel standard architecture

Siebel supports a number of OOB databases. The vendor Web sites for databases are listed in Table 2.4.

Table 2.4 Siebel-Supported OOB Databases

Database Product	Web site
Microsoft SQL Server	http://www.microsoft.com
DB2	http://www.ibm.com
Informix	http://www.informix.com
Oracle	http://www.oracle.com

Physical Design

The physical design view of the solution specifies which logical pieces fit into specific physical pieces of architecture. A logical design could have several possible physical implementations, differing in terms of size, brands of hardware, and so on. It includes a set of physical design models, including a physical database design for the LOB application, components design, and user interface design. It should include anticipated metrics to assess performance and uptime goals. The developers will use the physical design as a blueprint.

The team must determine what physical components are needed to implement the solution. This list will be similar to the baseline inventory of the current system, but it lists the hardware and software that is needed for the new platform, including:

- The number of new application servers.
- The number of disks, or other hardware and devices needed.

The list should also include additional information that you began to gather in your assessment, such as:

- The projected size of the new database.
- Projected memory requirements.
- The projected user base, including the number of concurrent users for each module. Do not count users more than once, and categorize the users according to usage profiles: heavy, medium, light, and self-service.
- The volume of batch, reporting, and query users. Specify how much this activity overlaps with normal OLTP, and if there are specific timing requirements for completing certain batch processing operations.
- Whether any upgrades will be required in the near future, and whether the upgrades should be performed in conjunction with the replatform project.

Important It is critical to determine as early as possible whether the replatforming effort can be executed concurrently with a major version upgrade. Both projects align closely on several key sets of activities; performing both efforts concurrently can save considerable time and costs over separately executed projects.

Contact your hardware vendor and fill out the capacity planning and sizing forms they send you. The certified partners will work with your hardware vendor to create the best Windows architecture for your application; and they will work with your project team to create the best architecture for your organization. Provide information to your certified partner that is as accurate and complete as possible so the certified partner can work with the hardware vendor and plan for growth. Missing information in this area will result in the wrong capacity plan.

SAP Hardware Requirements and Vendors

SAP refers all activities of the replatform to the certified partner and certified BASIS consultant. As a SAP customer you are currently registered with SAP you can review the hardware platforms supported by SAP by visiting the following Web sites and logging in with your user ID and password:

- <http://service.sap.com/platforms>
- <http://service.sap.com/solutionmanager>

For the OS and DB guide, visit <http://www.service.sap.com/osdbmigration>.

Microsoft provides a wide range of tools for putting SAP on the Windows platform at <http://www.microsoft-sap.com>.

For information on putting SAP on a Dell computer hardware platform, visit http://www1.us.dell.com/content/topics/global.aspx/alliances/en/sap_home?c=us&cs=555&l=en&s=biz.

Some of the SAP hardware vendor partners are:

- Bull Corp
Web site: <http://www.bull.com/partners/sap.html>
- Hewlett-Packard Co.
Web site: <http://www.hp.com/large/promo/68053/>
Web site: <http://h71028.www7.hp.com/enterprise/cache/54665-0-0-225-121.aspx>

- IBM Corp (Windows Server™)
Web site: http://www-1.ibm.com/partnerworld/pwhome.nsf/weblook/pat_sol_sap_edu_desc.html
- NCR Corp
Web site: http://www.ncr.com/media_information/2004/jan/pr012604.htm
- Siemens Corp
Web site: http://www.siemens.com/index.jsp?sdc_rh=null&sdc_flags=null&sdc_sectionid=0&sdc_secnavid=0&sdc_3dnvlstid=&sdc_countryid=0&sdc_mpid=0&sdc_unitid=999&sdc_conttype=2&sdc_contentid=206518&sdc_langid=1&sdc_pnid=&

PeopleSoft Hardware Requirements and Vendors

Frequently recommended platform vendors include:

- Dell:
Web site: <http://www.dell.com/peoplesoft>
e-mail: peoplesoft_sizing@dell.com
- Hewlett-Packard:
Web site: <http://www.peoplesoft-hp.com>
e-mail: peoplesoft@hp.com
- IBM:
<http://www.ibm.com>
- Unisys:
<http://www.unisys.com/products/>

Siebel Hardware Requirements and Vendors

Determine whether your organization purchased a Siebel Technical Support Package. If you have this package, your Siebel Technical Account Manager (TAM) may already have information on your organization's recommended capacities and projected growth.

If this information is not readily available, contact Siebel Expert Services to receive a Capacity Planning Questionnaire. A well-researched, accurately completed questionnaire will provide the hardware, disk space, and CPU specifications you need to maintain the current system and plan for the projected growth.

You can then plan your hardware purchase accordingly and have a clear understanding of what your system will look like on a Windows platform. The questionnaire results will also help you plan your development, test, stage, and production environments.

Visit this site for a list of Siebel hardware partners:

<http://www.siebel.com/partners/psearchresultssol.asp?psol=138>

Validating the Technology

Often in parallel with the design process, the team validates the technologies being used in the solution. During technology validation, the team evaluates the products or technologies to ensure that they work according to specifications provided by their vendor and that they will meet the business needs for the specific solution scenario.

Creating the Functional Specification for the Solution

The functional specification is the technical description of the solution. It should be fairly detailed, and represents the contract between the customer (the business sponsor for the project) and the project team about what the project will deliver. The Development Role, which is tasked with "building to specification" for the team, uses it to guide its work. The team as a whole uses it to build project plans and schedules. The Program Management Role takes the lead in creating it, with input from the role leads regarding their areas of responsibility.

Important Recall that the Program Management Role includes two demanding functional areas — project management and solution architecture — which may be filled by different people because of the different skillsets required. The architecture function is primary for this task (the collective knowledge of the development sub-team may be tapped as well). You may choose to have the certified partner fill the architectural responsibilities. If not, it is highly recommended that you ensure that the functional specification is reviewed and signed off by your SAP, PeopleSoft, and Siebel certified partners.

During the Planning Phase, the functional specification is baselined and placed under change control.

Because it includes all requirements documents and design specifications, the functional specification contains the information required to design the solution. It incorporates the conceptual, logical, and physical designs, and should include the following components:

- A summary of the vision/scope document as agreed upon and refined, including background information to place the solution in a business context.
- Any additional user and customer requirements beyond those already identified in the vision/scope document. For example:
 - Security requirements. Specify the strength of security to be used for things such as transactions, including a description of any encryption standards to be used.
 - Legal requirements. Document any legal requirements, including what needs to be done to adhere to these requirements. For example, LOB applications for healthcare might be required to meet certain customer scenarios, governmental standards, or business policies.
- The solution design. Make sure this covers the complete set of planned features for the solution.
- Specifications of the components that will be part of the solution.
- Quantitative measurements whenever possible. For example, database capacity, concurrent user capacity, performance metrics, and so on, should be clearly stated.

Note If your project includes no new functionality, and you want only to replatform the existing system, your functional specification will be relatively simple.

When the functional specification is complete enough for customer and stakeholder review, the team baselines the specification and begins tracking changes. This document is maintained as a detailed description of the various solution components and how the solution will look and operate. After it is formally approved at the milestone meeting that

marks the end of the Planning Phase, the functional specification can be changed only with customer/business approval.

Creating the Master Project Plan

After you have validated the solution and prepared a functional specification for it, create a project plan. The six most important individual plans that make up a replatform project are:

- Budget
- Development
- Test
- Deployment
- Training
- Support/operation

The Budget Plan

The budget plan determines the cost of the project. Work closely with your certified partner and hardware vendors to develop as accurate a budget as possible. Areas such as hardware, WAN circuits, software, consultants, contractors, personnel, training, communication, travel, and even entertainment for the team are all cost areas that must be factored into this plan. Document any assumptions you make in arriving at the estimates and remember to account for capitalization, depreciation, and expenditure factors.

The Development Plan

The sub-team that fills the Development Role creates the development plan, which details the replatform build process. It describes different aspects of the development endeavor, such as tools required, methodologies and practices to be followed, a schedule of events, and resources. The primary tasks that the development plan might describe are:

- Contacting the vendors for hardware
- Installing the hardware in the data center
- Porting application customizations
- Installing the OS
- Installing the DB
- Testing the installation of the OS and DB software
- Building the disaster recovery system

The development plan goes into detail about the work that the development sub-team will perform in coordination with testing, and operations. Whereas the physical design provides a blueprint of how the completed replatform solution will look, the development plan documents the process development will follow to build it.

The plan covers the development schedule — when and how the build process occurs, who is responsible for each task, and what resources will be used. It also identifies the standards for development in the organization, such as server naming standards and formats. The more detailed the development plan is, the easier it will be to coordinate the development effort.

The Test Plan

An LOB replatform test plan should address these topics:

- **Scope.** The test scope addresses what will and will not be tested.
- **Test strategy.** The test strategy defines any specific strategies for testing performance, quality, TCO, and other benchmark requirements. Review the test strategy from the original LOB implementation and reuse it wherever possible. The test strategy should map directly to the business goals.
- **Test cases.** A replatform test plan must include test cases that specify:
 - Setup test data.
 - Step-by-step instructions for test execution.
 - Desired results.

Reusing the test cases that exist in the current system documentation is a good practice to follow. Plan for each component that is affected by the replatform process to be tested

- **Testing resources.** The testing resources include the teams and people who perform the testing (development team for unit tests, test team for all other testing), and the Super Users Group for user acceptance testing.
- **Test schedules.** The test schedule defines when testing occurs and who is involved. It also defines the number of cycles the test team will execute during testing. The test phases are the number of cycles the test team will execute during testing.
- **Test procedures.** Include the testing procedures, walkthroughs, and setup of test data and test scenarios.
- **Tools and methods.** List the tools and methods you will use for testing. One of the most commonly used tools is Mercury TestDirector. More information about TestDirector is available on the Mercury Web site at <http://www.mercury.com/us/products/quality-center/testdirector/>
- **Reporting schedule.** Specify the types of and schedule for the reports that the test team will deliver.
- **Test categories.** A detailed list of the kind of testing you plan to perform.

Note For an SAP replatform, follow the special SAP Testing Guidelines that are a part of an OS/DB replatform.

Test Categories

Depending on your system requirements, you might perform one or more of the following types of testing. It is essential in a replatform project to run a complete battery of tests on the new platform as described by the categories in the following list. The reason for such exhaustive testing is that, depending on your scope, a number of things might have changed and the impact of the changes will not be apparent till testing. It is normal software development life cycle (SDLC) to test all possible points of impact for a change. The different types of testing are:

- **Unit testing.** Guarantees that individual components of the solution perform as expected. In a replatform project, the development sub-team during the Developing Phase examines each of the database objects as well as the code embedded in stored procedures, and makes sure triggers have been transformed accurately.

- **System testing.** Authenticates the functionality of the system as an integral whole. Testing and development must be done in parallel wherever possible or pipelined at a more granular level to finish the project on time. In a replatform project, system testing includes testing the application connection to a SQL Server database, and testing that the application can process data correctly. This is done primarily by Test during the Stabilizing Phase.
- **Stress/load testing.** Evaluates system performance and reliability under practical stress levels prior to implementation in actual production. Stress testing checks database coexistence issues that might occur due to a high level of activities carried on in parallel. Some problems can be discovered only when the system is under load. Test does this testing during Stabilizing.
- **Utility testing.** Tests utilities used for such purposes as scheduling, monitoring, and backups. The Development sub-team does some of this testing during the Developing Phase, and Test repeats the tests and verifies the results during Stabilizing.
- **Security testing.** Testing the security related to deployment ensures that only authenticated users are able to access to nonpublic data. The network path must be secure to prevent hacking. In addition, the security domain hosting the database must accurately validate and approve user requests.
Mapping the UNIX users and groups with the SQL Server users and roles must be verified. Testing also checks that the permissions and privileges are not augmented. After initial tests by Development, Test does this testing during Stabilizing.
- **High availability testing.** Ensures that the disaster recovery plans for the system are available and work as required. Production systems that use clusters must be tested for their failover functionality, while systems using log shipping must be tested for their recovery and reconnection to the log-shipped server. If the cluster is based on a Storage Area Network (SAN) or a third-party solution, those aspects must be tested as well. The developers who install and maintain the infrastructure that will be deployed to production do the initial testing during the Developing Phase. Then Test does a final round of testing during Stabilizing.
- **User acceptance testing.** Confirms that the business requirements have been met. These tests enable the users and support personnel to work with the new technology and understand its use. Users must be able to test the application under circumstances similar to the actual production process.
- **Regression testing.** Tests that a program has not regressed: that is, in a commonly used sense, that the functionality that was working yesterday is still working today. A set of test cases must be available that can be executed as a regression test. Each time the test is run the results must be logged and compared with the expected and previously obtained results. Use baselines from the current environment as a measure for success. The Test sub-team does all of the regression testing.

Important Although the test plan will be rolled up into the master project plan, which should be available to all members of the project team, check to ensure that the test plan is always available to the test sub-team.

The Deployment Plan

The deployment plan covers the deployment strategy, contingency plans, and support, as well as any tools or automation required. It also defines the strategy for preparing the end users and operational personnel before and during the deployment.

Planning the deployment includes the following steps:

1. Defining the goals of the deployment process.
2. Describing the type of deployment strategy that will be used, such as a straight cutover, parallel cutover, or a serial cutover, and describing the steps for it.
3. Planning and allocating resources for the deployment.
4. Defining roles and responsibilities for each resource — the deployment team as well as the operations team.
5. Defining and planning the rollback/fallback strategies.
6. Setting up a deployment schedule, including establishing/lifting code freeze and cutover times. This will need to be reviewed for possible needed updates, depending upon how long Developing and Stabilizing take.

How you plan your deployment depends upon your solution concept and which layers you are replatforming. For example:

- If you are deploying a replatformed OS layer, then your deployment plan needs to include a count of all the servers that need to be shut down and steps to install the Windows OS and bring the servers back online in a timely fashion.
- If you are deploying a replatformed DB layer, then your deployment plan needs to cover upgrading the DB server to the Windows OS, the installation of the new database, moving the data from the old database to the new one, and testing the database server.
- If you are deploying a replatformed application layer, then the deployment plan needs to cover shutting down the application servers, installing the Windows OS, installing the application software, compiling the EXE on Windows, testing the Web servers, and so on.
- The middleware deployment plan might include plans for installing a product such as Microsoft BizTalk Server and testing all of the interfaces and services for interoperability between different systems.

Deployment Tools

See the description of tools in the section "Setting up the Development and Test Environments later in this chapter. The procedures that you created using these tools are executed during deployment.

Code Freeze and Cutover

Deploying a replatform involves two time periods that affect the entire organization: code freeze and cutover. Release Management will need to coordinate the timing of these two periods with the business to accommodate the organization's needs and expectations.

The two periods are:

- **Code freeze.** The state and time at which all further programming, enhancements, and bug fixes are stopped in the current production environment.
- **Cutover.** Switching from the old (hardware and/or software) system to the new system. Cutoff marks the point in time when the new system is live and the old system has been shut down.

Communicate the timing of the two periods to the organization as early as possible. In some organizations it might be necessary to explain the concepts as well. Business users might need to be educated in the concepts of code freeze and cutover and take some time to understand what the replatform is and why the organization must wait on any new enhancements and bug fixes until the replatform is complete. Communications

responsibility might be shared between the Product Management Role (customer advocate) and the User Experience Role (end user advocate).

Typically the replatform team is developing the new environment while the operations team is maintaining production and supporting the day-to-day business on the present system. There is a point in the project when the business managers and IT managers must decide together when there needs to be a code freeze. A code freeze means that there cannot be any more changes, bug fixes or enhancements going into the production system for a specified amount of time. For a smooth and stable environment for replatform, the development and test teams need a system that is not subject to change during the freeze. Program Management, along with Development and Test teams, determine the duration of the code freeze. A code freeze typically lasts anywhere from a week to a month, depending on how the period is negotiated, and what service-level agreements (SLAs) are in place for the SAP, PeopleSoft, or Siebel systems.

During the cutover period, the old system is shut down and data is moved from the current production environment to the new environment. Ideally this takes place over a weekend and this document will refer to this period as "cutover weekend." However, cutover can often take longer than a weekend, in which case it is better to build a system and then switch it over as production. No matter how long the cutover period, cutover is the point in time when no more live production data is in the old production system. When the replatform goes live, all the data is in the new database. There are a few cutover approaches to consider.

Cutover Approaches

There are three cutover approaches you can take for deployment: a straight cutover, a parallel, or a serial cutover. Each approach has advantages and disadvantages. For SAP, PeopleSoft, and Siebel applications, the most common approach is the straight cutover. This is usually the easiest, and can typically be done over a weekend. Work with your staff and your certified partner to decide which approach works the best for your organization and replatform project.

For each approach, formulate a rollback plan if things don't work after moving to the replatformed system. A rollback plan enables the system to roll back to a consistent and stable state. Try your workaround strategies or contingency plans before rolling back to the original database. The rollback approach you implement depends on the cutover approach you use. The rollback approach must account for any transactions that were executed after the start of the deployment. To select the optimal rollback approach, create a list that documents the possible failures.

Straight Cutover

A straight cutover of the system replatforms the system from one environment to another, without taking any intermediate steps. This approach is used:

- When downtime availability does not limit the replatform.
- When the database affected does not have important data that must be accessible at all times.
- When a favorable opportunity is available for replatforming, for example, during a holiday.
- When multiple systems are tightly integrated.

The disadvantage of this approach — especially with large databases — is that replatforms that take advantage of rare business holiday opportunities might be jeopardized by failures and delays. To minimize the risk, the hardware might be duplicated to keep the database server running, or a verified image of the database

server can be maintained so that it can be restored with minimal problems in case a rollback occurs.

Rollback after a Straight Cutover

In this case, the UNIX environment needs to be frozen and a backup made at the time the cutover started. If a rollback occurs, it is possible to restart the UNIX environment with minimal effort. A rollback might require reestablishing broken sessions between the client computers and the database.

Parallel Cutover

A parallel cutover approach maintains both the old and new system in parallel, with the database replicated in the new environment. This approach is the most expensive but running two systems in parallel for a period of time is a way to mitigate the risk of any problems in the new system. A database is created in the new environment and data is moved on a continual basis. A process is put in place to replicate changes in the current database to the new database synchronously or asynchronously.

Replication is an approach for organizations that do not have adequate downtime available to replatform and move all the data to the new database.

It has several disadvantages, including:

- It requires adequate database server resources to run two parallel database installations together on different platforms.
- It requires a uniform mechanism to ensure database availability between the two data sets.
- Data integrity and validation is more complex.

Rollback after a Parallel Cutover

The rollback process involves reversing the direction of the replication, which means moving from the new SQL Server database back to the old database. Consequently, you must keep the old database updated during the period of the cutover.

Serial Cutover

A serial cutover approach enables you to replatform in phases. This approach is usually implemented when:

- There are multiple servers and databases that must be replatformed.
- Users of each database are independent of the other.
- Data within the database is partitioned based on business function.
- The time required for the replatform is limited to short periods.

The disadvantage of the serial cutover approach is that it is difficult to implement when the database is complex and interdependent with other infrastructure elements.

Rollback after Serial Cutover

The rollback approach is similar to that in a straight cutover with the entire replatform viewed as a collection of smaller cutovers. The serial cutover enables you to roll back to a previous consistent state without undoing all changes. It also enables you to solve problems in small increments.

Reducing Downtime

One of the key elements of deployment planning is working with operations to accurately estimate how long the entire deployment takes. Make sure that you include the time required to do backups in your estimate. Figure this downtime into the overall schedule

just as you account for business system availability requirements and maintenance schedules. Estimate deployment downtime based on the size and complexity of the solution. Use prototypes in these estimations.

Look for ways to lower the impact of the deployment by selecting a deployment time such as a holiday period, or a time when the business workload is low. Coordinate this time with an appropriate cutover strategy, moving several components in parallel, and consider any other options appropriate to your deployment period.

After you know the downtime for production, make sure you communicate the downtime in advance to the organization.

The Training Plan

The replatform training plan should address two basic types of training needs: training required for people who are implementing the solution and training for those who will use it. In order to define and document objectives, the training plan must identify the resources, processes, and skills that the user needs to use the system efficiently. It must describe the training delivery vehicles, such as online tutorials and classroom training that will be provided. Finally, it should specify the materials and resources, such as reference manuals and online help, that will be provided for each training audience.

- **Project team.** Some members of the project team that is performing the replatform process might need to address gaps in their knowledge or skills through training to enable them to fulfill their project responsibilities.
- **Users of the replatform solution.** This group includes internal and external users. Internal users, in turn, include IT operations staff who will operate the new environment as well as end users of the application. Operations staff may be participating on the project team, so training for project staff and operating the new platform should be scheduled separately to enable the individuals who overlap to participate in both types of training.

External users are end users outside the organization, such as external customers and suppliers, who might also use your LOB application. Form a team to train these end users to use the solution. The training can cover areas such as usage and maintenance of the solution.

Note Business users may need to be trained as administrators on some functions that IT operations normally performs, such as adding new users to the system.

It is highly recommended that a Microsoft technology expert work with the super users in an organization who are learning the Windows system to explain the platform differences and demonstrate the ease-of-use advantages Windows provides. The super users can then help train the rest of the organization.

Siebel provides training called "Train the Trainer." The concept is that one person is trained as the expert, and then that expert trains the rest of the people in the organization or the team. This training method can be very effective if the super users group is also involved.

The training plan should include a training schedule, which must be incorporated into the overall project schedule.

Training Resources

In general, most training for end-users in replatform projects focuses on training that trains end users in the new environment and how users will perform tasks differently.

Advanced training is also available for the professionals who will work on the replatform project. The following lists identify resources that you may want to use for your training.

Microsoft Technologies

Microsoft offers a variety of courses for professionals who support, implement, and develop solutions using Microsoft technologies. Training options are available in course, workshop, clinic, seminar, and online format.

- Microsoft Windows Training.
<http://www.microsoft.com/windows/training/default.aspx>
- Microsoft SQL Server.
<http://www.microsoft.com/sql/techinfo/training/default.asp>
- Microsoft Operations Manager.
<http://www.microsoft.com/technet/traincert/virtuallab/mom.aspx>
- Microsoft Official Curriculum (MOC). Microsoft offers training and certification courses designed to educate customers on how to optimize Microsoft products and technologies.
<http://www.microsoft.com/learning/default.asp>
- Microsoft Certification Training. Microsoft offers certification courses for professionals who have the skills and knowledge to successfully implement a Microsoft product or technology as part of a business solution in an organization.
<http://www.microsoft.com/learning/mcp/default.asp>
- Online Courses. Microsoft offers an online training option for learning about Microsoft products and technologies. These courses are designed for professionals who are unable to attend onsite training sessions.
<https://www.microsoftelearning.com/>
- **Instructor-Led and Computer-Based Training.** Microsoft offers both classroom-based and computer-based training courses for experienced UNIX administrators and developers who want to gain a thorough understanding of Windows as compared to UNIX or Linux systems.
<http://www.microsoft.com/windowsserver2003/migrate/unix/unixproresources>

LOB Technologies

See the following links for specific product training:

- **SAP Training**
http://www.microsoft-sap.com/coming_events.asp
- **PeopleSoft Training**
<http://www.peoplesoft.com/corp/en/training/index.jsp?site=corp>
- **Siebel Training and Workshops**
<http://www.siebelonmicrosoft.com/mssiebel/ComingEvents.aspx>
<http://www.siebelonmicrosoft.com/mssiebel/Technology/Workshops.aspx>

The Operations and Support Plan

The objective of the operations and support plan is to document how the solution will be operated and supported. The plan specifies how the operations team identifies and resolve user-related problems in the new environment. The plan must cover issues such as:

- Monitoring the LOB application on the new platform.
- Alerts that warn of problems with the application on the new platform.

- Issue logging, which is used to record and track issues as they occur.
- Issue resolution, which is a process that is established to resolve issues.
- The escalation process, which is a process that escalates issues until they are addressed and resolved.
- Change management, which is used on the new platform for continued maintenance.

The plan should identify processes and appropriate resources for each of these functions. The support plan should document the current operational state. The plan can be used to analyze the gaps in personnel knowledge and skills.

This plan must also include the SLA, written to function as a contract between the provider of the service and the customers. The SLA should specify how the organization's help desk provides support to the user population with respect to every interaction with the new replatformed environment.

Project Schedule

The detailed project schedule, like the master project plan, will require input from all of the team roles, who must calculate how long it will take them or their sub-team, such as development or test, to complete the tasks assigned to their role. It can be organized by project phase. Depending on the size of your project, available resources and partners, and the size of the system, you can expect each of the phases to last on average as follows:

- Three to four weeks for the Envisioning Phase.
- One month for the Planning Phase for limited replatforms such as replatforming the database layer. More extensive projects that replatform several layers will take longer.
- Two to three months for the Developing Phase.
- One month for the Stabilizing Phase.
- About a month for the Deploying Phase The Deploying Phase includes one weekend to one week for the actual deployment, and another two to four weeks to fix any issues under the covered warranty period. (See Chapter 5 for a discussion of the covered warranty period.)

Typical SAP OS/DB replatform projects take about ten to twelve weeks. Typical PeopleSoft replatform projects take about three to six months. Typical Siebel replatform projects take about six to eight months.

Factors that can have a significant impact on the project schedule and change these sample durations, include:

- If the replatform project includes replatforming the middleware layer, the project timeline will increase significantly, depending on the enterprise architecture, how many interfaces are involved, and how complex the interfaces are.
- For Siebel, if a large number of smart scripts are used, a rewrite of the scripts is required and the time it takes will depend on how much script must be rewritten.
- If the application uses hard-coded values, this will increase the duration of the project. You can check for hard-coded values and be forewarned during your proof of concept.
- If the replatform project includes changing the database layer in addition to the application layer, the duration of the project will be longer.

SAP provides a *specific* project schedule for OS and DB replatform projects that you can use when planning your replatform project. Siebel and PeopleSoft do not provide similar schedules, but you can work with Microsoft and your certified partners to develop a schedule.

Microsoft, SAP, PeopleSoft, and Siebel all provide *sample* project schedules. These replatform project schedules are generic. They are by no means complete or intended to be followed exactly. Study these project schedules and modify them as appropriate for your organization, industry, and the project management methodology that you plan to follow for your replatform effort.

A sample SAP project schedule is available from your SAP Certified Partner; or from Microsoft at plat-SAP@microsoft.com.

A sample PeopleSoft project schedule is available from Microsoft by request at plat-Psf@microsoft.com.

A sample Siebel project schedule is available at either of the following Web sites:

- <http://www.siebelonmicrosoft.com/mssiebel/Downloads/PDF/scid=kb-en-us-323950.pdf>
- http://www.siebelonmicrosoft.com/mssiebel/Downloads/Application_Replatforming/SiebelDBMig/GenericMigrationPlan.mpp

Setting Up the Development and Test Environments

Changing the operating system is a major step in the replatform effort. You need to set up development and test environments on Windows first, so the application and development tools are installed on Windows-based servers. Make sure the application is configured and compiled on a Windows-based computer before it is tested. Otherwise, if you replatform the OS only in production, there is no parallel environment to replicate bugs and test fixes.

The development and test environments are based on the configurations that support the planned work and software tools with respect to the replatformed databases and applications. In addition to development and test environments, replatform projects also often use staging, and conversion environments. Following is a brief explanation of the environments you will need for developing and testing the solution before deploying it to the production environment.

- **Development environment.** The development team uses the development environment for application development. The development environment requires a connection to a development database, but the language and tools that the team use depend on the application they are developing: either SAP, which uses the ABAP language; PeopleSoft, which uses People Tools; or Siebel, which uses Siebel Tools. The development database is used to develop and unit test the applications without affecting the rest of the organization. The number of development databases required will depend on the size and needs of the development team.
- **Test environment.** The test environment is where all business-case testing and end-to-end-testing happens after the development team sends a final build to the test team. The test environment usually has the full production-size database so all different scenarios can be thoroughly tested.
- **Staging environment.** The staging environment is an environment used in most organizations to test any production issues if the test environment is busy with the test cycle. Stage is usually an exact copy of the production environment. Staging is sometimes also used for performance, stress/load testing.

- **Conversion environment.** For huge data conversions or moves, a separate conversion environment might be required so that data is not overwritten in the development environment. If conversion routines must be written and tested, there is a chance that test data might be corrupted by the conversion routines. This environment is crucial in a replatform when you must move production data from one system to another without corruption.

Sometimes there is a separate performance tuning environment that is an exact copy of production.

Hardware

These environments usually have multiple servers to support the application. For example, most IT departments supporting these LOB environments keep the test and staging environments as close to production as possible, which requires multiple servers and network considerations to support the system properly. When building the environments, you must build, configure, and test the physical network and servers.

Software and Tools

The nature and configuration of the development and test environments will be dictated by the environment (the features and tools it includes) and the components that are being replatformed. Software and tools specific to the replatform that are required might include:

- **Development Tools**
 - Software development tools that come with the product you use, such as: Siebel Tools, People Tools, or SAP ABAP/BASIS.
 - Database modeling tools such as the Computer Associates data modeling tool, AllFusion ERwin Data Modeler:
<http://www3.ca.com/Solutions/Overview.asp?ID=260&TYPE=S>.
 - Software management tools (source control, build, bug tracking, and so on), such as:
 - Serena Merant Version Manager
<http://www.serena.com/products/ecm/vm/home.asp>
 - Mercury TestDirector (
<http://www.mercury.com/us/products/quality-center/testdirector/>
- **Test Tools**
 - Test tools for monitoring, measuring, recording, and reporting in the test environment, such as the Mercury tools:
 - LoadRunner for load testing
<http://www.mercury.com/us/products/performance-center/loadrunner/>
 - WinRunner for regression testing
<http://www.mercury.com/us/products/quality-center/functional-testing/winrunner/>
 - Additional information provided by Mercury at the following Web sites:
 - Peoplesoft solutions
<http://www.mercury.com/us/solutions/erp-crm/peoplesoft/>
 - SAP solutions
<http://www.mercury.com/us/solutions/erp-crm/sap/>
 - Siebel
<http://www.mercury.com/us/solutions/erp-crm/siebel/>

- **Database Tools**

Consider what data-moving tools you want to use for deployment. In the following list are some suggestions and Web links to more information:

- For more information about data migration, visit http://www.informatica.com/solutions/technology/data_migration/default.htm.
- For an SAP DB replatform, consider whether to use the SAP R3 Load/Unload tool and move the data from a homogeneous to a heterogeneous environment. For more information on this tool and on data mapping from multiple sources visit <http://www.informatica.com/solutions/technology/sap/default.htm>
- For a PeopleSoft replatform consider whether to use the PeopleSoft Data Mover tool or use Microsoft Data Transformation Services (DTS) or a combination of both.
- For a Siebel replatform, explore whether to use Siebel Enterprise Integration Manager (EIM) or to map the relationships and use the Informatica Extract Transform and Load (ETL) process. For a one-time data load you might be able to simply dump the data into SQL Server, but if data cleaning and repetitive loads will be required in the future, Informatica ETL is a much better, farsighted approach, and can be used for any data move.

For more information, visit

<http://www.informatica.com/solutions/technology/Siebel/default.htm>

- Third-party solutions tools to move from Oracle to SQL Server, such as DB Best products
<http://www.dbbest.com/>

Files such as the source code of database access routines, as well as supporting scripts, configuration data, and documentation should be made available to the development environment.

Testing the Environments

After the development test, staging, and conversion environments have been constructed, conduct a series of tests to ensure that the developers have access to all required tools and facilities. These tests should simulate the developers' day-to-day activities. Also test for the presence of the data sources and applications to be replatformed.

For more information about setting up the test environment, refer to the *UNIX Migration Project Guide*.

Closing the Planning Phase

Closing the Planning Phase requires completing a milestone approval process. The team documents its planning work in three major documents: a full set of functional specifications, the master project plan, and the master project schedule. It submits them along with an updated risk assessment to key stakeholders and customers for review.

Key Deliverables Checklist for the Planning Phase

A deliverables checklist for the Planning Phase includes:

- Functional specification
- Master project plan

- Master project schedule
- Updated risk management documents
- Development and test environments are set up

Major Milestone: Project Plans Approved

The Project Plans Approved Milestone is the time when the customer, stakeholders, and the core project team all signal their approval of the project plans and their agreement that the plans are sufficiently complete for the team to begin development.

The functional spec, master project plan, and master project schedule form the project baseline and provide the basis for making future trade-off decisions. The baseline should be placed under the change control procedures that were outlined in the project structure document. This does not mean that all decisions reached in the Planning Phase are final. But it does mean that as work progresses in the Developing Phase, the team must formally review and approve any suggested changes to the baseline by using a change control process.

3

Developing Your Replatform Solution

Introduction and Goals

This chapter provides an overview of the tasks required to perform the actual replatforming of the LOB application by using two examples: the development of PeopleSoft and SAP DB replatforms. In order to assure continued product support, SAP, PeopleSoft, and Siebel require that one of their certified partners or certified consultants perform the majority of the work involved in these tasks. In fact, not all of the information required to perform these tasks is publicly available. The overview is intended to provide enough context to enable your team to efficiently and effectively support the efforts of your selected partner or consultant, work with them, and manage them.

The focus in this chapter is on the database layer. The reason for this focus is that the database layer is the one that organizations most often replatform. A large majority of SAP, PeopleSoft, and Siebel implementations already have the application layer running on Windows Server and are using Internet Explorer as the Web browser for access to the application. Also, a common reason to replatform is to move to a database that better suits your organization's needs. If you determine that SQL Server best suits your organization's needs, then to support SQL Server, it is mandatory that you replatform on Windows.

The primary goal of the Developing Phase is to build the solution components: the infrastructure for the new platform, the OS replatform, the DB replatform, the application replatform and, if applicable, the middleware replatform as well as the documentation for each layer. Development might involve code development or modification, or database changes, which requires software developers or DBA support, respectively. The Development sub-team should also build a disaster recovery system for the new platform during this phase.

Although the Development Role (sub-team) is the primary team role driving the Developing Phase, all project team roles are active in building and testing deliverables. Release Management, for instance, might be working on rollout checklists and updating rollout and pilot plans and Test will be creating test cases. See the UMPG for a table showing the focus of each role during this phase.

The tasks and deliverables that relate to the primary focus for the Developing Phase are listed in Table 3.1.

Table 3.1 Developing Phase Major Tasks and Deliverables

Major Tasks	Deliverables
Starting the development cycle	
Building a proof of concept	Proof of concept
Developing the solution components	OS, DB, Application or Middleware Layer
Building the solution	A completed replatform build
Closing the Developing Phase at the Scope Complete Milestone meeting	Sign-off by customer, stakeholders, and project team that the milestone has been met

Starting the Developing Phase

The Developing Phase can be the most challenging and yet satisfying part of the replatform project. Major issues are likely to emerge soon after this phase begins. The degree to which your team can resolve issues in a satisfactory and efficient manner determines whether your schedules or funding requirements need to change.

Addressing Risk in the Developing Phase

Proactive risk management is crucial during the Developing Phase because it represents the pinnacle of potential chaos in the project's life cycle. Managing risks during this phase begins with updating the status of risks on your risk list and ensuring that risk management plans are current. You may be able to retire some risks, but will also need to add new ones.

To identify new risks, first review the task lists from the Envisioning and Planning phases to discover the points of high risk in the Developing Phase. If any item on these lists is not completely accounted for, it could present itself as a risk during this phase. Then review the following text to help identify layer-related risks.

Common Risks for Each Layer

There are some common risks that arise for each layer in a replatform project. While some of these risks cannot be addressed until deployment, preparing for them during the Developing Phase will ease the transition to the new platform later on. The following sections cover some of these risks by layer.

The OS Layer

The following problem related to the OS layer may occur if your organization is undergoing a complete replatform change from UNIX to Windows and your staff doesn't already include someone with Windows expertise.

- New Windows updates or service packs are released, and your staff does not know how to install these across your server environments (development, test, and so on). A common mistake is to leave the needed training until after the replatform effort. The mitigation is to start the training well ahead of when you deploy the solution.

The DB Layer

The most significant risk is that of undetected data conversion errors. There are two major forms of errors that might avoid detection during the conversion process:

- When you are pulling data from multiple sources, the individual source fields must be correctly mapped to the appropriate field in the target schema. If the mapping is

incorrect, the conversion process itself cannot detect that data is being placed in the wrong fields or wrong tables.

- Subtle details of the platform or database can have an impact on the conversion of individual data items. This occurs most commonly with "date" data: Julian date discrepancies, date range differences, and so on. The conversion process must be carefully monitored to ensure that converted data items make sense rather than just being syntactically valid.

Some of these problems can be addressed in advance; or by working with your database experts, they are easy to fix during the Developing and Stabilizing phases. You can use DB Best tools (<http://www.dbbest.com>) for some of the conversions to fix some of these problems; however, often you must check the data in person to make sure it is converting correctly.

The Application Layer

Common problems that can occur when replatforming the application layer are:

- The parameters or security settings for single sign-on that are tested during Developing and Stabilizing are not set correctly by operations during deployment. This makes the deployment environment different from test in a way that violates the assumptions of test. One mitigation strategy for this risk is to ensure that operations personnel are involved during development and testing.
- There are too many factors that affect response time to be able to test them all during development and testing, so the response time in the production environment is not fast enough. The mitigation is to set customer expectations accordingly and set aside a few days after deployment to tune for response time on the new platform in production.
- A user profile is not mapped correctly. The spec is correct but a mistake is made during implementation. Because user profiles cannot be ported (they must be set up by the application administrator), a recommended mitigation is to have Test verify the production setup.

Your organization's application administrator can work closely with the replatform test team to catch these kinds of mistakes.

The Middleware Layer

Common problems that can occur when replatforming the middleware layer are:

- A connection point is missed in the assessment of the old system and no provision was made for it, so, in the new system, a file that must be sent between applications doesn't work, resulting in a business loss. The mitigation is to have team members exercise additional care in the initial assessment.
- The online, batch, or real time interface is not fast enough in live production, though it was satisfactory in the development and test environments. Again, as for the application layer, set customer expectations accordingly. Some amount of performance tuning always happens in the real production environment, especially if the full production load or data could not be simulated.

These items sometimes have to be addressed after the deployment.

Building a Proof of Concept

The proof of concept (POC) tests whether the solution concept is viable for the organization by using actual customer data or other aspects of their current

implementation. The POC tests key elements of the solution on a nonproduction simulation of the proposed operational environment.

MCS and your certified partners will be heavily involved in a proof of concept (POC) for SAP, PeopleSoft, or Siebel. The replatform POC could be a continuation of some initial development work (a preliminary POC) that occurred during the Planning Phase or when your organization made the decision to replatform. Walk the team, operations staff and users through the solution to validate their requirements.

An example of a POC is as follows: An organization that wants to replatform the DB layer of an LOB system must be certain that SQL Server can process the same 10 million invoices per night that their current system processes. The POC proves that the new technology can handle this business requirement. The POC has additional value because you will discover how long it takes to implement a representative part of the solution. This will confirm or call into question your earlier decisions about whether to use a phased approach or a "big-bang" approach for the deployment cutover. All aspects of the current environment such as infrastructure, application, database, user profiles, and documentation must be assimilated in the proof of concept.

There might be some solution or documentation that carries through to the eventual solution development deliverables, but the POC is not meant to be production-ready. The POC is considered "throw away" development that gives the team a final chance to verify functional specification content and to address any issues prior to transitioning into development.

After completing a proof of concept, the team should have a good idea of the LOB system implementation — most likely a combination of a standard software package from SAP, PeopleSoft, or Siebel, plus a full list of customizations installed specifically for the organization. Completing the POC marks the point where the team is fully transitioning from conceptual validation to building the solution architecture and components.

Developing the Solution Components and Building the Solution

Developing is organized according to the same four-layer organization you used to define the solution concept.

The application layer is the only layer for which you need to create an executable. Each developer performs unit tests on their portion. It is a best practice to do daily builds for this development process. Build validation tests on each build expose unanticipated design and integration defects and make their diagnosis easier.

If you are replatforming the DB layer, you will be writing conversion programs to do data loads. It is recommended that you do daily loads on the database to build and test the load process.

Building the OS layer involves installing the operating system and setting up the parameters correctly to work with the SAP, PeopleSoft, or Siebel application. If there are any changes to the configuration of the OS, these need to be identified for the Test sub-team and operations.

As noted earlier, most solutions do not involve the middleware layer as part of the initial replatform project. However, every time there is a change in the OS, DB, or application layer it has downstream impacts on the middleware layer, so the interfaces need to be tested with each build.

Database Layer Replatform Examples

The remainder of this chapter focuses on the database layer for replatforming. The following two sections provide high-level examples of replatforming the DB layer for PeopleSoft and SAP applications.

Developing a PeopleSoft DB Replatform

This example of a replatform of a PeopleSoft system running Oracle database to SQL Server describes the essential development steps at a high level. The example is intended only as a roadmap to help you understand the build process for a replatform project. For actual build steps contact your certified partner or Microsoft at plat-Psf@microsoft.com.

At this point you should have all the following accomplished or available:

- All hardware set up
- The Windows OS installed
- All documentation from your planning sessions
- A model of what your architecture should look like on Windows

Note that the PeopleSoft schema is divided into two sets of tables.

- **Tools tables.** These tables contain the application logic, reference tables used by the application, and metadata tables that describe the objects in the database and the application. The tools tables are analogous to the SQL Server system tables that reside in each database.
- **Data tables.** These tables are the majority of tables in the schema that hold customer data relevant to the deployed components of PeopleSoft

The tools tables must be moved first during the installation and replatform of PeopleSoft, so the schema can be generated by PeopleSoft. Any customer modifications to the schema can be included. Modifications might include new tables, modified tables, and new or modified indexes.

Once the scripts have been generated from PeopleSoft and executed into the SQL Server environment, the resulting database schema is compatible with that of the source system, and is ready to receive data from one of several transfer methods.

Process Overview

The following diagram provides an overview of the replatform steps the development team should follow. These general steps are required to complete a replatform of a PeopleSoft database from another platform to SQL Server. Each step is described in detail in the sections that follow, and gives a general idea of things that happen during the Developing Phase for a DB replatform.

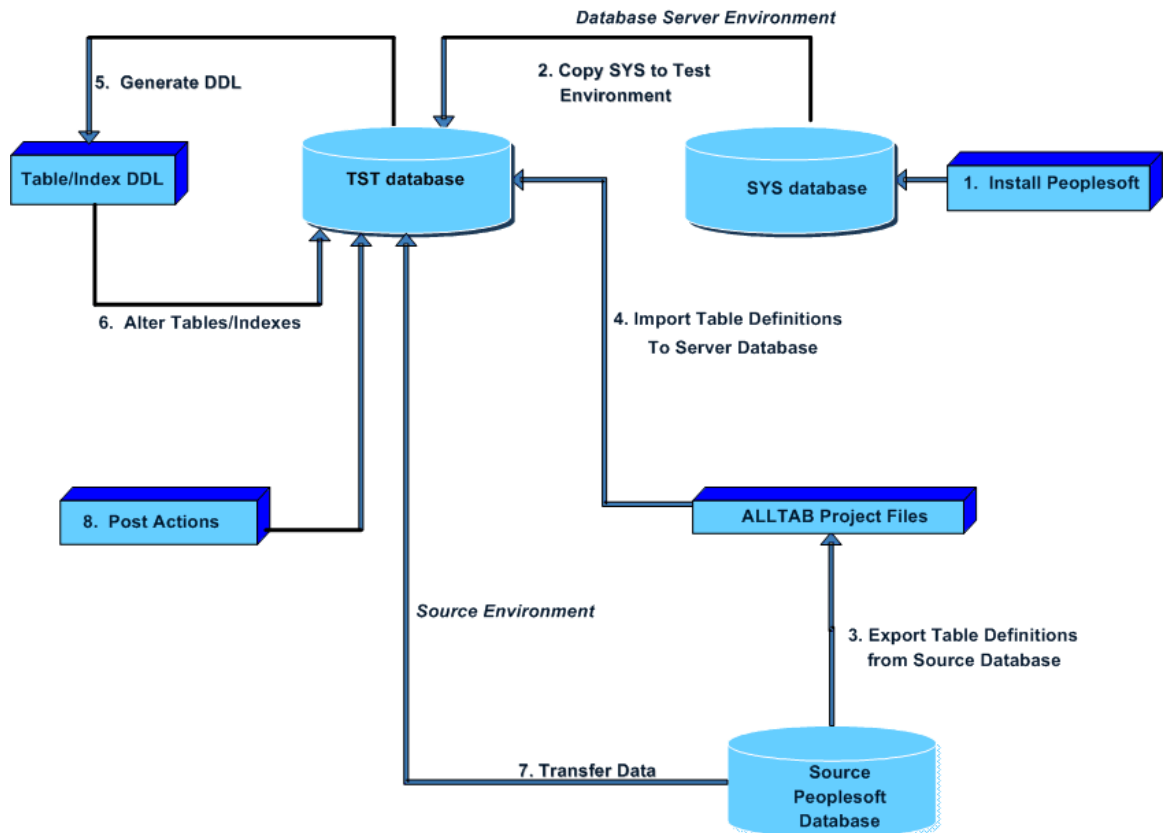


Figure 3.1: Replatforming PeopleSoft Database to SQL Server Database Environment

1. Install PeopleSoft

Note The installation of PeopleSoft should be conducted by a trained PeopleSoft administrator on your development sub-team and ideally, is under the supervision of a certified PeopleSoft partner or PeopleSoft consultant.

Refer to the People Tools Installation Guide for SQL Server for the installation background and requirements. The installation guide for PeopleSoft 8.4 is available on the PeopleSoft support Web site as file name PT84Install_MSS.pdf. The PeopleSoft support Web site is available at:

- <http://www.peoplesoft.com>
- <https://www.peoplesoft.com/corp/en/login.jsp>

Log on using your PeopleSoft customer ID and password.

Create the Initial Databases

The installation creates two initial databases in the selected SQL Server instance:

- A “demo” database that contains a small amount of demonstration data. This database is useful for testing patches and upgrades from PeopleSoft prior to deployment.
- A “SYS” database in which all tables are defined, but only the tools tables are populated. The SYS database is used as the starting point to create a test database on SQL Server that is used to start replatform development activities.

Note Naming conventions vary, but it is common practice to preface the database with the module of PeopleSoft that is represented in the database. For example, an HR system might have the names HRDEMO, HRSYS, HRDEV, and HRTST for the databases.

2. Copy the SYS Database to the Test Database

After installation, create a database where testing activities occur. You might call this database DEV or TST.

Creating this database is a matter of “cloning” the SYS database, using either the SQL Server backup and restore functions, or copying the SYS database files and attaching them under a new name.

When completed, the DEV or TST database is identical to the SYS database.

Note Make sure to observe your company's naming conventions and make certain that both internal and external names are updated in the process of copying SYS to the DEV or TST database. It is common to embed the database name in these file names and they should be altered accordingly.

3. Export Table Definitions from the Source System

This step takes the table definitions, including customer modifications, from the source PeopleSoft schema, and copies them to a series of files that are subsequently imported into the newly created database.

In PeopleSoft terminology, a “project” is a collection of objects. The OOB PeopleSoft product has predefined projects; as the customer makes modifications to the source application, additional projects are created to manage new or changed objects that must be tracked and implemented throughout the PeopleSoft environment. These are the customizations that you noted and documented when performing the baseline system inventory during the Envisioning Phase.

Move the project of table definitions from the current PeopleSoft system to the SQL Server PeopleSoft environment. As you do this you can modify the standard schema so you can copy data to it.

Creating a Table Definition Project

One of the standard projects in PeopleSoft is the ALLTABS project; it includes the internal definitions for all tables in the PeopleSoft schema. It also includes related objects such as columns and indexes.

It is possible that your ALLTABS project is not up to date with recent table modifications. To avoid missing these modifications, update the ALLTABS project or create another project that contains all standard and custom table definitions.

This is where the code freeze of your current production system is critical to the replatform. When you communicated the code freeze all your projects should have been updated.

Exporting the Project

A development team member exports the new ALLTABS project from the current PeopleSoft database.

4. Import Table Definitions to SQL Server

A development team member then imports the new ALLTABS project into the new PeopleSoft database created in Step 2.

5. Generate DDL

This step generates the scripts necessary to alter the database schema from the base PeopleSoft definition to include any customer modifications to tables and indexes. Rather than executing the DDL interactively, save the DDL to a script file for subsequent execution through Query Analyzer. Save table and index definitions separately; defer index creation until after the data migration is completed.

6. Alter Tables and Indexes

Use Query Analyzer in the PeopleSoft database to execute the script created in the previous step. When completed, the table structures in the destination database are compatible with those in the source database from which the data will be copied.

Execute the script containing the recreation of the tables only; wait to execute the script containing index DDL after the transfer process is completed.

Note The scripts created by the PeopleSoft Application Designer contain SET IMPLICIT TRANSACTIONS ON and COMMIT statements throughout. In this step it is not of any consequence, but for some future activities you might choose to make a mass change to remove these. Use caution when doing so: several PeopleSoft columns contain the word "COMMIT" and so mass changes must be made carefully!

7. Transfer Data

Make sure the table schemas for your current environment and the new SQL Server environment match. Now the transfer of all data between the environments can take place.

In theory you could omit the transfer of the tools tables that were already synchronized between the two environments; the export/import procedures described earlier have ensured that some tools tables — those that describe database objects — contain the same data.

In practice it might be more effort and risk to try to isolate these tables from the following transfer process than is worth the small amount of time saved. It is recommended that you transfer all tables.

There are several approaches to transferring the full contents of the PeopleSoft database from the current platform to SQL Server:

- For very small environments, the PeopleSoft tool Data Mover was designed to move data between your old and new databases. However, Data Mover is impractical for moderate to large volumes. If the database is in excess of 10GB, use another method.
- Data Transformation Services (DTS) can be employed to transfer the largest tables in the PeopleSoft database. A DTS high speed data pump can transfer at a much faster rate than PeopleSoft Data Mover. One approach with DTS is to select a subset of the largest tables in the PeopleSoft database and create a DTS package for each; performance might be slightly better if transfers occur with one object per package, as opposed to multiple objects per package.
- Another alternative for using DTS for transfer is to engage MCS Database Center of Excellence (COE) to provide custom DTS solutions to complete the transfer of all table data in the PeopleSoft database. Custom solutions provided by MCS

Database COE consultants have recorded transfer rates between 50 and 70 GB per hour. PeopleSoft databases 300 GB in size have been transferred by this method in under six hours.

Replatform Steps

1. Make a backup of the database that you are preparing to populate. The database at this point is small and the backup should be complete in a few minutes.
2. Execute the transfer procedures using Data Mover, DTS, or a combination of the two as determined by volume.
3. Once the transfer has been completed, execute the index creation script through Query Analyzer to build indexes on the data tables in the new PeopleSoft database.

8. Post Actions

After the transfer of data is complete, you must perform additional steps to make the new PeopleSoft database ready to use. Internal PeopleSoft security must be synchronized with SQL Server security (there are only a couple of IDs to be concerned with here), tables containing Uniform Resource Locators (URLs) specific to the environment, and so on. PeopleSoft administrators will understand these follow-up steps.

Two PeopleSoft audit procedures must be executed at this stage: run DDDAUDIT and PSAUDIT against the new environment to report inconsistencies that might have resulted from an incomplete migration process.

Complete the process with an additional database backup at this time.

Building an SAP DB Replatform

This example of a replatform of an SAP system running any supported database to SQL Server provides a high level overview, guidance, and tips for the development of the replatform. SAP provides specific step-by-step guidance about replatforming the database layer for SAP applications. During the Envisioning and Planning Phases you probably worked with your certified partner to determine all of your project steps. If not, contact your certified partner, SAP, or Microsoft (plat-SAP@microsoft.com) for the information you need.

SAP is one of the few ISVs that provide an actual tool (R3 Load/unload) for moving a database to a new platform. SAP as a company recognized that because their product runs on a number of databases, a customer might need to move a database to a different platform. SAP then created a well-defined process and tool to fulfill that need. SAP also provides a very specific set of checkpoints during the development and testing of a replatform.

When preparing for a DB replatform, remember to:

- Use the R3load SAP tool.
- Get a Migration Key from SAP. You can generate a Migration Key using the SAP System Marketplace at <http://www.service.sap.com/migrationkey>

Note The SAP OS/DB Migration Guide clearly states that no special customer support can be requested of SAP during the replatform. SAP guarantees, however, to fix any software errors in the SAP replatform tool.

Preparing for an SAP Replatform Using System Copy

Before you begin with the practical system copy tasks, you must make a number of fundamental decisions that influence the subsequent system copy procedure. Where possible, make a system copy to build a test, demo, training, or standby system. Perform upgrades in the test system first so you can identify problems that might result from custom modifications.

Note For Oracle and Informix, standby systems cannot be created with a system copy.

To replatform the database layer for SAP, you must make a SAP system copy and follow the process instructions of the SAP Note (see reference in the first bullet), combined with direction from your SAP certified partner. For your preparation steps and replatform steps, make sure that you:

- Have the most recent version of the SAP Note about copying SAP systems based on SAP Web AS 6. You can find SAP Notes at <http://service.sap.com/notes>.
- Have your SAP certified partner on hand to step you through the process.
- Write out a plan for the system copy with the help of your certified partner and appropriate personnel from your project team.
- Check with your SAP certified partner for the naming conventions you should use for the copy.
- Ensure that the SAP system release of the source and target systems are the same.
- Can use an SAP system copy for setting up system landscapes (where the SAP systems have different SAPSIDs) and for creating systems for testing, backup, and training.
- Have the correct version of the SAP installation kit before starting any copy procedures.
- Have the SAP version compatibility required for the copy operation.
- Decide which type of system copy you want to perform: homogeneous or heterogeneous.

There are two ways to perform a homogeneous system copy: by restoring a backup of the source DB to the target DB, or by unloading the source DB and then loading the data into the target DB. Consult with your SAP certified partner and the SAP documentation for detailed information and to obtain the proper tools for the job.

SAP Replatform Preparation Steps

Following is a generalized list of steps that you must take when preparing for an SAP replatform:

1. **System Copy:** Make sure that all required CDs for the system copy are available. You no longer require a special migration CD. The tools are located on the SAP Installation Master CD.
2. **Tool Version:** Check that you have the appropriate tool versions for your SAP Kernel.
3. **SAP License:** Once installation is complete and the SAP system copy has been imported, you will require a new license key for the target system. The license key of the source system is not valid for this system. You can order a new license key for the target system from the following sources:

- SAP Service Marketplace
<https://www.service.sap.com/migrationkey>
 - SAPNet R/3-Frontend
 - Telefax
For more information look at SAP Note 94998 at <http://service.sap.com/notes>.
4. **Archive Files:** You must make data archived in the source system (data that does not reside in the database but was moved to a different storage location using SAP Archive Management) accessible in the target system. Adapt the file residence information in the target system. Refer to the SAP Online Documentation (SAP Library → Cross-Application Components → Archiving Application Data) for help. Access to archive files is platform-independent.
 5. **Test Run and Schedule:** Perform a test run of the system copy. Use the time taken by the test run to calculate the system downtime:
 - If your target system will replace your source system, try to perform a complete test run, meaning that the entire database is exported from the source system, transferred to the target system, and imported there. System downtime will be roughly equal to the total test time (that is, time for export, transport, and import).
 - If you do not want to replace your source system, a partial test run (export of entire database or parts of it) might suffice to calculate the system downtime. The source system will be down only for the time of the export.
 - Calculating the system downtime is particularly important for very large databases (VLDB) or when tapes are used. The test run is also done to determine the amount of export data. Perform read/write actions only on local file systems. Do not use NFS-mounted file systems for these reasons:
 - Reading from NFS-mounted file systems might fail.
 - Writing to NFS-mounted file systems might cause corrupted dumps.
 6. **Test and Final Replatform:** After you have completed the basic system copy operation, refer to the SAP documentation under "Final Activities" to perform the final steps. Define a schedule for the test and the final replatform.

Preparing for an SAP Replatform without System Copy

The SAP system infrastructure (development, test, and production system) can be set up without making a system copy as follows:

1. Install all SAP systems (begin with the development system).
2. Customize the development system as described in the implementation documentation.
3. Transport the client-dependent and client-independent data to the test and production environments.

An alternative method of setting up the SAP system infrastructure is to install a system, customize it and then perform a system copy.

When copying a system that contains production data, choose an opportune time to make the copy, such as a month-end or year-end closing. Make sure that there is a well-defined starting point for the data in the new system.

Running the SAPinst to Export the Database

Before you export the database, shut down the SAP system. The database, however, must still be running to avoid inconsistencies on the target system.

Run SAPinst to export the database of your SAP system. Consult the SAP documentation about R3Load procedures to perform this part of the process.

You can also perform a remote installation using a standalone SAPinst GUI on a separate Windows or UNIX host. This enables you to perform the installation on a remote host while monitoring it with SAPinst GUI from a local host. To perform a remote installation, follow the SAP documentation for R3load procedures under the section titled, "Performing a Remote Installation with SAPinst."

After the data has been moved to the new system apply the migration key to use the new system.

For information on troubleshooting, see SAPinst troubleshooting guide at SAP service marketplace:

<http://service.sap.com/sapinstfeedback>

Testing the Replatform Solution

The testing process for an LOB replatform solution is designed to identify and address potential solution issues prior to deployment. Testing starts when solution development begins, and it ends when the testing team certifies that the solution components meet the project plan's goals for schedule and quality.

When building software, the development team designs, documents, and writes code that it then tests through unit testing and daily builds. When building hardware or network infrastructure, the team may be developing scripts, conversion tools, hardware configuration settings, network topologies, and other similar tasks. The team tracks development progress against a project schedule, resolves reported bugs, and documents test results.

The test team designs and documents test specifications and test cases, writes automated scripts, and runs acceptance tests on components submitted for formal testing. This team assesses and reports on overall solution quality and feature completeness, and certifies when the solution features, functions, and components meet the project goals.

Note SAP has very specific guidelines for testing. The SAP OS/DB Migration Check prepares your system for a smooth replatform ending with high availability and high performance of your new operating system and/or database. There is a checklist, as well as a review process that needs to happen three to four weeks in advance of going to production. For more information about Migration Check, visit <http://www.service.sap.com/osdbmigration>.

The second service session is performed on the new target system approximately four weeks after the start of production.

Closing the Developing Phase

Closing the Developing Phase requires completing a milestone approval process. The team needs to document the results of the different tasks it has performed in this phase to give stakeholders (including the customer) the information they need to sign off on the completion of development.

Key Deliverables Checklist for the Developing Phase

A deliverables checklist for this phase includes:

- Source code and executables
- Installation scripts and configuration settings
- Frozen functional specification
- End-user Help and training materials
- Test specifications and test cases (coverage tests)
- Test and bug reports from the Test Director tool
- Updated risk management document

Major Milestone: Scope Complete

The Developing Phase culminates in the Scope Complete Milestone. At this milestone, all functionality has been completed and the solution is ready for external testing and stabilization. This milestone is the opportunity for customers and users, operations and support personnel, and key project stakeholders to evaluate the solution and identify any remaining issues to address before beginning the transition to Stabilizing and ultimately to release.

Project teams usually mark the completion of a milestone with a formal sign-off. Key stakeholders, typically representatives of each team role and any important customer representatives who are not on the project team, signal their approval of the milestone by signing or initialing a document stating that the milestone is complete. The sign-off document becomes a project deliverable and is archived for future reference.

4

Stabilizing Your Replatform Solution

Introduction and Goals

The purpose of the Stabilizing Phase is to improve the solution quality to a level that meets the originally specified acceptance criteria for release to production. Although unit testing was conducted as the solution was being built, testers conduct testing during this phase on a replatform solution whose features are complete. Stabilizing Phase testing emphasizes usage and operation under realistic environmental conditions. The team focuses on prioritizing the bugs that Test discovers, with an emphasis on stability and reliability, fixing those with the highest priority, and preparing the solution for release. Test may develop new tests cases to reproduce issues reported from other sources and integrate them into the test suite.

While Test and Development are discovering and resolving bugs, Product Management is communicating the upcoming changes to the organization, User Experience is stabilizing user documentation and training materials, and Release Management is finalizing deployment plans. Once a build has been deemed stable enough to be a release candidate and pre-production testing is complete, the team deploys the solution to one or more pilot groups in the staging environment. Your pilot group can be the recommended Super Users Group discussed in chapter 1.

The CCB is extremely involved during stabilization and might meet daily to review all the bugs in the system to resolve them as soon as possible. The critical factor to remember during stabilization of a replatform project is that even if you replatform only one of the four layers, you still need to test all them to make sure that the new production system works end-to-end when it is deployed for live production.

Once a build has been deemed stable enough to be a release candidate, the solution is deployed to a pilot group. This phase culminates in the Release Readiness Approved Milestone, indicating team and customer agreement that all outstanding issues have been addressed.

The major tasks and deliverables for the Stabilizing Phase are listed in Table 4.1.

The tasks and deliverables that relate to the primary focus for the Stabilizing Phase are listed in Table 4.1.

Table 4.1 Stabilizing Phase Major Tasks and Deliverables

Major Tasks	Deliverables
Testing the solution	Bug list and test report
Resolving defects	Updated bug list and test report
Conducting the pilot in the staging area	Pilot user signoff
Building the solution	A completed replatform build
Closing the Stabilizing Phase at the Release Readiness Approved Milestone meeting	Sign-off by customer, stakeholders, and project team that the milestone has been met

Testing the Replatform Solution

For your LOB system to be stable enough to release to production, Test must ensure that the system works as a whole by testing the system end-to-end from both the system and the business perspectives.

During Stabilizing the Test sub-team ensures that all components that were developed work as designed. Test also ensures that the system works as a whole. As described in the test plan in Chapter 2, "Planning Your Replatform Project," Test conducts the following types of testing during this phase: system testing, utility testing, security testing, and high availability testing. The three most important types of testing done by Test during Stabilizing are stress/load testing and regression testing. It also coordinates user acceptance testing. The following sections cover these important types of tests.

Stress/Load Testing

System testing and stress/load tests are done to ensure that performance goals of the replatform project are attainable. The stress/load tests are usually conducted in the staging environment because this environment is most nearly identical to the production environment.

One of the major concerns in a platform change is that Windows and SQL Server are able to meet the business requirements for security and reliability. If it is a business requirement that the LOB system be up and available 99.999 percent of the time, then the system needs to be designed to have redundancy and be tested for availability to meet that requirement.

Regression Testing

For replatform projects, regression testing is the most important class of tests from the system perspective. Regression testing tests previously tested components and functionality of the system to make sure that they function properly even after a change has been made to parts of the system.

Be careful about bugs that already exist in the old production system. Problems will come up during regression testing that testers might classify as bugs arising from the new platform, but regression testing might show that these bugs already existed in the current production environment.

Report these bugs and check them against the current production bug list. Bring them up at the CCB meetings so that the entire team is aware of them.

User Acceptance Testing

The Super Users who also comprise your pilot group are the best group to perform user acceptance tests, which represent the end-to-end testing from the business perspective. This is done after the completion of functional and regression testing, and is the final stage in the testing process. It involves testing the system with data supplied by the actual user or customer rather than the simulated data developed as part of the testing process. Three or more cycles of user acceptance testing might be necessary before business users will sign off that the solution is ready for release to the production environment.

User acceptance testing also gives support personnel and users the opportunity to understand and practice the new technology through hands-on training. The process helps to identify areas where users have trouble understanding, learning, and using the solution.

Finally, user acceptance testing also gives Release Management the opportunity to identify issues that could prevent successful implementation.

Testing Criteria

The following sections discuss the different criteria Test takes into account when deciding whether the implementation is ready for release.

Pass/Fail Criteria for Test Cases

A test case passes if the actual result meets the pass criteria. If not, the test case is treated as failed.

Pass criteria are as follows:

- All processes run with no unexpected errors.
- All processes finish in an acceptable amount of time, based on benchmarks defined in the functional specification.
- Load tests show that a satisfactory level of capacity is in place and that the system can be scaled out when necessary.

If a test case fails, do not assume that the feature is defective. For example, a failure could occur because of misinterpreted, incomplete, or inaccurate project documentation. Analyze each failure to discover its cause, based on actual results and the results described in project documentation.

Test Suspension Criteria for Test Execution

The test team might suspend partial or full testing activities on a given build if any of the following events occur:

- A systems engineer is unable to install the new build or a component of it.
- A systems engineer is unable to configure the build or a component.
- A major feature has a fault that prevents a significant area from being tested.
- The test environment is not stable enough to trust the test results.
- The test environment is very different from the expected production environment, so test results cannot be trusted.

Replatform Resumption Requirements

The test team can resume testing if:

- The problem that suspended the testing is corrected.
- The development and test teams agree that fixing the bug can be postponed until the next iteration.

Replatform Release Criteria

Many organizations define release criteria in terms of the severity of open bugs. Severity is often measured on a scale of 1 to 4, where level 1 represents the highest severity, and level 4 represents the lowest.

The following are some possible criteria for allowing the code to be released to production:

- There are no open bugs with a severity of 1 or 2.
- Test cases scheduled for both integration and system test phases have successfully passed (especially test cases that cover major functionality or that were given a high priority or a medium priority).
- The final regression test has passed successfully.

Note The criteria for release vary widely, depending on both the architecture and the nature of the replatformed application.

Reporting the Results

After the test team completes a successful execution of the required test cases, prepare a test report to list the remaining open bugs. This final test report is called the Release to Production report and should contain the following:

- The areas that were tested and a summary of the test results.
- List of bugs or issues that were postponed to future versions.
- Description of projected impact on business if the remaining bugs are not fixed.
- Recommendation on whether to proceed with production.

Lifting Code Freeze — A Major Decision

After the bug count has come down to an acceptable level, with no high priority bugs, it is a good time for the CCB to consider lifting the code freeze on the old production environment. While the new system is being built by the replatform team, the development team for the old environment, in its development lab, has been continuing to make changes that fix problems in the LOB application that is currently in production. If the CCB votes to lift the code freeze, the replatform development team will incorporate the customizations into the Windows-ready version of the application. However, introducing these changes might require another round of testing.

Lifting the code freeze represents a major decision point that should involve the business decision makers, and a "go" decision needs to be communicated to the organization. A reason to lift the code freeze is that the business will want fixes and new functionality that have been put on hold to be brought into production as soon as possible. Although lifting the code freeze will add complicating factors and thus increase project risk, the business

may have an urgent need for a particular fix and argue successfully for a selective lifting of the code freeze.

A reason not to lift the code freeze is that the fixes can always be applied in a separate release after the replatform goes live. Normally, most organizations would wait until after their new production system is running, and after the production warranty has expired, to do another release and add more functionality to the newly replatformed LOB system.

Conducting the Pilot Program

During the pilot, the team tests as much of the entire solution, in a true production environment, as possible. A pilot release is a deployment to a subset of the live production environment or user group. Depending on the context of the project, the pilot can take various forms:

- In an enterprise, a pilot program is a group of users from different areas of the business.
- For replatform projects, a pilot might involve testing the most demanding application or database that is being moved with a sophisticated group of users that can provide helpful feedback.

The common element in all these forms of piloting is testing under live conditions.

Following Best Practices

Prior to beginning a pilot, the team and the pilot participants must clearly identify and agree upon the success criteria of the pilot. These should map back to the success criteria for the replatform solution effort.

Also, before the pilot is started, you must have a support structure and issue-resolution process in place, such as a helpdesk for the new platform application. This may require that support staff be trained. The procedures used for issue resolution during a pilot might vary significantly from those used during deployment and when in full production.

Any issues identified during the pilot must be resolved by one of these methods:

- Further development.
- Documenting resolutions and workarounds for the installation teams and production support staff.
- Incorporating resolutions and workarounds as supplemental material in training or help materials.

Deciding Next Steps

After enough pilot data has been collected and evaluated (pilots are typically run for one month), the team is at a point of decision. One of several strategies must be selected:

- **Stagger forward.** Deploy a new release to the pilot group.
- **Roll back.** Execute the rollback plan and revert the pilot group back to the previous configuration state they had before the pilot (as closely as feasible). Then try again with a more stable release.
- **Suspend.** Suspend the entire pilot.
- **Fix and continue.** Issue a fix to the existing code to the pilot group.
- **Proceed.** Advance to the Deploying Phase.

Creating Pilot Test Reports

As cycles of the pilot tests are completed, the team prepares reports detailing each lesson learned and how new information is incorporated and issues are resolved. The following might result from performing the pilot testing:

- Identification of additional risks.
- Identification of frequently asked questions for training purposes.
- Identification of user errors.
- The ability to secure buy-in and support from pilot users.
- Documentation of concerns and issue resolutions.
- Updates to documentation, particularly help files and deployment plans.
- Determination of whether or not all success criteria were met.

Release Readiness Checklist

Use the following checklist of questions, again organized by layer, to determine if the solution is ready for release to production:

- **OS layer**
 - Is the new Windows operating system stable in all environments (development, test, staging, conversion, production) of the enterprise for the LOB system?
 - Are the latest patches installed?
 - Has the system been running in a stable mode without any crashes or reboots during testing?
- **DB layer**
 - Is the SQL Server database working accurately?
 - Are all the stored procedures and triggers installed?
 - Have the business and technical performance criteria been met?
- **Application Layer**
 - Have all the customizations been imported and applied to the new system?
 - Is the application communicating to the new database effectively?
 - Are the response time and other performance criteria met?
 - Does the latest build contain all the customizations?
- **Middleware Layer**
 - Are all the online and batch interfaces communicating to the middleware software used in the enterprise?
 - Are the performance and response time meeting requirements?

Closing the Stabilizing Phase

Closing the Stabilizing Phase requires completing a milestone approval process. The team needs to document the results of the different tasks it has performed in this phase in order to submit the project to management for approval.

Key Deliverables from the Stabilizing Phase

A deliverables checklist for the Stabilizing Phase includes:

- Production-ready release
- Release notes
- End-user help and training materials
- Testing and bug reports
- Source documentation and executables
- Updated risk management document

Major Milestone: Release Readiness Approved

Stabilizing the replatform solution culminates in the Release Readiness Approved Milestone meeting. This occurs at the point when the team has addressed all outstanding issues. To prepare for this meeting, the Test team must have completed a Release to Production Report containing the following:

- Signoff by IT and the business that all high priority bugs have been resolved
- Signoff by the business on the user acceptance tests
- Signoff by the business on the pilot.

Key project stakeholders, users, operations and support personnel, and the project team all review the Release to Production Report to evaluate whether the new replatform system is ready to be released to production and identify any remaining issues they need to address before release. A formal sign-off indicating approval is normally required. This meeting is often known as the "go/no-go" decision meeting.

This milestone also marks the point when responsibility for ongoing management and support of the solution officially transfers from the project team to the operations and support teams like the internal help desk.

5

Deploying Your Replatform Solution

Introduction and Goals

The primary goal of the Deploying Phase is to place the replatform solution into the organization's production environment. The project team must follow the deployment plan and the mockup of production they specified and approved in the replatform project plan. (Refer to the Deployment Plan section in Chapter 2 for details.) This plan should have been updated as the replatform project progressed. Prior to deployment, the plan should undergo a final review with the replatform team.

Replatform deployment at the highest level involves:

- Shutting down the old system or running the old system in parallel.
- Copying data to the new system at the cutoff time on the specified date of the deployment weekend, according to the agreed upon code freeze and cutover strategy.
- Bringing up the new system live for production or running two systems in parallel.

Most organizations do not run two systems in parallel unless the old system needs to be available for a very specific reason, such as read-only mode to view data or for management to run reports. Sometimes if the LOB system is very large the old system is kept running until the new system is fully stable in production. These are extreme cases and it happens very rarely.

Once the go decision has been communicated and all the required paperwork to go to production is complete, you are ready for the deployment weekend.

The tasks and deliverables that relate to the primary focus for the Deploying Phase are listed in Table 5.1.

Table 5.1 Developing Phase Major Tasks and Deliverables

Major Tasks	Deliverables
Completing deployment preparations	Production mockup schedule and updated deployment plan
Creating operations procedures	Operations procedures to run LOB application on Windows
Deploying the solution	Solution in production
Stabilizing the deployment	

Major Tasks	Deliverables
Transferring ownership to operations	
Closing the Deploying Phase at the Deployment Complete Milestone meeting	Sign-off by customer, stakeholders, and project team that the milestone has been met

Deployment Contingencies

Before your project enters the deployment weekend make sure you have reviewed and updated your contingency plan. If the new system does not come up properly, you want to make sure that you can fall back on your old system and redeploy it before the next business day.

Fortunately, the new system seldom fails to come up; when it does, it is because of a major technical crash during the deployment weekend. Rare or not, you must be prepared to handle such a contingency. Despite the number of tests you have run, there is always a chance that some overlooked detail can cause something to go wrong at the last moment — a hardware failure, an unplanned event, or an unknown possibility. In any event, be prepared.

Deployment Weekend

The deployment weekend is described at a high level in the following sections by individual LOB. It is recommended that you work with your certified partner or certified consultant to go through all procedures and technical commands in detail. As in the Developing Phase chapter, the steps focus on the database replatform deployment.

SAP Replatform Deployment Weekend

Plan and execute the deployment weekend methodically and carefully. The steps at a high level are:

1. Do a backup of your system.
2. Shut down the old source system.
3. Copy the data from source to the target system using the R3/Load Unload Tool.
4. Apply the new license key.
5. Bring up the new system (with the application layer).
6. Check that all interfaces for the middleware are working properly.
7. Go live.

PeopleSoft Replatform Deployment Weekend

Plan and execute the deployment weekend methodically and carefully. The steps at a high level are:

1. Do a backup of your system.
2. Shut down the old source system.
3. Copy the data from source to the target system using the Data Mover, DTS, or a combination.

4. Bring up the new system (with the application layer).
5. Check that all interfaces for the middleware are working properly.
6. Go live.

Siebel Replatform Deployment Weekend

Plan and execute the deployment weekend methodically and carefully. The steps at a high level are:

1. Do a backup of your system.
2. Shut down the old source system.
3. Copy the data from source to the target system using the EIM Tool.
4. Bring up the new system (with the application layer).
5. Check that all interfaces for the middleware are working properly.
6. Go live.

Closing the Deploying Phase

After deploying the replatform solution, the team might continue stabilizing the deployment during this period as the project components are transferred from the test or staging environments to the production environment.

The Warranty Period

After you have successfully deployed the replatform project a final period of stabilization occurs while the replatform is in production, sometimes referred to as the production stabilization stage. Typically organizations ask the certified partner who has contracted for the replatform project for a 30 to 90-day warranty period after going live in production. Bugs that were not identified or found during planning, developing, stabilizing, and deploying might now come up in the actual production environment. The warranty gives you the opportunity to have your certified partner fix anything that was overlooked without incurring extra charges. Use your organization's normal change control process to fix the bugs and move the fixes to production as soon as possible. After a couple of weeks you should have a stable system and you can begin to close the project.

Project Review

At the completion of the replatform project, the project team analyzes the results and produces documentation that can be used in future replatform efforts. This can be done through a group "lessons learned" session, for example. During and after the session, document the findings and discoveries. The documentation should include a paper that describes:

- What was done and what happened.
- What worked.
- What did not work, or did not work as expected.
- What steps were taken to fix unexpected results.
- Major replatform issues (a summary).
- How to make the next replatform more effective and efficient.

In addition to this documentation it is a good idea to create a presentation for IT management and the end-users of the replatformed applications, which shows the evolution of the project and what you learned for the future.

The team should also conduct a customer satisfaction survey.

As a final action to conclude the project, the replatform team can decide how to proceed in the future. The next step is often the beginning of a new phase of replatforming that was not within the scope of the current effort, such as replatforming additional applications. Address and resolve any significant replatform issues before starting another replatform effort.

By following the roadmap in this guide to do a replatform project, the organization gains valuable experience in the tools required for the job and a proven process to replatform from UNIX to Windows. Create a project binder, burn a CD of the documentation, the replatform project and the presentations, and archive all you have done.

As the project draws to a close, provide for a knowledge transfer from the Certified Consultants to the in-house IT staff. The in-house IT staff should document the processes and steps in as much detail as possible to provide a reference for how the current LOB platform configuration was implemented.

Key Deployment Deliverables

The deliverables of the Deploying Phase include:

- Final versions of all solution documents, code, and test cases
- Training documentation
- Service Level Agreements
- Operation and support procedures
- Updated risk management document

Major Milestone: Deployment Complete

An important final task is obtaining final customer (end-user represented) approval of the project. After you complete the replatform and transfer ownership and responsibility to operations and support, obtaining customer sign-off signifies the achievement of the fifth and final major project milestone, Deployment Complete.

6

Operating and Maintaining the New Platform

Introduction

The purpose of this chapter is to direct you to some available resources for maintaining and managing your LOB system on the new platform.

For tuning system performance, see the links to SAP, PeopleSoft, and Siebel resources. For managing and optimizing Windows and SQL Server operations, see the links to Microsoft resources.

Monitoring Performance

The following sections present information and links about some of the performance tuning options available to you.

SAP Performance Tuning

For SAP performance tuning use your customer user ID and password to read performance tuning information at:

- <http://service.sap.com/earlywatch>
- <http://service.sap.com/goinglivecheck>
- <http://service.sap.com/books>

PeopleSoft Performance Tuning

For PeopleSoft performance tuning log onto the PeopleSoft Web site:

- <https://www.peoplesoft.com/corp/en/login.jsp>
Use your PeopleSoft customer user ID and password to look at the following article: "PeopleSoft Red Paper Series – Microsoft SQL Server Tuning Tips for PeopleSoft 8.x" by Ganapathi Sadasivam. If you have questions related to SQL Server tuning, send an e-mail to redpapers@peoplesoft.com
- <http://www.peoplesoft.com/corp/en/products/epm/index.jsp>

Siebel Performance Tuning

For information about common Siebel operations issues, visit the following Web site:

http://www.siebelonmicrosoft.com/mssiebel/Downloads/PDF/Siebel_MOM_FAQ.pdf

For Siebel performance tuning, visit the following Web sites:

- <http://www.siebelonmicrosoft.com/mssiebel/Downloads/PDF/scid=kb-en-us-323950.pdf>
- <http://www.siebelonmicrosoft.com/mssiebel/Downloads/PDF/scid=kb-EN-US-324217.pdf>

For monitoring information, visit the following Web site:

- <http://www.siebelonmicrosoft.com/mssiebel/Technology/MOM.aspx>

For more information go to Siebel Support Web and use your Siebel customer user ID and password at:

- <https://ebusiness.siebel.com/supportweb/>
Siebel Support Web has the knowledge base and a wealth of articles to help you identify and resolve performance problems. If you cannot find the solution of a performance problem on Support Web, you can log a ticket with the Siebel Support Customer Service and a Siebel Helpdesk representative will work with you to resolve the issue.

Managing Operations on Windows

After your replatform you will need operations tools to manage your IT operations on the new platform. Two of the tools that are likely to be useful to you are Microsoft Operations Manager (MOM) and Systems Management Server (SMS).

MOM was created to help prevent potential IT problems and support issues that arise. This tool is designed to help staff stay aware of the status of IT health, improve response times to any issues, and effectively share information about key issues.

For more information about Microsoft Operations Manager, visit <http://www.microsoft.com/mom/>

<http://www.microsoft.com/technet/itsolutions/cits/mo/default.mspx>

You can download the *Microsoft Operations Manager 2005 Operations Guide* at the following link:

<http://www.microsoft.com/downloads/details.aspx?FamilyID=A0E40758-CAB8-4588-B0F2-1508D84906CC&displaylang=en>

SMS is a tool for change and configuration management for the Microsoft platform, enabling organizations to easily deploy patches and upgrades. For an overview, see:

<http://www.microsoft.com/smsserver/techinfo/default.asp>

Microsoft Updates and Service Packs

Microsoft updates and service packs enable simultaneous installation of the latest updates to both servers and workstations. Microsoft releases updates and service packs after many hours of testing and evaluation. These updates and service packs are available from the Microsoft Windows Update Web site at

<http://windowsupdate.microsoft.com>.

<http://www.microsoft.com/technet/itsolutions/cits/mo/swdist/pmsms/2003/pmsms031.mspx>

Visit the following site for information on deploying and maintaining SMS:

<http://www.microsoft.com/technet/prodtechnol/sms/sms2003/default.aspx>

The following link provides information on Managing Windows Server 2003 Best Practices:

<http://www.microsoft.com/technet/community/events/windows2003srv/tnt1-106.aspx>

Microsoft SQL Server Operations Guide

Visit the following link for the SQL Server Operations Guide:

<http://www.microsoft.com/technet/prodtechnol/sql/2000/maintain/sqlops0.aspx>

Summary

This chapter concludes the Microsoft Solution Guide for Replatforming Line-of-Business Applications from UNIX to Windows: SAP • PeopleSoft • Siebel Throughout this guide you have been encouraged to work with your certified partner for SAP, PeopleSoft, and Siebel to have a successful replatform experience. If there are any questions or detailed information you need regarding any topic in this guide contact Microsoft at plat-LOB@microsoft.com

Glossary

code freeze

The state and time at which all further programming, enhancements, and bug fixes are stopped for stabilization purposes.

conceptual design

A major stage in the design process for a solution in which the project team translates the business requirements into a common language to be shared by users and developers and describes the feature set and usage scenarios that the solution must encompass. Conceptual design is analogous to the rough sketches created when designing a house. These are easily understood models jointly created by the customer and the architect.

customer

An individual or organization that expects to gain a business value from the solution. Also known as the business sponsor. Also, the recipient of a service or product.

cutover

Switching from the old (hardware and/or software) system to the new system; covering the overlap from when the new system is live until the old system has been shut down.

line-of-business (LOB) application

A line-of-business (LOB) application is a critical computer application that is vital to running an organization, such as an accounting, supply chain management, or resource planning application. LOB applications are usually large programs that contain a number of integrated capabilities and tie into databases and database management systems.

living document

A document that is regularly updated and referred to.

logical design

A major activity in the design process in which the team deconstructs scenarios into basic elements and makes high-level decisions about the interaction and integration of IT components prior to making specific technology decisions.

physical design

The third major stage in the design process for a solution in which the project team determines how to specifically implement the logical design. Physical design addresses the technology that will be used by the end user. It applies real-world technology constraints to the logical design, such as implementation and performance considerations. It corresponds to a contractor's blueprints for the physical elements of a structure—wiring, plumbing, heating, and ventilation.

Portable Operating System Interface (POSIX)

A set of IEEE standards that defines how applications work on different operating systems.

replatforming

Replatforming is the process of changing the infrastructure on which LOB applications run to another type of technology and vendor. The infrastructure consists of the server, network, and the operating system.

Redundant Array of Independent Disks (RAID)

A category of disk drives that employ two or more drives in combination for fault tolerance and performance. For more information about RAID, visit <http://www.webopedia.com/>

risk management

A proactive, formalized process for decision-making and actions to continuously assess what can go wrong in a project, determine what risks are important to address, and implement strategies to deal with those risks.

risk mitigation

In risk management activities, an action that may be taken ahead of time to either prevent a risk from occurring altogether or reduce the impact or consequences of its occurring to an acceptable level.

scope creep

Unmanaged scope change. The risk that additional user requirements will cause the project to expand beyond the original scope. Scope creep should be avoided when possible.

service level agreement (SLA)

An agreement between IT and the user community that defines the responsibilities of all participating parties and that binds IT management to provide a particular service of a specific agreed-upon quality and quantity. It constrains the demands users may place upon the service to those limits defined by the agreement.

solution concept

A high-level description of how the solution will meet goals and requirements. Includes an early description of functionality, an initial approach to building and delivering the solution, and project success factors and acceptance criteria.

stakeholder

A person with a significant interest in the outcome of a project. A business unit vice president is an example of a stakeholder.

sub-team

A smaller team created from a larger team for the purposes of faster implementation and lower overhead in the areas of process, management, and communication. Sub-teams also allow a focus upon a particular capability. A core MSF team will coordinate and synchronize sub-teams.

user

The person who uses the solution or services on a day-to-day basis. Individuals or systems that directly interact with the solution.

vision

An unbounded view of what the team wants to accomplish in a solution.

vision statement

A statement that expresses the long-term vision of the solution and provides a context for decision-making. A deliverable of the Envisioning Phase in the MSF Process Model.

Note: See the *UNIX Migration Project Guide* for a more extensive glossary of IT project-related terms.

Acronyms

Acronym	Definition
CCB	Change control board
CPU	Central processing unit
DTS	Data Transformation Services
FDA	U.S. Food and Drug Administration
HCFA	Health Care Financing Administration
HIPPA	Health Insurance Portability and Accountability Act
JAD	Joint Application Development
MCS	Microsoft Consulting Services
OLTP	Online transaction processing
OOB	Out of the box
POC	Proof of concept
SAN	Storage area network
SFN	Services for NetWare
SI	Systems Integrator
SLA	Service level agreement
SMTP	Simple Mail Transfer Protocol
SSL	Secure Socket Layer
SUS	Software Update Service
TAM	Technical account manager
TCO	Total cost of ownership
TCP/IP	Transmission Control Protocol/Internet Protocol
UAN	Universal Application Network
UMPG	Unix Migration Project Guide
UNC	Universal Naming Convention
UPS	Uninterruptible power supply
URL	Uniform Resource Locator
VLDB	Very large database
VPN	Virtual Private Network
WAN	Wide area network
WINS	Windows Internet Naming Service

References

For more information about UNIX to Windows replatform and migration solutions refer to the following Microsoft Web sites and references:

- Microsoft Windows Server System™ Web site
<http://www.microsoft.com/windowsserver2003/migrate/unix/resources/default.aspx>
- Microsoft TechNet Web site: Interoperability and Migration
<http://www.microsoft.com/technet/interopmigration/default.aspx>
- The UNIX Migration Project Guide
<http://www.microsoft.com/technet/itsolutions/migration/unix/umpg/default.aspx>
- The UNIX Application Migration Guide
<http://msdn.microsoft.com/library/default.asp?url=/library/en-us/dnucmg/html/ucmgpl.asp>
- Solution Guide for Migrating Oracle (UNIX) to SQL Server (Windows)
<http://www.microsoft.com/technet/interopmigration/default.aspx>

Note: This guide will be available on TechNet in February 2005.

Microsoft offers various assistance programs to help plan your replatform in a way that adds the most value to your business. Microsoft Consulting Services offers the following services to help start your replatform project:

Strategy briefings — A single day of presentation and discussion to learn about Microsoft tools for UNIX-to-Windows replatforming, and Windows Server System features. The strategy briefing helps provide a clearer picture of your organization and technical needs and how the proper replatform strategy can add value to your organization.

Quick Start — An architecture design workshop where technical decision makers can examine the issues and details of a UNIX-to-Windows replatform. Following the examination is a proof-of-concept engagement where a prototype implementation is put together to demonstrate training, planning, and testing.

Quick Plan — An in-depth program that helps you develop a plan based on your requirements and existing environment. This plan considers account servers and desktops, available resources, and security needs. Quick Plan also includes a risk assessment that helps identify key areas that will demand attention.

Contact Microsoft Consulting Services at mcsdal@microsoft.com for detailed information on Strategy briefings, Quick Start or Quick Plan.

Microsoft also offers a TCO study in which a representative from Microsoft Consulting Services spends two weeks with your organization to help you with your TCO analysis. Contact the representative for your specific LOB application replatform at one of the following e-mail addresses:

- plat-LOB@microsoft.com
- plat-SAP@microsoft.com
- plat-Psf@microsoft.com
- plat-Sie@microsoft.com

White Papers, Case Studies, Benchmarks and Links

Following is a list of white papers, case studies, benchmarks, and links that show why some enterprises decided to run their LOB applications on a Windows platform:

SAP

- SAP Case Studies
<http://www.microsoft-sap.com>
- Search Results on Microsoft.com for SAP
<http://www.microsoft.com/resources/casestudies/FindCaseStudyResults.aspx?SearchTypeId=1&Keyword=SAP>
- Suncor Energy — Suncor Migrates SAP from Sun/Oracle to Windows, Lowers TCO by 42 Percent
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=14874>
- Analog Devices — Chip Manufacturer Raises Engineering Efficiency and Saves U.S. \$310,000 Annually – Integrates ePAS Application with SAP
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15458>

PeopleSoft

- PeopleSoft Case Studies
<http://www.peoplesoft.com>
- Search Results on Microsoft.com for PeopleSoft
<http://www.microsoft.com/resources/casestudies/FindCaseStudyResults.aspx?SearchTypeId=1&Keyword=PeopleSoft>
- El Paso Corporation — Energy Company Reduces Expenses with Switch to New Database System
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15785>
- United States Department of Agriculture (USDA) — USDA Increases Performance and Scalability of HR System, Supports More Agencies
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15260>

SIEBEL

- Siebel Case Studies
<http://www.siebelonmicrosoft.com>
- 30000 User — Comparative TCO Analysis of Siebel 7.x CRM Application on Alternative Platforms
http://www.siebelonmicrosoft.com/mssiebel/Downloads/Application_Replatforming/Benchmarks/Unisys_30k_Perf_Brief.pdf
- 12000 User — Comparative TCO Analysis of Siebel 7.x CRM Application on Alternative Platforms
http://www.siebelonmicrosoft.com/mssiebel/Downloads/Application_Replatforming/Benchmarks/HP_Siebel_Benchmarks.pdf
- 5000 User — Comparative TCO Analysis of Siebel 7.x CRM Application on Alternative Platforms

http://www.siebelonmicrosoft.com/mssiebel/Downloads/Application_Replatforming/Benchmarks/IBM_Siebel_5000_Benchmark.pdf

http://www.siebelonmicrosoft.com/mssiebel/Downloads/Application_Replatforming/Benchmarks/Unisys_Siebel_5k_Benchmark.pdf

SQL Server Benchmark for SAP, PeopleSoft and Siebel

<http://www.microsoft.com/sql/evaluation/compare/benchmarks.asp>

Examples of Other Replatform Case Studies

Microsoft has an extensive library of UNIX replatform and Windows Server System case studies. A search tool that is useful for locating case studies or examples of specific UNIX replatform projects is available at

<http://www.microsoft.com/resources/casestudies/FindCaseStudy.aspx>

Following is a brief list of recent case studies:

- Grand Expeditions — Travel Company Slashes Web Costs, Speeds Introduction of New Features
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15397>
- WineBid.com — Online Wine Auction Site Grows Revenue by 10% with Comprehensive Portal Solution
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15135>
- Indiana Department of Administration — State of Indiana Realizes 90-day ROI Migrating to Microsoft Windows and SQL Server Database
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=15028>
- Group 4 Falck A/S — Group 4 Ports Real-Time Critical Monitoring Application by Using Microsoft Windows Services for UNIX
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=14884>
- Mahindra British Telecom Limited (MBT) — MBT Uses Microsoft Windows Services for UNIX to Rapidly Migrate Customer's Solaris Application to Windows
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=14925>
- Boston University — Boston University Lightens Administrative Burden with Windows Active Directory and Dell PowerEdge Servers
<http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=13803>

Appendices

Appendix A: Windows Platform Capabilities

After you have determined which business goals are appropriate for your business requirements and replatform project, you might want to investigate in detail how the Windows platform can support your business goals and requirements.

For more information about the Microsoft Windows Server™ 2003, the Windows Server System, and replatforming on Windows visit the following Web sites:

- <http://www.microsoft.com/windowsserver2003/>
- <http://www.microsoft.com/windowsserversystem/>
- <http://www.microsoft.com/windowsserver2003/migrate/unix/default.mspix>

Following is a list of information and links that are specific to each goal:

- **Available Expertise.** Microsoft provides a large resource pool of Windows experts and training and certification options, typically at lower costs compared to other options.
- **Operations Productivity.** The Windows platform helps manage network users' productivity by actively enforcing policy, automating tasks, and simplifying updates.
- **High Server Performance.** Windows Server 2003 is a high performance server system. As of the publication date of this guide, the Transaction Processing Performance Council recognizes systems configurations from Dell, Hewlett-Packard, IBM, and RackSaver among its ten best-rated systems in terms of cost per transaction per second. The average user population represented in these benchmarks is approximately 24,400 from a range of 17,200 to 35,000 users. Nine of the ten systems use Windows Server 2003 as their OS.

For the most current information about TPC-C benchmarks on cost per transaction per second, visit http://www.tpc.org/tpcc/results/tpcc_price_perf_results.asp.

For more information on the TPC-C benchmark database, visit http://www.tpc.org/tpcc/results/tpcc_results.asp?orderby=hardware.

For more information on the TPC-H benchmark databank, visit http://www.tpc.org/tpch/results/tpch_results.asp?orderby=hardware.

For more information on this example of Microsoft SQL Server performance, visit <http://www.microsoft.com/resources/casestudies/CaseStudy.asp?CaseStudyID=12112>.

For more information on Microsoft SQL Server, visit <http://www.microsoft.com/sql/>.

- **Scalability and Flexibility.** Microsoft Windows Server 2003 provides the flexibility that LOB applications need to perform as demands increase. For information on Windows Server 2003 scalability, visit the following Web sites:
 - <http://www.microsoft.com/windowsserver2003/techinfo/serverroles/appserver/scale.mspix>
 - <http://www.siebel.com/crm>
 - <http://www.microsoft.com/sql/evaluation/compare/benchmarks.asp>
- **Integration.** Microsoft supports strong integration of its platform with SAP, PeopleSoft, and Siebel systems. For example, the SAP system can use the "service publication" feature of Microsoft Active Directory® to detect SAP R/3

systems and their services, such as the application servers, message servers, database, gateway service, and SAP Internet Transaction Server (ITS) instances. The SAP R/3 version 46C Microsoft Management Console (MMC) snap-in is the first component to use information provided by Windows Active Directory. SAPGUI for Windows uses Active Directory to obtain a list of SAP systems. SAP Central User Administration (Web Application Server version 6.10) supports synchronization with Active Directory allowing the easy management of the identities in your organization.

True integration is not limited to the operating system and the database but also includes non-Microsoft tools used in an organization.

- **Interoperability.** Microsoft provides powerful management tools for integrating legacy systems with the Microsoft platform. The Windows Server System interoperates seamlessly with other systems through its adherence to industry standard technologies such as Extensible Markup Language (XML) and the Simple Object Access Protocol (SOAP). Microsoft has made a commitment to supporting UNIX applications on the Windows Server System, and provides a suite of applications, partners, training, and guidance to ensure the success of your system replatform.

The Microsoft Services for UNIX (SFU) product provides powerful management tools that enable the integration of your existing UNIX systems with Windows Server. SFU enables organizations to continue to run their custom Portable Operating System Interface (POSIX) – compliant applications and scripts, often with no required changes. SFU provides for seamless integration with existing systems, enabling a more incremental migration if desired. SFU also offers manageability components for simplifying network administration and account management across both platforms. SFU provides tools that enable a UNIX-based workforce to incrementally replatform to Windows Server, using their existing UNIX skills while becoming immediately productive on Windows Server.

- **Reliability.** Microsoft approaches reliability for technology and support through the following:
 - Vision and framework
 - Technology and integration
 - Best practices recommendations
 - Global updates

For information on Microsoft's approach to reliability, visit <http://www.microsoft.com/mscorp/twc/reliability/default.msp>

- **Security.** Microsoft addresses security needs with technologies for identity management, data integrity, and protection from intentional attack. For more information about Windows Server 2003 security, visit <http://www.microsoft.com/windowsserver2003/technologies/security>.

Windows Server 2003 can provide authentication and identity store services within heterogeneous Windows Server System and UNIX environments. Windows uses the standards-based security and directory services provided by LDAP and Kerberos to provide secure and single sign-on capabilities among Windows, UNIX and Linux systems.

For more information on security and directory services, visit the following Web sites:

- <http://www.microsoft.com/security/default.msp>
 - <http://www.microsoft.com/sql/techinfo/administration/2000/security/default.asp>
- **Manageability.** Windows provides the following management tools:

- **Systems Management Server (SMS).** SMS is an enterprise-level tool for managing all the platforms, servers, and workstations of a Windows environment. SMS can create software and hardware inventories of a Windows infrastructure, and download and distribute patches and software updates and upgrades. Your IT managers stay on top of changes that may or may not be compatible with your IT policies. IT professionals can save time with automated software distribution mechanisms, dramatically reducing the time spent creating and testing distribution scripts.
- **Microsoft Operations Manager (MOM).** MOM enables organizations to elevate service levels with management packs that make available corporate operating knowledge and best practices. MOM management packs enable IT staff to resolve IT OS and application problems by tapping into the advice and knowledge of the creators of the related applications and technologies.
- **Resource Kit Utilities.** Resource kits contain tools to assemble an active defense that often enable a quick resolution to problems that might occur. Many of the best IT administration tools are built right into the Windows Server operating systems, and provide powerful problem identification and resolution tools at no extra cost.
- **Vendor Support.** SAP, PeopleSoft, and Siebel certified partners and Microsoft consultants offer extensive experience and performance in doing replatform projects. Certified partners work closely with Microsoft, SAP, PeopleSoft, and Siebel Alliance teams and are extensively trained in Microsoft technologies to provide the best service.

For more information about certified partners, visit the following Web sites:

- <http://www.microsoft-sap.com/resources.asp>
- http://www.peoplesoft.com/corp/en/partners/find/directories/all_partners.jsp
- <http://www.siebelonmicrosoft.com/mssiebel/Partners.aspx>

Appendix B: Templates

The following templates are available on the Computer Consultants International, Inc. (CCI) Web site at:

- <http://www.cci-worldwide.com/replat/sap>
- <http://www.cci-worldwide.com/replat/peoplesoft>
- <http://www.cci-worldwide.com/replat/siebel>

Inventory Template

The InventoryTemplate.xls file provides a kick start to Inventory. The file is a Microsoft template that you can modify to suit your own project requirements.

Sample Project Plan

The sample.mpp is a sample project plan for replatform projects.

Master Test Plan Template

The Master Test Plan Template.doc file provides an MTP template that you can customize for your replatform project.

Test Cases Template

The Detailed Test Cases Template.doc file provides a template that you can customize to document the test cases used in testing your replatform project.

For more information about these templates, see:

Computer Consultants International, Inc.

Web site: <http://www.cci-worldwide.com/replat>

e-mail: replat@cci-worldwide.com