

Martin Riedel

Managing SAP[®] ERP 6.0 Upgrade Projects





Contents at a Glance

1	Introduction	13
2	Getting Started	17
3	Planning an Upgrade Project	57
4	Managing an Upgrade Project	117
5	Executing an Upgrade Project	159
6	Upgrade Tools and Services	199
7	Customer Cases and Upgrade Communities	253
Арр	endix	
Α	SAP ERP Upgrade Checklist – Project Preparation	323
В	SAP ERP Upgrade Checklist – Blueprint	327
c	SAP ERP Upgrade Checklist – Realization	333
D	SAP ERP Upgrade Checklist – Final Preparation for Cutover	341
E	SAP ERP Upgrade Checklist – Production Cutover & Support	343
F	References	345
G	Authors and Contributors	351

Contents

1	Intro	oductio	n	13
2	C	·		47
2	Gett	ing Sta	urted	17
	2.1	Introd	uction to SAP ERP 6.0	17
		2.1.1	Evolution of SAP ERP 6.0	18
		2.1.2	Functional Overview of SAP ERP 6.0	20
		2.1.3	What Runs Best on SAP ERP 6.0?	22
		2.1.4	Integrated and Switchable Industry	
			Functionality	24
		2.1.5	Solution Browser Tool for SAP ERP	25
	2.2	SAP Er	nhancement Packages—Concept and Overview	25
	2.3	Discov	very Process	29
		2.3.1	Collecting Preliminary Information	29
		2.3.2	Stakeholder Analysis and Involvement	31
		2.3.3	Determining the Business Value of new	
			Functions	31
		2.3.4	Organizing Delta Training and Workshops	32
		2.3.5	Trying out new Functions	32
		2.3.6	Finalizing the Business Case and Justifying	
			the Upgrade	33
	2.4	Why U	Ipgrade Your ERP Software?	34
		2.4.1	Upgrade Approaches	35
		2.4.2	Essentials of an Upgrade Justification	38
		2.4.3	Operational Excellence	39
		2.4.4	Business Strategy	40
		2.4.5	Sustainability	41
		2.4.6	тсо	43
		2.4.7	Cost Reduction Opportunities for a	
			Technical Upgrade	46
		2.4.8	The Downside of not Upgrading	48
	2.5	Detern	nining the Value of an Upgrade	48

3	Planr	ning an	Upgrade Project	57
	3.1	Determ	ining the Upgrade Strategy and Scope	58
		3.1.1	Factors Influencing the Complexity of an	
			Upgrade	58
		3.1.2	Technical Considerations	67
		3.1.3	Enhancement Packages Installation Scope	70
		3.1.4	Unicode Conversion Options	74
		3.1.5	Other Factors Influencing the Upgrade	81
	3.2	Resourc	cing Models and Approaches	86
		3.2.1	Internally Managed	87
		3.2.2	Externally Managed	88
		3.2.3	Summary	91
	3.3	Schedu	ling an Upgrade	92
	3.4	Estimat	ing Cost and Effort	99
		3.4.1	Trial Upgrade	100
		3.4.2	Preliminary Upgrade Assessment	101
	3.5	Project	Standards and Procedures	108
	3.6	Risk Ma	anagement	110
		3.6.1	Identifying Risks	110
		3.6.2	Risk Management During the Project	112
		3.6.3	Risk Classification	113
		3.6.4	Identifying Potential Upgrade Project Risks	114
		3.6.5	Risk Identification Checklist	115
		3.6.6	SAP Safeguarding for Upgrade	116
			5 5 15	
4	Mana	aging a	n Upgrade Project	117
	4.1	Buildin	g a Project Team	118
		4.1.1	Project Team Scope	119
		4.1.2	Project Team Roles	119
		4.1.3	Project Team Training	121
		4.1.4	Project Team Availability	122
	4.2	Quality	Assurance and Testing	122
		4.2.1	Testing Focus	123
		4.2.2	Test Stages and Test Progression	124
		4.2.3	Testing Resources	126
		4.2.4	Test Systems	128

4.2.5 Test Data Management 128

	4.2.6	Choosing Between Manual or Automated	
		Testing	129
	4.2.7	Benefits of Automated Testing	131
	4.2.8	Benefits of Manual Testing	132
	4.2.9	SAP Solution Manager	132
	4.2.10	Test Activities in the SAP Upgrade Road Map	134
	4.2.11	Test Case Templates for Enhancement	
		Packages for SAP ERP	136
	4.2.12	Successful Testing	137
	4.2.13	Quality Management and Quality Gates	139
4.3	Cutove	r Planning	140
	4.3.1	Input from Previous Project Phases	140
	4.3.2	Cutover Plan Contents	141
	4.3.3	Cutover Prerequisites	143
	4.3.4	Sample Cutover Plan for a Weekend Upgrade	144
	4.3.5	Unicode Conversion Phase	147
	4.3.6	Post Cutover Activities	148
4.4	Best Pra	actices	149
	4.4.1	Project Management	149
	4.4.2	Technical Best Practices	153
	4.4.3	Upgrade Project Errors	157

5 Executing an Upgrade Project 159

5.1	Manag	ing the System Landscape During an	
	Upgrad	de Project	160
	5.1.1	Recommended System Landscape	161
	5.1.2	Project Preparation	162
	5.1.3	Upgrade Blueprint	164
	5.1.4	Upgrade Realization	165
	5.1.5	Final Preparation for Cutover	167
	5.1.6	Production Cutover and Support	168
5.2	Downt	time Minimization	169
	5.2.1	Definitions: Downtime, Uptime, Runtime	170
	5.2.2	Why is Downtime Necessary?	172
	5.2.3	Downtime Facts and Figures	172
	5.2.4	Choosing an Upgrade Strategy	172
	5.2.5	Elements of Business Downtime During an	
		Upgrade	175

	5.2.6	Factors that Influence Upgrade Runtime and	
		Downtime Duration	176
	5.2.7	Incremental Conversion (ICNV)	177
	5.2.8	Customer-Based Upgrade (CBU)	179
	5.2.9	Unicode Conversion	181
	5.2.10	Near Zero Downtime	182
	5.2.11	Unicode Conversion Downtime	183
	5.2.12	Best Practices – Upgrade Tuning	184
	5.2.13	Testing	186
	5.2.14	Splitting Downtime	186
5.3	Trainin	g	187
	5.3.1	Consider the Training Focus	188
	5.3.2	Create a Training Plan	189
	5.3.3	SAP Education	191
	5.3.4	E-Learning	193
5.4	Lessons	Elearned	194
	5.4.1	Project Management Aspects	194
	5.4.2	Functional Aspects	196
	5.4.3	Technical Aspects	197
Upgr	ade To	ols and Services	199
6.1	SAP So	lution Manager in Upgrade Projects	201
	6.1.1	Two Scenarios	204
	6.1.2	SAP Upgrade Road Map – Overview	206
	6.1.3	SAP Upgrade Road Map – Details	209
6.2	Technic	al Upgrade Tools	214
	6.2.1	SAPup and Upgrade GUI for ABAP	215
	6.2.2	SAPjup and SDTGUI for Java	216
	6.2.3	Synchronization of ABAP and Java Upgrades	217
	6.2.4	Upgrade Preparation	218
	6.2.5	Application-Specific Upgrade (ASU) Toolbox	218
6.3	Suppor	ting Upgrade Tools	219
	6.3.1	Solution Browser Tool for SAP ERP	219

- 6.3.3 Upgrade Dependency Analyzer 221
- 6.3.4 Scenario and Process Component List 223
- 6.3.5 Business Process Change Analyzer 223

6

	6.3.6 6.3.7	Solution Documentation Assistant SAP Custom Development Management	224
	0.5.7	Cockpit	224
6.4	Testing	g Tools	224
	6.4.1	SAP Solution Manager for Testing	224
	6.4.2	SAP Test Data Migration Server (SAP TDMS)	227
	6.4.3	Business Process Change Analyzer (BPCA)	229
	6.4.4	Solution Documentation Assistant	230
	6.4.5	Upgrade Project Testing with	
		SAP Test Workbench	231
	6.4.6	SAP Quality Center by HP	233
	6.4.7	SAP LoadRunner by HP	233
	6.4.8	SAP Test Acceleration and Optimization	234
6.5	Tools f	or Downloading and Installing SAP	
	Enhan	cement Packages	235
	6.5.1	Maintenance Optimizer	235
	6.5.2	Enhancement Package Installation Tools	
		and Process	239
6.6	Upgrad	de Services	241
	6.6.1	SAP Upgrade Value Assessment	242
	6.6.2	Quick Upgrade Analysis for SAP ERP	243
	6.6.3	Technical Upgrade Planning for SAP ERP	245
	6.6.4	Upgrade Coaching for SAP ERP 6.0	246
	6.6.5	Technical Upgrade Service for SAP ERP	248
	6.6.6	SAP Enterprise Support Services	248
	6.6.7	SAP Safeguarding for Upgrades	250
6.7	SAP Te	esting Services	250
Cust	omer C	Cases and Upgrade Communities	253
7.1	Custor	ner Cases	253
	7.1.1	Cincinnati Insurance Company	254
	7.1.2	Indesit Company SpA, Italy	258
	7.1.3	Nebraska Public Power District (NPPD)	266
	7.1.4	Pacific Coast Companies, Inc	268

7

	7.2	Upgrad	le Community	297	
		7.2.1	Deutschsprachige SAP-Anwendergruppe e. V.		
			(DSAG) – Cooperation at all Levels	298	
		7.2.2	SAP ERP Upgrades, ASUG and YOU	302	
		7.2.3	Club des Utilisateurs SAP Francophone (USF)	306	
Ар	pendi	×		321	
А	SAP E	RP Upgr	rade Checklist – Project Preparation	323	
В	SAP E	RP Upgr	rade Checklist – Blueprint	327	
С	C SAP ERP Upgrade Checklist – Realization 3				
D	D SAP ERP Upgrade Checklist – Final Preparation for Cutover 3				
Е	SAP ERP Upgrade Checklist – Production Cutover & Support 34				
F				345	
G	Autho	rs and C	Contributors	351	
Inde	ex			359	

How should the system landscape be set up during an upgrade project? How can downtime be minimized, and what support tools and methods exist? What about training the project team and end users? Chapter 5 answers these questions and provides recommendations based on lessons learned from numerous upgrade projects.

5 Executing an Upgrade Project

This chapter looks at the execution phase of an upgrade. Based on two main upgrade challenges perceived by customers, its focus is on downtime minimization and training. In addition, we will look at system landscape strategies that can help you successfully complete your upgrade project.

You can follow several different paths to tackle downtime. We will show you how careful analysis of your situation can help you manage this issue successfully. We will also describe the options offered by SAP to keep downtime to a minimum.

The section on a recommended system landscape outlines a recommendation for how to set up and manage your system landscape in an upgrade project. It is not feasible within the scope of this book to show all possible system landscapes, as each is customer-specific. However, it is instructive to examine and understand a basic setup that shows how the system landscape evolves throughout the project. From this, you can begin to analyze your own system landscape and take the necessary measures to build the appropriate environment specific to your requirements.

When planning to tackle an upgrade, it will most likely become apparraining ent a knowledge gap exists for many of your staff, either on the technical side or in terms of the functionality that will be implemented. In the section on training, you will find numerous recommendations for deter-

Downtime minimization

Recommended system landscape mining the training your organization will need as well as suggestions for relevant SAP courses.

Lessons learned Finally, we will look at some of the lessons learned from upgrade projects. These will give you insight into what makes a successful project, based on real experience from SAP's involvement in numerous upgrade projects.

5.1 Managing the System Landscape During an Upgrade Project

The goal of this section is to provide guidance and recommendations for a standard three-system SAP customer landscape; however, it cannot fully explore all of the potential additional systems and interdependencies you might have in your specific landscape.

A number of factors have a direct influence on the specific system landscape you choose for the upgrade project, most notably the following:

- The code freeze strategy.
- ► The technical availability of hardware for setting up temporary systems during the project (sandbox and training systems).
- ► The operational ease of setting up sandbox systems, shadow systems, and so on in your IT environment or with your hosting service provider.

Depending on the scope of the upgrade, your upgrade project can last several months. During this time, it is often inevitable that you will need to make at least some changes to the production system. For many customers, it is not possible to expect the business to suspend all changes to the system during that time. Therefore, you must define an appropriate change management strategy that will restrict and regulate changes to the production system within the context of the system landscape you are using for upgrade testing and project refinement.

Code freeze The code freeze period will usually start after the development system has been established. From that point in time, double maintenance of coding and other system configuration changes is necessary. At a point

close to the cutover weekend (usually after the final integration testing at the latest), you will also have to define a "hard" code freeze strategy which restricts transports even more and allows only the most urgent of changes to be implemented. Section 4.4, "Best Practices," in Chapter 4, provides a recommendation for a code freeze strategy.

You should also consider the need to set up separate systems for training Multiple systems purposes, interface testing, modification adjustments, and custom developments. Most customers, however, will use a three-tier system landscape with an additional sandbox as the "playground" for the upgrade project.

5.1.1 Recommended System Landscape

This section provides you with recommendations on how to set up your system to minimize upgrade risks and minimize the duration of the code-freeze period. The examples show a typical three system landscape, consisting of a development system, a quality assurance/consolidation system, and a production system. The recommendations show how additional copies of these systems are used to perform required activities during the upgrade project such as adjusting custom developments and testing. We will look at how the systems evolve during the different phases of the upgrade project:

- Project preparation
- Upgrade blueprint
- Upgrade realization
- ► Final preparation for cutover
- ▶ Production cutover (go-live) and support

The scenarios are based on a suggested timeline, which runs over four months. For each phase, each scenario gives a suggested duration, in weeks. The recommendations assume that you have the appropriate hardware available for creating additional systems. We describe actual project work, how each system is upgraded to the new release from the old release, and the transport routes required. **Recommended** Recommended project activities are suggested for each team involved at each phase, as follows:

Project management

The team in charge of the upgrade project that manages all activities that are part of the project.

Technical

The IT team that manages the system technology such as hardware, the operating system, and database software.

Development

The software development team responsible for custom developments and modifications to the core SAP software.

► Business experts

Experts from the organization's business units who understand the business processes and how SAP functionality is used within each process.

Although the system landscape shown in Figure 5.1 assumes an upgrade from SAP R/3 4.6C to SAP ERP 6.0, the source release is not a relevant factor in these examples, except for very old SAP R/3-releases. The main objective is to show how the system evolves with systems running either the source or the target release, culminating in an upgraded production system.

5.1.2 Project Preparation

Project preparation is the first phase of the project, where initial work is done on an upgraded copy of the production system.

Sandbox system The project management team must first arrange for an upgrade project system to be available. The next step is to prepare the upgrade project system (the sandbox system) as a copy of the production system (see Figure 5.1). Ideally, you should include as much realistic production data as possible in the upgrade project system.



Figure 5.1 System Landscape During the Project Preparation Phase

Project Activities

The following activities should be carried out in the project preparation phase:

- The project management team creates a detailed project plan, nominates the project team, and orders temporary hardware.
- The technical team prepares the sandbox system (SBX), also known as the upgrade project system.
- The development team reviews custom developments and modifications.
- ► The business experts study material on the new release (for example release notes and the contents of the Solution Browser tool for SAP ERP). They start preparing test scenarios and planning test execution.

Deliverables/Output

You have now established an upgrade project system (the sandbox system). The first version of the project plan is available, along with com-

prehensive understanding of the scope of the project in terms of technical, process, and functional changes that will be included.

5.1.3 Upgrade Blueprint

In the upgrade blueprint phase, the focus is on experimenting with the sandbox system through familiarization and testing of the new software (see Figure 5.2).



Figure 5.2 System Landscape During the Upgrade Blueprint Phase

Discovery Using this system, the technical, development, and business teams can begin the process of "discovering" the new software.

Project Activities

The following activities should be carried out in the upgrade blueprint phase:

The technical team executes a technical upgrade on the upgrade project system. A key activity is running and measuring the upgrade regarding downtime, and testing downtime minimization approaches.

- Developers perform SPDD/SPAU adjustment and test custom developments and modifications. SPDD/SPAU adjustment is not a necessary step and can be skipped to reduce project effort.
- Business experts carry out upgrade Customizing and testing of business processes (test cycle I: takes two weeks)

Deliverables/Output

At the end of this phase, business processes should be running properly in the sandbox system and there should be detailed documentation of the adjustment activities carried out by the development team.

These activities will help you refine the project plan and better understand the scope of the project. Refine the project plan

5.1.4 Upgrade Realization

The upgrade realization phase marks the beginning of the dual maintenance period (of the DEV' and DEV systems). Figure 5.3 shows the system landscape during the upgrade realization phase.



Figure 5.3 System Landscape During the Upgrade Realization Phase

- Dual maintenance In this phase, you create a copy of the development system (DEV') and perform a technical upgrade on the main development system (DEV). Alternatively, you can create the 6.0 DEV system as an upgrade from a copy of the production system (if size and security considerations allow it). This has the following advantages:
 - Consistency of DEV and PRD concerning development objects is easily enforced.
 - ► The upgrade of the production copy can yield a good assessment for the production upgrade (discarding factors such as machine size).

A disadvantage of copying the production system and upgrading it for the creation of the DEV system is the loss of versioning information, especially for ABAP developments.

Development activities concerning the upgrade project take place primarily in the main development system. However, the copy of DEV (DEV') is used to provide continuous support to the production system.

During dual maintenance, any changes you make to the contingency system because of production support requirements must also be made in the upgraded development system. It is important to consider a code freeze during this period. For suggestions on managing a code freeze during the dual maintenance period, see Section 4.4.2, "Technical Best Practices," in Chapter 4. Changes you make to the development copy (not yet upgraded) are transported to the quality assurance system (QAS) and from there to the production system (PRD).

Project priority Although the upgrade project should take priority, it is possible to continue working on concurrent projects. For example, you could work on a project that introduces custom development in the upgraded development system but for which the changes are not transported to the production system until after final the production cutover.

Project Activities

The following activities are carried out in the realization phase:

► The technical team sets up a temporary development system for maintenance (DEV') and upgrades the development system (DEV).

- ► Developers manually redo the SPDD and SPAU adjustment, manually redo the adjustment for custom developments, and perform short unit testing in the development system (DEV).
- ► The business experts redo Customizing adjustments and perform unit testing in the development system (DEV).

Deliverables/Output

By the end of this phase, you should have completed the unit testing for custom developments in the development system (DEV).

5.1.5 Final Preparation for Cutover

In the final preparation for cutover phase, you create a copy of the quality assurance system (QAS') and upgrade the original quality assurance system (QAS). Figure 5.4 shows the system landscape during this phase.



Figure 5.4 System Landscape During the Final Preparation for Cutover Phase

During this phase, you continue dual maintenance of the development QA system systems. You also transfer changes that you make in DEV' to DEV, and upgrade transport changes to both quality assurance systems.

Project Activities

The following activities are carried out during this phase:

- ► The technical team sets up the temporary quality assurance system (QAS'), upgrades the QAS system, transports project work to the QAS system, and continues to refine the cutover plan based on the work done so far.
- Developers correct errors in custom developments.
- Business experts perform final integration tests in the QA system (test cycle II: takes one week) and regression testing in the upgraded QA system.

Deliverables/Output

During this phase, the testing of business processes is completed, the downtime estimate is refined, and the cutover plan is finalized and approved. You should now be in a position to embark on the final phase: the cutover weekend where you upgrade the production system.

5.1.6 Production Cutover and Support

The production cutover and support phase is the final phase that marks the completion of the upgrade project and culminates with the go-live of the production system. Figure 5.5 shows the system landscape for this project phase.



Figure 5.5 System Landscape During the Production Cutover and Support Phase

Project Activities

The following activities are carried out during this project phase: Fi

Final phase

- ► The technical team upgrades the production system (PRD) (including downtime). When finished, it performs post-upgrade activities.
- ▶ Business experts sign off on the upgraded production system.
- ► Depending on the archiving strategy, DEV' and CON' may be archived.
- ► It might also be advisable to keep DEV' available for some time after the upgrade to check the former functionality in case of unexpected errors.

Deliverables/Output

At the end of this phase, the SAP ERP 6.0 system is released for productive operations and the temporary system landscape is removed. This is the final milestone: formal project closure. However, there is still a need for ongoing support of the upgraded system. Appropriate support activities must be adjusted and established.

5.2 Downtime Minimization

This section discusses system downtime in the context of an SAP upgrade project. Downtime is probably the biggest challenge of an upgrade and a crucial topic because it is the time when the system cannot be used by the business. In an SAP customer feedback survey, 54% of organizations identified downtime minimization as being an important consideration in both planning and executing an upgrade project.

Managing downtime is not just about controlling the time the system is Co unavailable, but also about controlling the costs incurred during downtime. Furthermore, costs are not linear: for longer downtime costs can increase exponentially.

The total amount of downtime permitted by the business has an effect on many of the other decisions you make when planning an upgrade project. Therefore, you must carefully determine the maximum downControlling costs

time that will be available and precisely detail the upgrade tasks that have to be performed during that time.

In this section, we will look at the factors that influence downtime and what you can do to tackle the challenge of downtime within your upgrade project. You will learn how to decide on the upgrade strategy to use and also find details about SAP tools you can use to help minimize downtime. We will also provide you with recommendations for how to reduce downtime costs.

5.2.1 Definitions: Downtime, Uptime, Runtime

This section provides an overview of what the terms *downtime*, *uptime*, and *runtime* mean in the context of an upgrade project. Figure 5.6 shows the business and technical perspectives of the downtime and uptime phases.



Figure 5.6 Business and Technical Downtime and Uptime

Downtime (both technical and business), uptime, and runtime are described in Table 5.1.

Term	Definition
Technical Downtime	This is the time period during which the upgrade tools perform the upgrade process without the sys- tem being available for end users. It does not include the time for data backup and final testing.
Business Downtime	This is the total amount of time during which the system is not available for end users. It includes the technical downtime and the time necessary for data backup and final tests.
Uptime	The time during which end users can use the sys- tem's applications in production while the upgrade process is running.
Runtime	The overall time it will take to carry out the upgrade process. It is measured from the start of the upgrade process to the end (when the system is available for productive use again), and consists of all upgrade uptime, technical downtime, and busi- ness downtime.

Table 5.1 Definitions of Downtime, Uptime, and Runtime

In any system, you can further divide downtime into planned and plunplanned downtime. *Unplanned downtime* concerns items over which you have little direct control such as hardware, or operating system failures, as well as human error.

Planned and unplanned downtime

Planned downtime, on the other hand, is very much under your control, in particular in terms of when it takes place. You are required to plan downtime for system and infrastructure maintenance and for the implementation of patches, upgrades, and changes to transports. Planned downtime can be minimized through the following:

- Scalable components that enable rolling maintenance
- Improved upgrade and patch processes
- Proven software lifecycle management and propagation engines (e.g., the transport management system or the change management service)

5.2.2 Why is Downtime Necessary?

One of the advantages of SAP's technology is that it allows customers to adapt, extend, and modify SAP software, and these extensions will be kept and adjusted to the new release during the upgrade process. Furthermore, most of the required processing steps that are part of the upgrade can be performed during system uptime.

Downtime is necessary whenever live, running transactions have to be replaced by new functionality and there is a potential risk for data inconsistency, for example due to a change in the processing logic, or a change to the data model or structure. You must prepare users for business downtime and make them aware of the need for both technical downtime and business downtime.

5.2.3 Downtime Facts and Figures

SAP has analyzed the average downtime (business downtime and technical downtime) for customers upgrading to SAP ERP 6.0.

Average business
downtimeThe average business downtime was calculated separately, according to
the SAP source release. Results showed 34 hours (for source release SAP
R/3 Enterprise) and 48 hours (for source release SAP R/3 4.6C). For all
upgrades to SAP ERP 6.0, the average technical downtime was 7.2 hours.

The chart shown in Figure 5.7 illustrates the minimum possible technical downtime for 110 customer upgrades to SAP ERP 6.0 SR3 where the downtime-minimized strategy was used.

5.2.4 Choosing an Upgrade Strategy

Preconfiguration mode

With the system switch technology, most upgrade activities are moved
 into uptime. When upgrading with the system switch upgrade procedure, SAP provides you with two upgrade strategies: the downtime-minimized strategy and the resource-minimized strategy. You must decide which strategy to use as determined by the requirements of your organization. Two factors are key to this decision:

- Maximum downtime permitted by your organization
- System resources available



Figure 5.7 Technical Downtime Hours Versus. Number of Upgrades (Snapshot of 110 Upgrades to SAP ERP 6.0 SR3)

The downtime and the consumption of system resources depend on the interaction of several parameters you can set for the upgrade. To optimize the duration of downtime and the consumption of system resources, parameter settings are grouped into preconfiguration modes. Instead of setting the parameters manually, you choose the preconfiguration mode that suits your system resource situation. You select the preconfiguration mode in the upgrade GUI during the upgrade procedure. By setting the preconfiguration mode, you can choose either a resource-minimized or a downtime-minimized upgrade strategy. For more details, see the SAP ERP 6.0 upgrade guides. Table 5.2 shows an overview of the three available preconfiguration modes.

SAP recommends using the downtime-minimized strategy for the majority of upgrades. The additional resources needed should be available because SAP ERP 6.0 technically requires them.

Preconfiguration Mode	Features
Low resource use	 Low system resource consumption Early start of downtime; shadow system operation during downtime (upgrade strategy parameter <i>resource-minimized</i>) ICNV tool cannot be used
Standard resource use	 Late start of downtime; import and shadow system operation while the system is still in production operation (upgrade strategy parameter <i>downtime-minimized</i>) Database archiving mode is off during downtime Database backup required before downtime ICNV tool can be used
High resource use	 Late start of downtime; import and shadow system operation while the system is still in production operation (upgrade strategy parameter <i>downtime-minimized</i>) Fast import Database archiving mode is on, which results in a large amount of archiving logs during downtime ICNV tool can be used



A downtime-minimized strategy results in shorter downtime, but has a higher demand on system resources for the parallel operation of both a production and shadow system.

The resource-minimized strategy means no additional system resources are needed during the upgrade, but results in longer downtime and an offline backup is required after the upgrade.

Manual selection Manual selection is also possible for setting the preconfiguration mode parameters. This gives you the flexibility to control all parameters.

5.2.5 Elements of Business Downtime During an Upgrade

The technical upgrade process involves a number of phases, including SAPup running the SAP upgrade application SAPup. However, business downtime is a period of time during the upgrade process that is shorter than the overall upgrade process duration, as you can see in Figure 5.8.



Figure 5.8 Business Downtime During the Upgrade

There are various elements of business downtime during the upgrade process, as follows:

- Business ramp-down and ramp-up, when productive use of the system is stopped and restarted (locking/unlocking users, rescheduling jobs, shutdown of interfaces)
- ▶ Post-upgrade transports and manual adjustments
- Business validation and acceptance testing, when the system is running but not yet available for productive use
- ▶ Pre- and post-upgrade system backups

5.2.6 Factors that Influence Upgrade Runtime and Downtime Duration

Runtime and downtime during an upgrade depend on many different factors, which can be broadly classified as hardware, software, configuration, and strategy.

Downtime factors Downtime is influenced by the following factors:

Hardware

- Number and type of CPUs for the central application and database servers
- ▶ Type and performance of the storage medium (I/O throughput)

► Software

- Start release and target release
- Version of the upgrade tools
- Configuration
 - ► Upgrade parameterization (e.g., the number of parallel processes, instance profile parameters, and database parameters)
 - Number of clients (e.g., 100, 010, and 200)
 - Number of installed languages
- Strategy
 - ▶ Upgrade strategy: downtime-minimized or resource-minimized
 - ▶ Usage of Incremental Conversion (ICNV), Customer-Based Upgrade (CBU), or Incremental Upgrade & Unicode Conversion (IUUC)

Runtime factors Runtime is influenced by the following factors:

Hardware

- ▶ Number and type of CPUs for the application and database servers
- ► Type and performance of the storage medium (I/O throughput)

Software

- Start release and target release
- Number of included support packages
- Binding parts of an enhancement package into the upgrade process, and the number of technical usages selected

- Number of modifications on standard SAP objects
- Number of custom objects
- Version of the upgrade tools

Configuration

- Upgrade parameterization (e.g., the number of parallel processes, instance profile parameters, and database parameters)
- Import destination time
- Strategy
 - Usage of ICNV

Note

The size of the database generally has no direct impact on the upgrade runtime or downtime.

In SAP ERP 6.0, SAP has improved the update process to allow for a onestep upgrade to SAP ERP 6.0:

- ► Direct upgrade path from R/3 or mySAP ERP 2004 to SAP ERP 6.0 with the option to include the latest parts of an SAP enhancement package.
- Cost and downtime reduction with no additional implementation steps necessary for enhancement packages if they are embedded with the upgrade.

5.2.7 Incremental Conversion (ICNV)

The primary goal of ICNV is to reduce downtime during an upgrade.

Note

ICNV is only available if you are using the downtime-minimized upgrade strategy.

ICNV is a configurable process that can be can be stopped and restarted. It allows for the conversion of large tables during system uptime. Furthermore, you can select the specific tables to be processed by ICNV. ICNV requires additional resource usage of the database, as well as a sufSAP ERP 6.0: Software lifecycle improvements ficient number of background work processes. It is also recommended that you execute ICNV as early as possible. This requires more careful planning.

The procedure is designed to not be overly complicated and is fully integrated into the upgrade process. The conversion process is executed during uptime. The database load is expected to be higher during this process and therefore, it is possible to define exclusion times during which no ICNV processes are running.

Progress prediction The conversion process calculates the estimated end of the process. This information helps you plan the upgrade timings accurately. Large tables are converted during uptime but the switch to the new structure is made only during downtime (in PARCONV).

Configuration

ICNV offers several features to configure the incremental conversion process:

- Batch hosts can be specified.
- The number of running batch processes is adjustable.
- Exclusion times for processing can be specified for each table. (This enables you to run conversion jobs at times with relatively low table I/O.)
- ► The log files of the conversion processes for each table can be accessed.

After selecting the tables, you can choose to be guided through the necessary steps by the ICNV Assistant (see Figure 5.9).

For the upgrade scenario, two steps must be started manually:

Initialization

- Extension by a flag field
- Building an index on the flag field
- Creation of an update and deletion of triggers
- ▶ Replacement of the table by a view and renaming of the table
- Start of the data transfer

The remaining steps (transfer, switch, and delete entry in ICNV) are then performed by SAPup.

Control E	idit <u>G</u> oto Mi <u>o</u>	gration Viet	w Utilities(M) Syste	em Help SAP
Increme	ntal conve	rsion		
261	• • • •	Ð 昱 &	6] H	
Progress Info	Monitor for Incremental Table Conversion Progress Information There was not yet a data transfer Compute Progress			
Vorklist No. of Ta	bles 2 /No.	of Process	ses 0	CrICNV Assistant
Table	Proc.	Product.	State	- [Step 1]
MARA USR02	Hold	✓ Y	For conversion For conversion	Initialization
				Step 2 Step 2 Data Transfer
				Step 3 Switch Documentation
				Step 4

Figure 5.9 ICNV Assistant

5.2.8 Customer-Based Upgrade (CBU)

A customer-based upgrade (CBU) is a special upgrade procedure that, compared with the standard procedure, can significantly reduce system downtime when you upgrade a production system.

SAP recommends a CBU when the time needed to implement customer transports means that the upgrade cannot be performed within the upgrade downtime window that is available.

CBU can be used to create and perform optimized, customer-specific upgrades.

- CBU criteria A customer-based upgrade is intended for systems with extensive customer developments, a large volume of delta transports, and where there are high demands on system availability. The goal of a CBU is a significant reduction of business downtime with the inclusion of all requested package contents. This can be achieved through the creation of repository export data at a customer site from an upgraded copy of the production system. During a CBU, a customer-specific export of the substitution set for the upgrade takes place using a special upgrade procedure at your site. This individual export and the special upgrade tools for the CBU remove the need for the following actions when you upgrade the production system:
 - Import of customer transports after the upgrade (with the exception of Customizing transports)
 - Modification adjustment (transaction SPAU)
 - ► ABAP load generation after the upgrade

However, as with many tools, there are prerequisites to be met before you can implement a CBU successfully:

- You must create a copy of the production system.
- ► After you have made the first copy of the production system, you must freeze the repository of the production system. Any transports and corrections you make in the production system after you make this copy are lost when you perform the CBU upgrade and need to be repeated manually later.
- The hardware you use to upgrade the second copy must be similar to the hardware of the production system. This is required for the platform-specific load generation and allows for a reliable estimate of the production upgrade downtime.
- The CBU is a customer-specific upgrade approach changing the standard upgrade procedure in some aspects. Therefore, it is strongly recommended to involve experienced consultants certified for this methodology, to minimize risks.
- CBU advantages The advantages offered by a CBU can be significant. In general, a CBU allows you to plan the upgrade of the production system more precisely. You can also analyze and, if necessary, avoid or optimize critical or long-running database modifications.

5.2.9 Unicode Conversion

Unicode is state-of-the-art technology for technical language support for all SAP solutions. It is the only future solution to support complex systems and system landscapes in a global, multinational, and multilingual environment. Multiple-Display Multiple-Processing (MDMP) configuration is no longer supported with SAP ERP 6.0.

SAP supports four paths to convert your system to Unicode depending on the version of SAP R/3 from which you are upgrading:

- Unicode conversion independent of an upgrade project (if upgrading from 4.7 or higher).
- ► Independent Unicode conversion before or after the upgrade (if upgrading from 4.7 or higher).
- Combined Upgrade & Unicode Conversion (if upgrading from SAP R/3 4.6C or higher).
- ► Twin Upgrade & Unicode Conversion (if upgrading from lower than 4.6C).

Section 3.1.4 "Unicode," of Chapter 3, Planning an Upgrade Project, describes the different approaches to a combination of upgrade and Unicode conversion. SAP provides guides that describe the detailed processes and procedures involved in Unicode conversion, with or without an upgrade. Extensive information on Unicode conversion is also available in the *SAP NetWeaver Application Server Upgrade Guide* by Bert Vanstechelman, Mark Maergerts, and Dirk Matthys (2nd edition, SAP PRESS 2007) and in *Unicode in SAP Systems* by Nils Bürckel, Alexander Davidenkoff, and Detlef Werner (SAP PRESS 2007).

In the context of an upgrade project, upgrading and converting to Unicode in the same project affects the runtime and downtime of the overall upgrade. It is therefore important to consider ways to minimize the runtime and downtime of the Unicode conversion as part of the upgrade process. All Unicode conversion paths follow the same basic steps during downtime:

- 1. Prepare the non-Unicode system.
- 2. Export the non-Unicode database and convert it to Unicode data.

Unicode conversion paths

- 3. Create a new Unicode database and import the Unicode data.
- 4. Perform post-conversion activities in the Unicode system.

Within any Unicode conversion (either independent or together with an upgrade), you can use the Near Zero Downtime approach for significant downtime reduction. This is described in the next section.

5.2.10 Near Zero Downtime

For customers with extremely high system availability requirements, SAP offers the Near Zero Downtime approach. It is a combined approach using both standard SAP upgrade tools and SAP migration tools and, in the case of a release upgrade, allows reducing business downtime to two to four hours.

Production With the Near Zero Downtime method, the upgrade is performed on a clone of the production system. During this upgrade, all changes to the production system are logged at the database level. After the completion of the upgrade on the clone, the real downtime begins. During this downtime, the data changes from the production system will be transferred to the clone. The data transfer is performed by the tool based on the Migration Workbench. The tool will also transform the data structure in case of changes in the data model between the old and the new release. After system validation, the clone takes over the role of the production system. See Figure 5.10 for an overview of the process.

The method can also be used for an upgrade combined with Unicode conversion. In this case, the business downtime can be reduced to five to seven hours, independent of the database size.

- Advantages Compared to a standard upgrade, the Near Zero Downtime process has both advantages and disadvantages. The advantages of Near Zero Downtime are as follows:
 - Greatly reduced downtime (approximately four hours), independent of the database size.
 - Improved system availability.
 - Reusability for the implementation of support packages, enhancement packages, operating system patches, and database patches.

- ► Late go/no-go decision and very fast reset of the system.
- The standard upgrade project remains unaffected.



Figure 5.10 Near Zero Downtime Process

The disadvantages of Near Zero Downtime are as follows: Disadvantages

- Additional project effort and a longer upgrade project (minimum of six months).
- Additional hardware requirements.
- Code freeze for four to six weeks prior to go-live.
- Restricted transport for one to two weeks prior to go-live.

5.2.11 Unicode Conversion Downtime

For Unicode conversion, you can follow various recommendations to minimize downtime during the conversion process.

The two main options for downtime reduction in Unicode conversion Downtime reduction in Unicode conversion Downtime reduction of unused data) and to parallelize the conversion process. Downtime is highly dependent on hardware performance; therefore, hardware tun-

ing (e.g., through additional CPUs and increasing I/O throughput) is recommended.

Estimating You can use SAP resources to get information on estimating downtime, for example see SAP Note 857081 "Unicode conversion: downtime estimate." This addresses expected downtime, potential bottlenecks, possible measures for improvement, how to analyze test results, and how to compare results of different migration projects.

SAP UnicodeThe SAP Unicode Downtime Minimization service is recommended byDowntime Optimization ServiceSAP when you run or plan to run a Unicode conversion project with a
large database and want to select the best option to minimize downtime.

The SAP Unicode Downtime Optimization service (which is run as a workshop) investigates all available downtime minimization options and aims to recommend the best one for your situation.

5.2.12 Best Practices – Upgrade Tuning

This section details several best practices for minimizing downtime during an upgrade project. Upgrade tuning can minimize both upgrade runtime and downtime. In planning an upgrade, you can take steps to reduce the total upgrade time, but there is a trade-off between cost and time reduction. For example, you can invest in faster CPUs and better storage, use new backup tools, and implement automated testing tools to help with the go/no go decision, all of which can help minimize downtime but affect the overall cost of the upgrade.

Tuning of a standard upgrade

f a You can carry out upgrade tuning actions within a standard upgrade (as described in the SAP ERP 6.0 upgrade guides). These will have an effect on the phases of the upgrade: prepare, upgrade uptime, upgrade down-time and follow-up activities. In general, the following will help reduce the amount of upgrade runtime and downtime:

- Using the latest upgrade software tools
- ► Using the latest software release DVDs (as available from SAP)
- Using the downtime-minimized strategy

However, you can take additional specific measures to reduce down-time:

Include all required support packages into the upgrade.
 Downtime is reduced if support packages are included in the upgrade project rather than being imported after the technical upgrade. Also, it may be a requirement to include support packages in the upgrade.

- Include all required technical usages from enhancement packages into the upgrade, and using the SAP Enhancement Package Installer. This drastically reduces the duration of technical downtime. Embedding the SAP enhancement packages within the upgrade to SAP ERP 6.0 requires only one upgrade window and therefore only one period of downtime (embedding enhancement packages in the upgrade is only possible from SAP ERP SR3 on).
- ► Use transport(s) created for automatic modification adjustment (SPDD/SPAU).
- ▶ Use Incremental Conversion (for details, see Section 5.2.7).
- Use as many parallel upgrade processes as possible.
- Create a well thought out cutover plan.
- Consider a backup strategy.

You can also employ additional upgrade tuning measures such as hardware improvements, SAP tools and services, and cutover planning that are not described as part of a standard upgrade in the upgrade guides.

However, when considering to apply any of these measures, you should run at least three test upgrades. It is recommended that the first time you test the upgrade you should run a standard upgrade, following the procedures described in the SAP ERP upgrade guides. The results you get from this will give you a baseline. You can then run a "tuned" upgrade, using one or more of the tuning measures described in the text that follows and compare the results with those of the standard upgrade baseline. You should then run an additional tuned upgrade to again determine whether the tuning steps are effective. It is not worthwhile to pursue a tuned upgrade if the standard upgrade produces acceptable results in terms of downtime.

Additional tuning measures

5.2.13 Testing

It is highly recommended that you perform as much testing as possible before finally embarking on the actual upgrade to determine the duration of the upgrade runtime. This is essential to your planning of the upgrade. There are techniques you can use to reduce the runtime and therefore you should perform pre-upgrade tests to discover how to tune the upgrade process.

Test upgrade For example, a test upgrade on a copy of the production system and a thorough analysis of the upgrade log files can identify many possibilities for effective manual tuning activities. Therefore, a highly recommended preparation for achieving an optimal reduction of production system downtime is to perform multiple upgrade tests, including manual tuning measures.

One of the main purposes of running test upgrades is to measure downtime and to find out if the upgrade can take place within the downtime window available to you. This section suggests ways to minimize downtime as much as possible if you are not able to achieve the desired downtime. For example, understanding and analyzing the factors that influence downtime will enable you to adapt your environment appropriately with the goal of minimizing both downtime and runtime.

Further downtime
minimizationFurther downtime minimization—beyond standard upgrade technol-
ogy-can also be achieved by using specific SAP services, such as the
SAP Downtime Assessment Service and the Customer-based Upgrade
service, as well as the Near Zero Downtime approach.

Note

Because each system is highly individual regarding its configuration and application data, a forecast of runtime and downtime is only possible when analyzing the results of a test upgrade with a representative set of data for your organization.

5.2.14 Splitting Downtime

SAP usually releases older versions of SAP ERP software on database and operating system versions higher than those used initially. This enables

customers to run an SAP release on the latest database and operating system versions, if required. Thus, the maintenance periods of SAP releases can be prolonged. During the upgrade of an SAP/database/operating system combination, this feature may be used to split the system downtime, for example, into two different weekends:

- 1. Perform the upgrade of the database and operating system to the version required by the SAP ERP target release. It is recommended that you do this at least four weeks before the SAP ERP upgrade.
- 2. Upgrade to SAP ERP.

5.3 Training

This section contains recommendations and options for training should consider during an upgrade project. This can include training for the following:

- The administrator, to perform the technical upgrade.
- ► Developers, to handle modification adjustments, re-implementation of custom developments, and new development on the new release.
- The project team, to handle the upgrade project.
- Power users, to cover delta and new functionality in the new release and to perform delta Customizing.
- End-users, to cover delta and new functionality in the new release.

Within all phases of the project potential requirements for training exist. Relevant, targeted training is important to ensure the smooth running of the entire upgrade project and the adaptation to new processes and functionality. You should devote an appropriate portion of the upgrade project budget to training needs.

For upgrades from source releases SAP R/3 4.6C on, training efforts and costs typically amount to no more than 5% of the project budget. For additional information, see the (unnumbered) Section "Cost and Effort Factors of an Upgrade Project," in Chapter 3, Planning an Upgrade Project.

Index

A

Add-ons 63 Application-Specific Upgrade Toolbox 218 ASAP 109 ASU Toolbox 218 Authorizations 84 Automated testing 131

В

Best practices 149 BPCA \rightarrow see Business Process Change Analyzer Business *Case* 29, 31, 32, 40 *Function* 24, 26 *Process platform* 38 *Value* 48 Business downtime 171 Business function 137, 237 Business Process Change Analyzer 135, 223 Business process expert 120 Business Suite 17, 42

С

Code freeze 155, 160
Combined Upgrade & Unicode Conversion 76, 148, 181
Communication 151
Complexity drivers 58
Costs 45
CRM → see Customer Relationship Management
Custom development 47, 81
Custom Development Management Cockpit 224
Customer Relationship Management 17, 28

Customer-Based Upgrade 176 Customer-based upgrade 176, 179 Customizing 81 Cutover plan 140, 144

D

DEV system 165 Developer test 125 Developers 120 Documentation 39, 152, 155 Downtime 154 *Minimization* 169 *Planned* 171 *Unplanned* 171 Downtime-minimized strategy 172

Ε

Early Watch Alert 231, 250 $eCATT \rightarrow see Extended Computer Aided$ Test Tool $EHP \rightarrow see Enhancement package$ E-learning 193 Enhancement package 25, 26, 37, 47, 70,237 4 22 Installation 70 Maintenance Optimizer 236 Testing 136 Enhancement Package Installer 185, 239 Enterprise services 19, 38 Bundle 27 Enterprise Support 42 Enterprise support services 248 Extended Computer Aided Test Tool 232 Extended maintenance 42

F

Final preparation for cutover 214 Frontend software 64 Functional test 124, 125, 127, 128 Functional upgrade 29, 36, 37, 65

G

GoingLive functional upgrade check 249

Н

Hardware requirements 68

I

ICNV \rightarrow see Incremental Conversion ICNV tool 174 Incremental Conversion 176, 177 Incremental Upgrade & Unicode Conversion 176 Industry 24 *Add-ons* 24 Industry solutions 83 Integration test 125, 126, 127 Interfaces 64 IUUC 176

J

JSPM 235, 240

Κ

Key users 120

L

Learning Map Builder 214 Load test 124, 128

Μ

Mainstream maintenance 42 Maintenance 41, 42 Strategy 42 Maintenance Optimizer 235 MDMP \rightarrow see Multiple-Display Multiple-Processing Meetings 150 Modifications 82, 102 Module test 127 MOPZ \rightarrow see Maintenance Optimizer Multiple-Display Multiple-Processing 60, 76, 181

Ν

Near Zero Downtime approach 182 NetWeaver 18 NetWeaver Composition Environment 38

0

Offshore resources 89, 128

Ρ

Performance test 124, 128 PLM 17 PRD system 166 PREPARE 218 Procedures 109 Product Availability Matrix 67 Product Lifecycle Management 17 Production cutover 214 Project Duration 92 Management 117 Manager 119 Methodology 109 Schedule 92,98 Sponsor 119 *Team* 118 Project management structure 108

Index

Q

QAS 166 QUA \rightarrow see Quick upgrade analysis Quality gate 139 Quality management 139 QUE 245 Quick Sizer 68, 220 Quick upgrade analysis 33, 243 Quick upgrade evaluation 245

R

Regression test 126, 127 Resource-minimized strategy 172 Resourcing 86 Resourcing model 86 Risk classification 113 Risk management 110 Risk-impact-analysis 111 Rolling wave approach 99 Runtime 171

S

Safeguarding for upgrades service 250 SAINT 235, 239 Sandbox system 32, 154 *Upgrade project system* 162 SAP Business Suite 17, 42 SAP ERP Upgrade Course Finder 191 SAP LoadRunner by HP 234 SAP Notes 152 SAP Quality Center by HP 132, 134, 233 SAP Test Acceleration and Optimization 234 SAPehpi 235 SAPjup 214, 216, 235 SAPup 175, 214, 235 Scenario and process component list 60, 223 Scenario test 125, 127 Scheduling 96 SCM 17, 28 SDTGUI 216 Security 84

Service-oriented architecture \rightarrow see SOA SOA 18, 38 Solution Browser tool 25, 31, 39, 219 Solution Documentation Assistant 224, 230 Solution Manager 132, 133, 134, 149, 201.224 Source release 63 $SP \rightarrow see Support package$ SRM 17 Stakeholders 119 Standards 109 Strategic upgrade 36, 65 Stress test 124, 128 Supplier Relationship Management 17 Supply Chain Management 17 Support package 26, 28 Switch Framework 24, 28, 73 System landscape 59, 161 System switch technology 172

Т

TAO 234 TCO 35, 37, 38, 43, 46 Model 43, 44 $TDMS \rightarrow see Test Data Migration Server$ (TDMS) Technical downtime 171 Technical test 126 Technical upgrade 29, 36, 46, 65 Cost 106 Technical upgrade planning 245 Technical upgrade service 248 Test Acceptance test 120 Automation 130 Coordinator 120 Log 131 System 128 Test case 130 Catalog 136 Template 136 Test Data Migration Server (TDMS) 129, 227 Test Organizer 232

Test Workbench 231 Testing 122, 153 *Costs* 122 *Integration testing* 120 *Success factors* 138 *User acceptance test* 120 Testing services 250 Testing tools 224 Three-system landscape 61 Training 152, 187 Trial upgrades 100 Twin Upgrade & Unicode Conversion 181

U

Unicode 68, 75 Unicode conversion 60, 74, 147, 181 *Independent* 181 Unicode Downtime Minimization service 184 Unit test 124, 125 Upgrade *Blueprint* 212 Upgrade (cont.) Coaching 246 Justification 31, 55 Realization 213 Services 241 Value assessment 32, 55, 242 Upgrade assessments 101 Upgrade dependency analyzer 60, 221 Upgrade Experience Database 122, 153 Upgrade GUI 215 Upgrade project system 162 Upgrade Road Map 109, 133, 134, 149, 206 Upgrade Scoping Evaluations 101 Upgrade strategy 64 Upgrade tuning 184 Uptime 171 User acceptance test 126, 127

۷

Value Determination 50 Proposition 31, 32 Volume test 124