## Typographic Conventions

<table>
<thead>
<tr>
<th>Type Style</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Example</td>
<td>Words or characters quoted from the screen. These include field names, screen titles, pushbuttons labels, menu names, menu paths, and menu options. Textual cross-references to other documents.</td>
</tr>
<tr>
<td>Example</td>
<td>Emphasized words or expressions.</td>
</tr>
<tr>
<td>EXAMPLE</td>
<td>Technical names of system objects. These include report names, program names, transaction codes, table names, and key concepts of a programming language when they are surrounded by body text, for example, SELECT and INCLUDE.</td>
</tr>
<tr>
<td>Example</td>
<td>Output on the screen. This includes file and directory names and their paths, messages, names of variables and parameters, source text, and names of installation, upgrade and database tools.</td>
</tr>
<tr>
<td>Example</td>
<td>Exact user entry. These are words or characters that you enter in the system exactly as they appear in the documentation.</td>
</tr>
<tr>
<td>&lt;Example&gt;</td>
<td>Variable user entry. Angle brackets indicate that you replace these words and characters with appropriate entries to make entries in the system.</td>
</tr>
<tr>
<td>EXAMPLE</td>
<td>Keys on the keyboard, for example, ( F2 ) or ( \text{ENTER} ).</td>
</tr>
</tbody>
</table>
## Document History

<table>
<thead>
<tr>
<th>Version</th>
<th>Date</th>
<th>Change</th>
</tr>
</thead>
<tbody>
<tr>
<td>1.0</td>
<td>2015-12-12</td>
<td>First version created</td>
</tr>
</tbody>
</table>
# Table of Contents

1. **SAP Standards for E2E Solution Operations** ................................................................. 6  
   1.1 Control Center Approach ............................................................................................ 7  

2. **Overview of the Standard for Upgrade Management** .................................................. 10  
   2.1 Upgrade Management in the Application Lifecycle ....................................................... 11  
   2.1.1 Release Management ............................................................................................ 11  
   2.2 Types of Solution Changes ......................................................................................... 12  
   2.3 Key Focus Areas of Upgrades .................................................................................... 13  
   2.3.1 Program Definition ............................................................................................... 14  
   2.3.2 IT Infrastructure Planning ..................................................................................... 16  
   2.3.3 Application Adaptation ......................................................................................... 18  
   2.3.4 Test Management .................................................................................................. 22  
   2.3.5 Technical Change Management .......................................................................... 22  
   2.3.6 Business Downtime .............................................................................................. 25  
   2.3.7 Preparation for Go-Live ....................................................................................... 32  
   2.3.8 Protection of Investment ..................................................................................... 34  
   2.4 Upgrade Support Tools ............................................................................................. 35  
   2.4.1 Scope and Effort Analyzer ..................................................................................... 36  
   2.4.2 Work Centers ....................................................................................................... 36  
   2.4.3 Project Management Tools ................................................................................... 36  
   2.4.4 Solution Documentation Tools ............................................................................. 37  
   2.4.5 Innovation Discovery ............................................................................................ 37  
   2.4.6 Upgrade Dependency Analyzer ........................................................................... 37  
   2.4.7 Custom Code Management .................................................................................. 38  
   2.4.8 Test Management Tools ....................................................................................... 38  
   2.5 Basic Architecture ..................................................................................................... 39  
   2.6 Roles and Responsibilities ......................................................................................... 41  

3. **Lifecycle of Upgrade Management** .............................................................................. 42  
   3.1 Plan .................................................................................................................... 43  
   3.1.1 Roles .................................................................................................................... 43  
   3.1.2 Upgrade Discovery ............................................................................................... 44  
   3.1.3 Upgrade Evaluation ............................................................................................. 45  
   3.2 Build .................................................................................................................... 45  
   3.2.1 Roles .................................................................................................................... 45  
   3.2.2 Project Preparation .............................................................................................. 47  
   3.2.3 Upgrade Blueprint ............................................................................................... 47  
   3.3 Run ..................................................................................................................... 48  
   3.3.1 ERP Upgrades and Experience Database ............................................................. 48  
   3.3.2 Evaluation of Upgrade Experience ...................................................................... 49  
   3.3.3 Upgrade Realization ............................................................................................. 50  
   3.3.4 Preparation for Cutover ....................................................................................... 52  
   3.3.5 Production Cutover and Support ....................................................................... 52  
   3.4 Optimize .................................................................................................................... 53
1 SAP Standards for E2E Solution Operations

IT organizations face new challenges every day as they attempt to remain effective and future safe while also keeping costs for day-to-day operations as low as possible. They are also being challenged more than ever to demonstrate their value to businesses. Therefore, it is important to optimize the day-to-day tasks that have less obvious business value and to use KPI and benchmark-based reporting to make IT processes more visible, demonstrating the real value that IT can provide.

In order to minimize the costs of IT, it is necessary to standardize and automate IT processes end-to-end (E2E) without reducing the SLAs required by the business, such as stability, availability, performance, process and data transparency, data consistency, IT process compliance, and so on.

Based on the experience gained by SAP Active Global Support (AGS) while serving more than 36,000 customers, SAP has defined process standards and best practices to help customers set up and run E2E solution operations for their SAP-centric solutions.

The Build phase of SAP best practices supports a Build SAP Like a Factory approach, consisting of the following processes:

- Custom code management
- Change, test, and release management
- Incident, problem, and request management
- Solution documentation
- Remote supportability

During the Run phase of a solution, adapting your IT infrastructure to a Run SAP Like a Factory operation impacts both application operations and business process operations. Therefore, operations processes, such as technical monitoring, end-to-end root-cause analysis, technical administration, and data volume management need to be optimized to achieve state-of-the-art application operations. In business process operations, the same applies to business process and interface monitoring (including performance optimization), data consistency management, and job scheduling management.

Quality management processes and tasks need to be established throughout the lifecycle to guarantee continuous improvement of the end-to-end operations processes while simultaneously ensuring the flexibility needed to react to changing requirements.
This figure shows an organizational model for solution operations that aligns SAP best practice topics and E2E standards with SAP’s control center approach.

The Operations Control Center executes and controls the Run SAP Like a Factory processes, while the Innovation Control Center ensures optimal custom code management and a smooth transition to production with integration validation procedures. SAP connects to these control centers from the Mission Control Center to ensure that professional support is available to the customer. The following Application Lifecycle Management (ALM) functions are not provided directly in one of the control centers because they must be handled across different areas:

- Change, test, and release management
- Incident, problem, and request management
- Solution documentation
- Remote supportability

The quality management methodologies are an essential part of SAP’s Advanced Customer Center of Expertise (CoE) concept and ensure that the KPI-driven processes are continuously improved across all processes and teams. In addition, the quality manager roles ensure consistent and value-centric reporting to the business and management. This unified reporting platform is known as the Single Source of Truth.

### 1.1 Control Center Approach

The Operations Control Center (OCC) is the physical manifestation of the Run SAP Like a Factory philosophy. The OCC allows for automated, proactive operations, which simultaneously reduces operational costs while increasing the quality of IT services, leading to improved business satisfaction. The OCC also drives continuous improvement of business processes and IT support. To achieve these goals, it relies on a close interaction with both the Innovation Control Center (ICC) and the SAP Mission Control Center (MCC).
The OCC is a central IT support entity at the customer site, which monitors the productive SAP environment as well as important non-SAP applications. During operation, the OCC requires a workforce of 2 full-time equivalent (FTE) per shift to ensure that incidents are detected and resolved as quickly as possible. The OCC is equipped with large screens that display the status of business processes, IT landscape components, as well as exceptions and alerts. If problems occur, you use a video link to get live support from SAP and partners. The customer usually sets up the room with assistance from SAP Active Global Support (AGS). The customer is responsible for managing the OCC and the team of technical and functional IT operators who act on the alerts.

The OCC is most effective when closely integrated with other IT processes, such as IT Service Management (ITSM) and Change Management. Central monitors and dashboards based on application and business process operations display the current status of business and IT-related processes. This data can also be used to drive continuous improvement.

An effective system monitoring and alerting infrastructure is fundamental to the success of an OCC.
The OCC is most effective when closely integrated with other IT processes, such as IT Service Management (ITSM) and Change Management. Central monitors and dashboards based on application and business process operations display the current status of business and IT-related processes. This data can also be used to drive continuous improvement.

An effective system monitoring and alerting infrastructure is fundamental to the success of an OCC. For Job Scheduling Management, the OCC supervises all background monitoring processes, SAP controls and legacy background operations. It reacts to job monitoring alerts according to predefined error-resolution activities, and triggers follow-up activities for error handling if the relevant task are not completed within a certain timeframe.

**Figure 3: OCC Concept**
2 Overview of the Standard for Upgrade Management

Changing business conditions mean changing IT solutions, and every evolving organization needs to adapt its IT environment accordingly. This drives the need to upgrade, update, or enhance your SAP software on a regular basis. An up-to-date system landscape, both in terms of software releases and underlying technology, is an essential prerequisite if you want to benefit from the latest available functionality, leverage new technologies that foster innovation, and guarantee long-term protection of IT investments.

Regardless of what determines the specific business case, an upgrade project, just like any other project, must be completed in scope, in budget, and in time. By implementing the concepts as outlined in this document, you can ensure that when upgrading to the next generation of your SAP solutions, all activities related to the change are performed as efficiently as possible in order to save both resources and budget.

The main focus of the SAP Standard for Upgrade Management is to provide guidance for holistic and effective quality management of the required project from its earliest stages of evaluation until after a successful cut over of the production system. This also includes best practices how to review and ensure upgradeability of a solution already in the implementation or operation phase to protect your investments in a later change project.

This document does not deal with application-specific details of the upgrade. SAP already provides comprehensive upgrade and installation guides for all supported software lifecycle events of its products. Quick links to the respective SAP information sources are provided in the appendix of this document. We strongly recommend reading the latest version of these documents carefully before executing the upgrade and following the provided instructions closely.

Business and management aspects of the project execution, such as creating business cases, budgeting, staffing, and reporting, are also not discussed in detail in this document.

The execution of an SAP upgrade is a relatively uniform sequence of activities; there are fewer variations in the general procedures than in implementations. SAP provides a number of supporting documents, tools, and services that are designed to further decrease the complexity and risk of such projects. The aim of this document is to supplement other resources by providing a focus.

First of all, you need to establish the sequence of activities that you need to perform. SAP Solution Manager provides an upgrade roadmap, which serves as a starting point for this definition. The relevance and impact of each major step is highlighted to help you keep track of the time and effort required. There is a small number of key focus areas that can determine the success or failure of an upgrade project, regardless of your specific landscape. You can use various parameters to identify which areas to focus on throughout all activities.

SAP provides tools and services that support you through all stages of the upgrade project. The central platform for these tools and services is SAP Solution Manager. Knowing these tools and services is crucial for an efficient, fast, and safe transition to the new release. Therefore, this document also outlines the most important ones.

Having the right focus helps you direct resources and attention where and when they are needed most, ensuring that no relevant points are overlooked.
2.1 Upgrade Management in the Application Lifecycle

Within the Customer Center of Expertise team, the quality manager for protection of investment is mainly responsible for upgrading the technology framework and application components of the company’s software landscape on a regular basis. This careful maintenance results in a well-defined, harmonized software landscape that is far easier to upgrade than a software landscape made up of different product releases and unaligned support packages. Another objective of the quality manager is to prevent unnecessary modifications or custom development being introduced into the software landscape. Keeping custom code to a minimum helps reduce development costs in general and upgrade costs in particular.

In addition, the quality manager works closely with the program management office, which oversees all ongoing projects. In the context of upgrade and release management, the quality manager must create a master release plan for the products and solutions currently in place and any that are due to be deployed. The master release plan must be aligned with the company’s overall strategy and constraints, especially budget constraints. It must also be aligned with SAP’s general release strategy. When creating the master release plan, the quality manager should take the following factors into account:

- The company’s own strategic needs, for example, whether there should be stronger emphasis on customer management, integration of supplier systems, or compliance
- Operational issues of the existing solutions, such as functionality gaps or compatibility issues
- End-user feedback on the existing solutions, which may include a demand for change
- Improvements in terms of better usability, functionality, performance, or technological handling contained in a new release
- Expiration of maintenance contracts on existing software requiring an update to a new software release
- The other projects in the company portfolio and their relationship and dependency with the upgrade project. Most of the time, dependencies will exist with respect to timelines and utilization of company resources.

2.1.1 Release Management

To properly manage the planning and deployment of solution changes, your company’s application lifecycle management (ALM) concept should contain an update and release management strategy. The Customer Center of Expertise (CCoE) should be responsible for this talk. The CCoE is a team of quality managers located in the company’s application management unit that is active across all business units. The team sets basic rules that facilitate communication and collaboration between your business and IT departments and aim to bring all stakeholders to the table to resolve challenges and issues.

To address the need of faster innovation and speed, rapid prototyping of new business models and capabilities is required. This is delivered in the framework of Build SAP like a Factory. The new business solution is integrated into the existing solution landscape using industry best practices and building blocks delivered by SAP, known as Rapid Deployment Solutions. All integration issues and perceived gaps are managed in the Innovation Control Center so that any gaps are resolved or closed as quickly as possible.

To bring all these components together, SAP provides a new “Build SAP + Run SAP Like a Factory” methodology with the following objectives:

- Provide a new value release with tangible and measurable benefits for each LOB every six months
- Provide an upgrade to the latest release and technology with near-zero downtime every six months

One key element of an accelerated release strategy is to include the latest SAP software versions (OS/DB version, SAP releases, enhancement packages, and support packages) at the beginning of a major customer release. New
business developments are then automatically based on the latest SAP technology. Combining SAP and customer releases allows you to benefit from latest SAP innovations, legal changes, and software corrections immediately, for example, to improve of the level of security and performance of your SAP systems.

For more information on the “two value-releases per year” concept, see https://support.sap.com/solutionmanager → Knowledge Transfer → White Papers → Two Value Releases per Year.

2.2 Types of Solution Changes

The lifecycle of any IT solution, from the first implementation concept to the final phasing out, can be described as series of business configuration states connected by permitted transitions. While the business scope and scale of each configuration change can vary widely, the management of these changes can be best described as a repetitive lifecycle. SAP uses the Application Management Lifecycle described in ITIL (V3) as a commonly agreed model to guide you through this sequence of business configuration processes.

The following figure outlines the most important change events in the lifecycle of your solution:

![Change Events in the Application Lifecycle](image)

The scale and frequency of the changes and, consequently, the impact on your solution varies depending on the underlying business requirements. In the application lifecycle, we distinguish the following main types of change:

<table>
<thead>
<tr>
<th>Type</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Installation</td>
<td>An installation, also known as the initial implementation, is the completely new setup of a system, mostly on new hardware. Migration might be necessary from former legacy systems. In addition to the initial implementation of an SAP product, the installation of new add-ons or other software components also fall under this category.</td>
</tr>
<tr>
<td>Update</td>
<td>An update is a set of corrections for software errors and severe performance problems in the SAP system. Updates are delivered, for example, as hot-fixes, support packages (SPs), support package stacks (SPSs), or SAP Notes. Support packages are compiled periodically and made available in SAP Solution Manager. This maintenance process aims to correct known errors in applications by minimizing the</td>
</tr>
</tbody>
</table>
Type | Description
--- | ---
impact to any existing landscape elements and running processes. Regression tests are needed and there should be no disruptive changes in user interfaces. New functionality or different behavior of existing processes is not expected.

Enhancement | An enhancement contains a larger number of objects where the main aim is not to correct errors but to enhance features and functions. Enhancements are delivered as enhancement packages (EHPs). New functionality is expected (enabled by switches) but different behavior of existing processes is not. There should be no migration effort necessary. An enhancement changes the version of a software component but not the release.

Upgrade | An upgrade contains all objects of a software release. The main focus is to provide additional functions and features, as well as redesign certain processes. Usually, the same functionality of the previous software is also available within the higher release, which enables you to perform a technical upgrade first. Different behavior of existing processes and a certain amount of migration effort often result from upgrades.

Business Improvements | During a business improvement, new business processes are implemented in an existing system. This may include developing or updating custom programs and customizing, or activating business functions. However, the SAP software release, software component versions or patch levels are not changed.

For clarity and readability, the term “update event” is used in this document for the software change events 2-4 described above. New installations or business improvements are not in scope of this paper.

Note
For clarity and readability, “update event” is used in this document as a catch-all term to describe updates, enhancements, and upgrades. New installations or business improvements are not covered in this document.

2.3 Key Focus Areas of Upgrades

Although the exact details of upgrade-related tasks and the technical upgrade itself vary greatly depending on the release, platform, and interfaces, there are several key focus areas that always need to be considered in detail. This section describes these key areas because properly managing them is a prerequisite for a successful upgrade process.

In practice, most upgrades for SAP applications are technical upgrades, where the intention is to leave the existing functionality unchanged while a new release is applied. Therefore, we concentrate on this specific case for most of the discussion of the key focus areas. If you are implementing additional functionality in parallel to the upgrade, you need to take the relevant SAP E2E Implementation Standard into consideration as well.
2.3.1 Program Definition

This focus area helps you to align upgrades with corporate strategy.

Large corporations usually maintain a number of complex solutions that consist of SAP and non-SAP products running on several productive systems. In most cases, this makes up a heterogeneous environment in which some of the following characteristics may vary over time:

- IT provider concept
- Data center concept
- Server architecture
- Operating systems and databases
- SAP products and versions
- Language support
- Communication structure.

Within those environments, there are always mutual dependencies between the systems. Therefore, it is extremely important to define a global corporate IT program before starting the first upgrade project so that you can perform an effective impact analysis.

The methodology for this type of upgrade impact analysis is as follows:

- Provide full transparency about the current situation
- Collect all information about intended changes
- Put all these facts into the context of upcoming upgrades
- Derive the impact and consideration in preparation for the upgrade

You need to consider the following dimensions of an IT program:

Technical Infrastructure

The installed hardware and software components are the basis of any IT solution. Full and transparent documentation of all these components is crucial for the planning and execution of an upgrade project. All relevant characteristics that describe each of the existing solutions are determined and stored into a central repository. The SAP Solution Manager System Landscape, accessed by transaction SMSY or via the System Landscape Management work center, provides a possible storage location.

In addition to the technical solution components, there are also other important factors to be considered when looking at the technical infrastructure. For example, the provider concept, geographical location, the data center concept, and server architecture are the most important characteristics.

An SAP release upgrade almost always triggers changes at all levels of the vertical software stack and requires at least some additional hardware resources. Therefore, when planning an individual upgrade, IT has to consider the following questions to decide whether IT changes are necessary in parallel to the software upgrade:

- Which changes are intended in the foreseeable future, either driven by the corporate IT program or the local system owners?
- Is a change of the provider model planned and if so, would systems be moved into a hosting scenario?
- Is any geographical relocation planned?
- Is it possible to merge servers into one data center?
- Will global agreements be negotiated with hardware vendors regarding new architectures (new CPU type, blade server, adaptive computing, and so on)
- Will the global policy regarding your operating system or database product change?
These considerations save effort and costs for the individual project and underline the importance of having a well-defined global IT program that is constantly updated in advance of major change events like an upgrade.

**Business Solutions**

A business solution consists of all the business processes and scenarios that use one or more software components of the technical infrastructure. When creating a corporate-level upgrade program, you need to consider the following factors for each system that makes up your business solutions:

<table>
<thead>
<tr>
<th>Factor</th>
<th>Considerations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Business context</td>
<td>• What business areas and scenarios are implemented?</td>
</tr>
<tr>
<td></td>
<td>• How do you rate the criticality and business impact of system failures?</td>
</tr>
<tr>
<td></td>
<td>• What are business requirements that need to be considered?</td>
</tr>
<tr>
<td>Lifecycle phase</td>
<td>Is it in an implementation, continuous improvement, upgrade, or system consolidation phase?</td>
</tr>
<tr>
<td>SAP maintenance situation</td>
<td>What type of contract situation do you have: standard maintenance, extended maintenance, or customer specific maintenance?</td>
</tr>
<tr>
<td>Clusters of dependent systems</td>
<td>• Can you determine which systems have to be treated as a cluster because they exchange data or because business processes run across system borders?</td>
</tr>
<tr>
<td></td>
<td>• Which systems belong to the same set of repository template sender and receivers or take part in a customizing distribution?</td>
</tr>
</tbody>
</table>

The complete documentation of your business solution is an important prerequisite for answering the questions above. Doing so helps you to understand the dependencies in your systems and solutions. Dependencies may have an impact on timing and sequencing of upgrades or the determination of the target release. They will usually increase the complexity of end-to-end integration tests. Or they require specific planning and preparation for the productive downtime of one of the systems in a cluster.

Beside these more software-related dependencies of your business solution, it is also crucial for the success of an upgrade project to get the commitment and involvement of the business departments as early as possible when planning an upgrade. In many cases, business requirements are the main driver for upgrades of an SAP solution. In these cases especially, early involvement of business is an essential prerequisite for a successful upgrade project. But even if IT or maintenance benefits justify an upgrade, business needs to already be involved during the planning phase to avoid complications during project execution. In fact, it is the business that has to finally sign-off on the upgrade.

More often than not, upgrade projects that are completed successfully on time and in budget are those in which both IT and the business were involved from the beginning.

**Solution Operations**

An upgrade project relies on well-established solution operations. Any intended change is both a potential risk for the robustness of operations while also providing potential benefits. Some changes may be a precondition for a successful upgrade. Alternatively, an upgrade might be the perfect opportunity to use an improved tool or process and return the first payback on a previous investment. Therefore, it is important to perform an impact analysis for solution operations as early as possible.

Solution operations comprises the people and their assignment to roles, the solution support processes that comprise all relevant operational tasks, and a platform that facilitates the processes and provides the tools that
help people to fulfill their tasks. You have to consider the following impacts and dependencies related to solution operations when planning an upgrade:

- **People**
  You need to clearly define the assignment of roles, descriptions of available skill sets, whether there is potential to invest in these skill sets. Decide whether knowledge pools or centers of expertise are necessary and what skill changes are required to manage the new solution. It is also important to identify whether there are bottlenecks of resources with a certain skill set.

- **Processes**
  You also need to clearly define the established processes, their scalability, and robustness.
  If you plan to change or add business scenarios or introduce new technologies, be aware that the impact to solution operations must be carefully analyzed.

- **Platform**
  You need to identify which tools have been deployed and whether you intend, or are required to, change some of them. In addition, identify and scenarios that might require investment.

**Summary**

An upgrade project cannot focus solely on the systems to be upgraded. Dependencies may exist on various technical and organizational levels and might not be visible in the first place.

Certain questions and conflicts can only be resolved on a strategic program level. Therefore, these topics need to be aligned with the corporate IT strategy as soon as possible before the upgrade project starts.

Upgrade projects are not only IT projects. An upgrade has an impact on the whole corporation. Therefore, business has to be involved from an early stage. Experience shows that building a solid business case is a key success factor of any upgrade project.

In case of large solution landscapes requiring several upgrades of SAP software, a special pool of in-house upgrade experts should be set-up to create a center of excellence. This will leverage the knowledge and experience in all projects.

### 2.3.2 IT Infrastructure Planning

This focus area helps to ensure the compatibility and capacity of your systems

A proper infrastructure plan is a prerequisite for a successful upgrade realization as well as for a reliable and well-performing target solution. In this section we discuss the major challenges of this task, which are ensuring the compatibility of all connected software components and the correct sizing of the hardware needed for the future solution.

**Component Compatibility**

Every SAP system relies on a stack of technical software components to run. Frequently called the vertical or technology stack, it is mainly comprised of the operating system, the database, the SAP kernel, the SAP application, add-ons, and the user interface. Between all those elements of the stack, there are compatible combinations and restrictions. By the time a component is developed, it is developed and tested in an environment made up of the components and its versions available at that point in time. After completion, additional combinations may be tested on top. But the number of variants of components is limited.
For more information about the released combinations, see the Product Availability Matrix on SAP Service Marketplace at http://service.sap.com/pam. For SAP add-ons, you should also check the respective release restriction and add-on release strategy SAP Notes.

You also need to consider the compatibility between different SAP and non-SAP systems connected in the solution, which can be referred to as the horizontal stack. In general, different SAP systems can communicate to each other independently of their respective release. But in some specific cases, complex restrictions apply depending on certain criteria. For example an Employee Self-Service (ESS) business package running on an SAP ERP backend and a SAP NetWeaver Portal may require dedicated releases on both ends depending on the used self-service scenario.

The Upgrade Dependency Analyzer (UDA) (http://service.sap.com/uda) helps you analyze known dependencies when upgrading technical components in their landscape. This tool provides dependency information to support the planning of upgrade projects. Starting with SAP Business Suite 7 and SAP NetWeaver 7.0, SAP has set up a stronger governance process to ensure the version interoperability within a system landscape. For more information see SAP Note 1388258.


A special challenge arises if a Unicode conversion is necessary. SAP NetWeaver releases 7.40 or higher and products based on it are Unicode only. Upgrades of non-Unicode systems to releases higher than SAP NetWeaver 7.40 without prior Unicode conversion will not be supported, for more information see https://service.sap.com/~sapidb/01200252310000995832014E and SAP Note 2033243.

When converting Unicode, characters have to be properly encoded and exchanged between all systems connected to the system that is converted. The most complex situations exist when Unicode systems communicate with systems using SAP-proprietary multi-display/multi-processing (MDMP) or blended code pages. Such communication requires special consideration and maintenance effort and, therefore, should be avoided if possible.

Generally, the business impact of converting systems to Unicode should be taken into consideration during project planning. In this context, the ordering of Unicode conversions is important and some systems need to be grouped together for conversion. When installing additional languages in a Unicode-converted system, the data exchange with non-Unicode systems needs careful analysis. Additionally, the compatibility with non-SAP software and peripheral equipment like printers or fax machines is a critical aspect that is often forgotten. All of these business impacts are solved once the whole business landscape is running on Unicode, which is the target state recommended by SAP. For more information on Unicode-specific topics, see the SAP Service Marketplace at http://service.sap.com/unicode.

Finally, you have to check the compatibility for connected non-SAP products or add-ons to SAP products. It is important to verify with the respective vendors whether their software is released and certified with the planned SAP target release and what adjustments or changes are required.

**Technical Component Sizing**

Different code needs different hardware. As there is usually a lot of different code, the additional hardware required can also be quite considerable. The larger the gap between your current release and the target release, the more additional hardware is required.

Determining the right size of all relevant system resources is a crucial aspect of the upgrade. Only this way you can achieve a performance on the established levels with no bottlenecks.
SAP provides the SAP Continuous Quality Check (CQC) Going Live Support as part of SAP Enterprise Support to conduct a plausibility check of the sizing of the target system. This may include upgrade as well as Unicode sizing. For details, see the SAP Service Marketplace at http://service.sap.com/goinglive-fu.

You can minimize the extra IT investment and related costs by performing housekeeping tasks, such as data volume management and performance optimization, before attempting to upgrade your solution. For more information, see the SAP Standard for Data Volume Management.

Summary

A compatibility check of all software components and the proper sizing of the hardware of a solution are important tasks to ensure a successful upgrade project and a smooth solution transition. These tasks have to be addressed early in the planning phase of an upgrade project as they may have an impact on the overall IT program definition and the creation of a business case for the upgrade.

Use the SAP Product Availability Matrix to check OS/DB compliance, platform requirements, availability of country versions and languages, or SAP solution extensions by partners.

Use the SAP Upgrade Dependency Analyzer to get a high-level overview of possible conflicts between different systems.

Data Volume Management and performance optimization are housekeeping activities that can have a positive impact on the required additional hardware resources. These activities have to be considered and completed in advance of an upgrade project.

If a new SAP GUI version is required, start the distribution before the upgrade project.

2.3.3 Application Adaptation

This focus area helps you to understand Development Adjustment Requirements

Adjustments to the existing custom developments are always required when new SAP software versions are installed. Even when the intention of the maintenance project is that everything works as before, customizing, coding, and interfaces have to be reviewed. With the introduction of enhancement packages, this challenge has reduced significantly compared to classical release upgrades because most changes, such as UI, process, or data model changes, are not immediately active for the users. Nevertheless, thousands of lines of SAP code could be changed within the system, for example, in the SAP NetWeaver basis layer, or because of support package corrections. On the other hand, nobody wants to redo the whole implementation.

The key is to set the right focus and to invest the development resources where it really matters. The first step to accomplishing this goal is to compile a list of all repository objects, customizing settings, and business process steps that are affected by an upgrade or maintenance. SAP provides tools to automate this step. You usually analyze modifications with the SPDD (DDIC objects), SPAU (repository objects) or SPAU_ENH (impact to enhancements) tools. Additionally, for planning and preparation purposes, you can use tools such as the Modification Browser (transaction SE95).

Developments in the customer namespace are not directly affected when implementing new SAP software versions. However, custom development objects which work correctly in one release may not work in an upgraded version. There are a variety of reasons for this. Most important is the fact that custom code usually contains static or dynamic references to SAP objects. If they are changed, the impact has to be reviewed. In particular, if custom transactions of executable reports have been created as copies of SAP programs, these cloned objects present unique challenges after a software change.

To identify such critical changes to custom code objects, perform syntax checks using the ABAP Test Cockpit (ATC) or the Code Inspector (transaction SCI), at least for the most important and critical custom developments.
as soon as an upgraded version of these programs exists, for example, after a test upgrade. Additionally, the Custom Development Management Cockpit (CDMC) offers an upgrade impact analysis. It compiles a list of custom objects with a reference to an SAP object which will be changed by the upgrade. The custom objects in this list can be better classified by upgrade impact, and adjustments and testing can be better planned.

If a Unicode conversion is in the scope of the project, run the Unicode Conversion Check (transaction UCHECK) on the upgraded system. The tool compiles a list of objects to be adjusted and those from the customer namespace that are not compliant to the Unicode syntax requirements. Even if a Unicode conversion is not planned together with the upgrade, SAP recommends making the entire customer coding Unicode compliant during the upgrade adjustment phase. Unicode compliant coding can also run on non-Unicode environments. This helps reduce test efforts for a Unicode conversion in the future.

### Reduce Development Scope to Used Objects

While the technical change impact analysis of custom developments does lead to a better understanding of the overall adjustment needs, it is often not sufficient to significantly reduce the effort. The lists can contain thousands of objects to be inspected in detail, still requiring a lot of unnecessary effort. You should combine this technical approach with a business-related view of the importance of the identified changes.

To keep this task manageable, you need good and complete documentation of the implemented processes and custom developments to use as a reference. This documentation should be readily available in SAP Solution Manager.

Based on this documentation, you can identify those business areas that require most attention and perform a more detailed impact analysis in those areas.

You need to know which of your developments are actually used, and how often. This statistical analysis should be a regular task of your operations team so that you have a reliable usage history whenever you conduct your projects. One important information source is the Workload Monitor (transaction ST03N), which generates statistical workload and performance data. You can store the history in the source system for periods longer than the normal retention period. You can extract the statistics from your managed systems into the BI InfoCube in SAP Solution Manager.

Run Usage and Procedure Logging (UPL) to track the usage of custom code objects on deeper modularization unit levels. UPL is a new function based on the core functionality of SAP Coverage Analyzer and is available in any ABAP-based system. It logs all called and executed ABAP units, such as programs and function modules, down to classes, methods, and subroutines. You can also evaluate the usage of SAP Smart Forms. This new, enhanced SAP NetWeaver capability does not affect system performance and will catch usage information of ABAP routines when they run. UPL provides complete coverage of object usage, without estimations or evaluation of ABAP call stacks. This includes the detection of dynamically called ABAP elements. UPL is the only technology to close the existing gaps in the SAP workload statistics. With the secured access to the UPL data, your usage information is protected against 3rd party access. The full reporting capabilities, with enriched information in the BW of SAP Solution Manager, give you the flexibility to analyze ABAP usage on demand.

You can use this information to focus your development scope on the objects that are actually used by your company, which greatly reduces the development efforts for upgrade and update adjustments.

The Scope and Effort Analyzer (SEA) is a new SAP Solution Manager function available with SAP Solution Manager 7.1 SP11. This tool makes it possible to perform an early upgrade impact analysis without a physical EHP/SP installation. The SEA generates reliable effort estimation for required adjustment and test activities and helps to reduce the test scope significantly.

The SEA performs a comprehensive analysis with minimal customer input. All steps are performed in the background, after you have entered the necessary data. As a prerequisite, UPL needs to be activated in the system. We recommend collecting production system usage information data with UPL for at least three months. A proper amount of usage data will have a positive influence.
on the result quality and reliability of the SEA. In order to improve the result quality, ensure that all main business activities and periodic tasks are logged. UPL configuration and activation for your managed systems is part of the Custom Code Management scenario in SAP Solution Manager Configuration (transaction SOLMAN_SETUP). To set up UPL centrally in SAP Solution Manager for the managed systems follow the steps described on the SAP Service Marketplace at http://service.sap.com/~sapidb/011000358700000427102012E.

For more information on the Scope and Effort Analyzer and the prerequisites required in your SAP Solution Manager system and managed systems, see http://service.sap.com/SEA.

SAP strongly recommends performing a sandbox test upgrade as early as possible in the project. This ensures that the project team can collect valuable experience on upgrade execution very early. It enables key users to get an early impression of the user experience and capabilities of the new release, which can help planning training measures. If you use a copy of the production system for this test upgrade, you can also get an initial estimate of the downtime for the actual upgrade. All this information helps to better plan and prepare the project and, therefore, should be the first step of any well managed upgrade project.

Besides modifications and custom developments, which were discussed above, the following areas have to be considered for an upgrade impact analysis:

**Customizing**

Customizing is a major task if new functions or enhancements are activated in the course of an upgrade project (Delta Customizing). However, even in the case of pure technical upgrade, there can be changes to SAP standard processes and functions that require adjustments or extensions of the customizing settings (Upgrade Customizing).

SAP release notes and the SAP Implementation Guides (IMGs) are the basic source for information about delta and upgrade customizing. SAP Solution Manager also offers features that allow you to compile a list of customizing settings that should be considered for adjustments.

**Application Changes**

SAP always attempts to create downward compatible programs. This ensures that you can still execute the old functions and programs in the new release. However, introducing new business functions, legal requirements or new frontend technologies in the SAP software sometimes makes it necessary to redesign the SAP standard. This can mean completely changing or even replacing existing transactions, reports, function modules, or data models. Therefore, even customers who use only the SAP standard to run most of their business processes may have to deal with application changes and adjust their business processes when they upgrade.

To identify application changes, carefully analyze the SAP upgrade guides and release notes, particularly the release restrictions published for every SAP product version. Additionally, we recommend using the Application Specific Upgrade (ASU) Toolbox to analyze changes in the application areas. The ASU Toolbox enables you to recognize additional, application-specific steps before and after the upgrade, in addition to the actual technical upgrade. The ASU Toolbox covers the following main areas of application changes:

- Checks whether certain customer-specific data still exists and is useable, such as report variants, display variants, exits, and so on.
- Provides information on necessary customizing settings for new functionalities
- Starts reports that adjust your data to the new release that are not started automatically during the upgrade

You can integrate the ASU Toolbox into standard upgrade procedures using automatically generated task lists that generate all required tasks in the optimal sequence. It is also integrated into the SAP Upgrade Roadmap, which you can access from SAP Solution Manager.
Security Management

Another important topic in the context of application changes are changes to the authorization or security concept. Usually, a new SAP software release means a lot of changes, extensions, new authorizations, or even completely new authorization concepts. Therefore, authorizations must be defined or tailored for new and changed business processes and authorization objects in the target landscape. In systems based on SAP Web AS ABAP, you should use the Profile Generator (transaction SU25) to perform a step-by-step comparison and correction of changed authorizations. This transaction also enables you to view changed transaction codes that have been assigned to roles.

An upgrade also requires a revision of the general security policy. In particular, take into account new functions, technologies, and protocols (such as access using HTTP), as well as new operating system and database software.

The effort and criticality of these activities are frequently underestimated, especially when special legal requirements apply, for example, compliance with certain standards (FDA, GxP, SOX). Revising and testing the authorization and general security concepts are crucial for a final sign-off of the upgrade. Skipping these tasks can easily lead to delays and extra costs in the project.

Interfaces

Today, business solutions usually consist of many systems and applications communicating through a variety of interfaces. Ensuring the stability, consistency, and performance of these interface communications, while one of the connected software components is upgraded, is a key task for the project.

The challenges differ with the type of interfaces you use. Connections between SAP software components are normally the least critical group of interfaces because SAP takes care of the compatibility. However, if interface technologies are changed or new interfaces are added, the interface management concepts and handling procedures need to be inspected and adopted to keep performance and stability of the business processes.

For connections between SAP systems and partner software, you have to make sure that this software is released and certified for use with the new SAP release. Sometimes, this requires an update of the partner software as well. Contact your software vendor to get the information on the interoperability with the SAP software.

If you use custom interfaces, the same considerations apply as for custom developments in general. Good candidates for a more accurate inspection are any type of file uploads to the upgraded system using methods such as BATCH INPUT or CALL TRANSACTION because changes to the transaction (format, required inputs, and so on) usually lead to errors in the file upload process. Therefore, it is normally better to use BAPIs to implement such interfaces. This interface technology is more stable and changes are better documented. In future, BAPIs will be replaced by enterprise services technology (ESR). The most important BAPIs are already available as enterprise services (as of SAP NetWeaver 7.0). Therefore, you should consider using enterprise services when changing your old interface programs.

Summary

Application adaptation is always necessary during an upgrade. In fact, adjusting business processes is one of the major cost drivers of any upgrade, particularly in highly customized systems.

Enable Usage and Procedure Logging in your systems. This tool collects valuable information on used objects in the system, which can help reduce the application adjustment effort.

You can get an early estimate for the adjustment costs by using the Scope and Effort Analyzer, even before physically updating a system. You can minimize the adjustment costs and efforts by focusing on critical business scenarios and processes that are actually used. Unused repositories, data, and customizing objects do not need to be adjusted and should be considered candidates for housekeeping.

Perform a test upgrade on a copy of the production system (sandbox upgrade) as early as possible. This enables you to analyze your system and set the right focus for your adjustment work.
Analyzing and correcting custom developments is the biggest block of adjustment work. Use sophisticated tools like SAP Custom Development Management Cockpit (CDMC) to analyze these developments in the sandbox system.

Use the Application-Specific Upgrade Toolbox (ASU Toolbox) for a systematic, tool-based approach for executing application specific adjustments.

2.3.4 Test Management

This focus area helps you to test your system efficiently.

Due to the amount of program code and the complexity of the implemented business logics, you cannot assume that a technical analysis of the software alone guarantees maximum business continuity. For example, changes in the SAP standard business logic may not be visible in a technical analysis but can lead to serious issues in custom developments. Proper testing is the only way to detect such changes. This is why testing the business applications is usually the key cost driver of upgrade projects and the expected test effort should be known as early as possible. You can use the Scope and Effort Analyzer (SEA) for this evaluation task. The SEA is described in detail later in this document.

The complete test phase from planning to test execution is supported within SAP Solution Manager. For a complete overview, see the SAP Standard for Test Management.

2.3.5 Technical Change Management

This focus area helps you to manage parallel changes.

For a system in the continuous improvement phase, most organizations have a comprehensive process in place for managing changes. This process typically involves the following stages:

1. Requirement evaluation
2. Decision-making workflow
3. Implementation
4. Testing
5. Deployment

Such a process enables you to assess whether the final value of a change is worth the effort required to implement it. In addition, you can minimize the potential impact of the implementation on the rest of your solution.

It is important not to compromise any aspect of this process in the course of an upgrade. A project could be tempted to allow mass exceptions from the set of rules because there are usually many changes that have to be implemented in a short period of time and because there is often not enough time for a thorough evaluation if a change is really required.

The following aspects are the key success factors not only from change management perspective but also for the whole project.

Start Early

As mentioned before, using the Scope and Effort Analyzer and an early sandbox upgrade provides all the information needed to plan ahead for a full-blown technical change management process. This time should be
used to determine the real need for change and the best option for its implementation. If time is lacking at this stage, the project will end up adjusting any issues that arise to make everything work just as it did before. With several thousand modifications and custom objects, this will cause an enormous amount of effort.

**Link Change to Test and Training**

Ideally, you do not want to test everything and retrain everybody every time you perform an upgrade, especially for technical upgrades with a small gap between the respective releases. However, even if you do not plan to implement any major changes, some changes cannot be avoided. It is then extremely important to reflect every single aspect of these changes in test planning and delta training.

In the evaluation phase or on program definition level, it may become apparent that the existing change management procedures require changes themselves. The change management process is one of the most comprehensive solution-support procedures, with many people involved in several roles and a sophisticated workflow implemented using a number of tools. The reliability of this process is the first priority. Do not reassign roles, change the sequence, or implement new tools unless urgently required. If absolutely necessary, for example, in order to facilitate the upgrade at all, try to adapt the process as early as possible and before the upgrade so that the new scenario is established in day-to-day business by the time you start the main upgrade project.

SAP Solution Manager provides powerful tools to implement change request management, change impact analysis, test management, the testing itself, and e-learning. Details about standards in technical change and test management are outlined in the SAP Standard for Change Control Management and the SAP Standard for Test Management respectively. The white paper *Two Value Releases per Year* also describes the best practices for change management.

**Manage the Code Freeze**

After upgrading the development system, you can begin implementing the changes to repository objects of the upgraded system. However, you might still need to make changes in the current release, such as emergency corrections and other maintenance tasks, while the upgrade process is ongoing. Therefore, SAP strongly recommends using a dual landscape.

A dual landscape involves creating a parallel copy of the development system (known as a retrofit system) and using this to make all developments for the upgrade. You can then continue to use the original development system to perform any maintenance or corrections for the current release while the upgrade project is ongoing. However, at this point, there are two different versions of each object in your system object (one in each development system) but still only one production system. Therefore, you need to manage your changes carefully. For example, if you make a correction in the current release and transport it to the production system, the correction will be overwritten by the second version of the changed objects when the upgrade is eventually implemented in the production system. The only solution to this problem is dual maintenance; that is, making the changes twice.

Most of these changes have to be done manually because customizing transports between different releases are not supported and repository transports contain critical objects. This process is both costly and error prone.

To reduce the amount of manual effort required, SAP Solution Manager supports the retrofit process, both as part of Change Request Management and as a stand-alone function. Retrofitting changes in SAP Solution Manager significantly improves the process of synchronizing dual transport landscapes. The key advantages are as follows:

- Automatically detect conflicts between objects
- Automatically retrofit most changes made in the production landscape
- Generate a complete, object-level work list of all transports that need to be synchronized
- Log all changes
The following figure shows a typical dual landscape based on a three-tier production landscape. Any emergency corrections or short-term maintenance for the current release are developed in the original development system before being transported as normal through the quality assurance system and into the production system. All development for the next release takes place in a parallel landscape. For example, you make the major changes required to implement the upgrade in the copy of the development system. These changes are then retrofitted to the current landscape and eventually transported to the production landscape when the new release is implemented.

![Dual Landscape Concept with Retrofitting](image)

**Figure 5: Dual Landscape Concept with Retrofitting**

To minimize the effort involved, SAP recommends introducing a development freeze during this period. This means that no requests for change are implemented in the production system, except for urgent corrections resolving severe issues with a large business impact. In particular, this also means that there should not be any business roll-outs after the development system is upgraded. Even if the respective customizing and coding is already in the development system before it is upgraded, a go-live of the functions in the double maintenance period will usually affect the upgrade project because the same developers are needed for corrections in the stabilization phase of the roll-out as well as for the double maintenance.

Major changes to connected systems should be completely avoided, at the very latest after the start of the integration tests. Otherwise, the results of the tests will be unreliable.

It is often difficult to negotiate and agree on a development freeze with the business departments. They are normally afraid of losing flexibility during the period. Therefore, we recommend planning and communicating the freeze early to minimize the impact of the business. Unfortunately, many projects simply cannot afford to have weeks of development freeze due to business requirements. To minimize the length of the freeze, it is a general best practice to start with the upgrade adjustment in the sandbox system early, before the double maintenance period is in place. This saves time after the development system upgrade because you can simply reapply the prepared changes.

If a development freeze is not at all acceptable for the business, a solution might be to introduce the release management described in the Two Value Releases per Year concept. This means that business changes and SAP upgrades can be bundled into one release, which helps retaining the flexibility. For more information, see the SAP Support Portal at [http://support.sap.com/solutionmanager](http://support.sap.com/solutionmanager) ➔ Knowledge Transfer ➔ White Papers.
You should include representatives of the upgrade project in the change request process for the previous release during the double maintenance period. Experience shows that this is the best way to ensure that the tradeoff between the expected benefits from the requested change and the extra risk and effort for the upgrade are properly taken into account.

Summary

Use your established change management procedures also during your upgrade project. However, include the upgrade project team in the decision process for requests for change after start of double maintenance period. Document any change during the upgrade project. This provides the basis for efficient testing and training. Create a maintenance landscape for your current production environment in parallel to the upgrade project landscape. This will ensure maximum availability for your business. Implement a strict development freeze period, at the very latest after upgrading the development system. This will minimize project efforts for double maintenance and provides a stable landscape for integration testing. If possible, start adjustment work and unit testing already in the sandbox upgrade system. This helps reduce the overall duration of the development freeze period.

2.3.6 Business Downtime

This focus area helps you to minimize system outage. Business downtime is one of the classic challenges of any upgrade and maintenance project. In the era of 24/7 business, just-in-time production, and continuous processes, businesses cannot simply shut down. For example, an oil field cannot be stopped to accommodate a software update. SAP is constantly finding ways to shorten the technical upgrade time. One such method is to use the switch framework. This involves creating a shadow instance of your SAP landscape in which you execute a large part of the upgrade process while the normal system is still running. This significantly reduced the technical downtime. In this section, we describe SAP best practices for managing and minimizing business downtime, and the latest SAP technologies to reduce planned outages to nearly zero.

Typical Pattern of Planned Downtime

Business downtime is a time period during which the production system is not available to the business (end users, interfaces, background processing). It is further divided into types, for example, the ramp-down phase or technical downtime. “Uptime” refers to the time during which the production system is running and available for the business. Business downtime usually starts with the ramp-down phase, which ensures the consistency of the systems and database, and the controlled ramp down of productive use. For example, this involves the following tasks:

- Locking all business users
- Rescheduling background jobs
- Processing update tasks
- Cleaning up data queues
- Deactivating interface connections

A consistent backup of the database and file system ensures a proper reset point in emergencies. After the ramp-down phase, the technical downtime starts. During technical downtime, the maintenance tool runs on the system and the system is updated. During this process, the system can be up and running (but with
controlled access only), or shut down to optimize the deployment process and ensure data consistency. Technical downtime can usually be optimized by adjusting tool-specific parameters and settings, or by increasing the available hardware and disc input/output time, depending on the update event tasks.

The post-processing phase follows technical downtime. During post-processing, technical system checks are performed, to ensure the technical correctness of the systems. This can be followed by customer-specific software update tasks, such as importing customer transports, updating software add-ons, supplementing or generating objects and programs, and so on. These activities are customer-specific and can change with the scope of different update events.

The next phase is validation. During this phase, selected business users (functional core team) test the functions of the production system. These tests concentrate on selected business-critical processes and form the basis for deciding whether the meets the business requirements. If the team approves the new system version, the ramp-up process can start. The ramp-up process involves tasks such as releasing end users for business, establishing interface connections, scheduling background processes, and so on. After the ramp-up process is complete, the regular uptime, or production use of the system, can start again. However, if the team rejects the upgraded version of system, you need to restore the backup version or reset the system. This is unlikely to happen if you planned and performed the upgrade properly.

Holistic optimization of downtime focuses on all elements of the business downtime, not just on the technical part. To review the activities, and their duration, activity by activity, is an intensive task. Instead, you should initially focus on the most time-consuming activities.

Managing Planned Downtime

Business users expect continuous availability of functions provided by SAP systems. Each outage, even if announced in advance (planned downtime), can disrupt business.

To reduce planned business downtime, consider the following aspects:

- What is the acceptable downtime for the business?
  This business downtime requirement can influence the tool or technique you use to perform maintenance, and the cost of optimization procedures and techniques.

- Which update event is planned?
  Installation of support packages, enhancement packages, database maintenance, or other activities. Consider the technical dependencies of the selected update event, for example, database or operating system patch version prerequisites.

- Which systems participate in the update event, for example, based on technical or functional dependencies?
  Define the constellation of systems to run maintenance, based on your needs and dependencies.

Reducing planned business downtime usually starts with a proof-of-concept project to evaluate activities, timings, and downtime, and specify the next steps.

Agree on Possible Maintenance Windows

There should be a general agreement between the parties involved about the availability of business functions. Depending on the criticality of the supported business, the frequency and the duration of maintenance windows should be aligned between the IT department and the business users.

The regular maintenance windows should be used for regular system maintenance activities requiring outage. It is often acceptable to business users to have a short outage (less than four hours) monthly, during a period of low system activity—usually at weekends or during the night. The frequent short outage allows you to perform maintenance activities and introduce minor system changes, improvements, or innovations.

A four-hour window is usually too short for major system changes, including updating the SAP software stack with a new SAP release, new enhancement package, or new support pack stack. A separate outage needs to be agreed
with the business for this. Typically, this longer outage (24-48 hours, or even longer), is planned for long weekends.

**Encapsulate the System Within the System Landscape**

Business functions provided by the IT systems have various critical points across components in the system landscape.

In some cases, only selected components of the system landscape require an outage, for example, only a BW system. To minimize the impact on the supported business, consider whether the affected components can be shut down individually, and the remaining components that support the critical business, such as the ECC or CRM system, can stay in operation.

This selective treatment requires documented knowledge of business processes and their criticality across the system landscape.

When shutting down system components selectively, ensure that the interfaces connecting the disabled components are deactivated; monitoring these interfaces might cause errors. This should be communicated in advance to the persons responsible for monitoring.

### 2.3.6.1 Downtime and Runtime – Cut-Over Planning

Especially in complex customer releases, the duration of the downtime can exceed the standard maintenance window. The time during which implementing a new release affects your business processes goes beyond just the business downtime.

48 to 72 hours before the downtime begins, restrictions are placed on the system. During this phase, no repository changes can be made in the production system. This phase also impacts the development system. No development activities can be performed during the upgrade of the development system.

![Figure 6: Impact of Implementing a New Release](image)

In the preparation for the planned downtime, the uptime part of the cutover should be kept as short as possible to minimize the duration of the restricted phase on the production system, and minimize risk. Therefore, it is important to prepare a detailed cutover plan.

A cutover plan is a to-do list for all steps of the planned activity in the system. You need to include a sufficient level of detail in the cutover plan to ensure that the tasks are performed correctly.

Part of the cutover plan involves estimating the duration of the business downtime. You make an initial rough estimate based on business requirements and general information such as database size, experience from other upgrade, and so on. After performing mock runs and other tests, you can adjust the downtime estimate based on
the results to form your final estimate. This final downtime has to be approved by the business as a planned downtime.

The cutover plan should generally contain the following information:

- Activities, steps, and timing of the planned upgrade
- List of key personnel responsible for each item, including availability, contact details, and so on
- Detailed timeline of all steps, automatic and manual
- List of checkpoints and quality gates
- Detailed security backup plan
- Detailed contingency plan, including time buffer for unexpected situations and recovery/restoration plan in the event of errors
- Process for emergency corrections or transports to the production system
- Clearly defined point of no return

SAP strongly recommends that you perform two test runs of the cutover plan in order to simulate the results before the actual production cutover. It is important that you perform any test runs in a representative environment, that is, using similar data volumes, hardware, and usage. Otherwise, your test results will not be an accurate reflection of your final cutover.

### 2.3.6.2 Minimize Planned Downtime

Typically, the activities related to update events require system downtimes for a number of hours, or even days. Depending on the criticality of the functions running on the affected system, the downtime is usually planned for weekends, or even long weekends, in order to minimize the impact on the business. With the increasing demand for 24x7 services, this may no longer meet business requirements.

The Software Update Manager (SUM) is the standard tool for implementing support or enhancement Packages and for release upgrades for ABAP-based, Java-based, or dual-stack systems.

SAP has invested in tools and methodologies to help you upgrade your systems with almost no business downtime at all. In some cases, the SAP systems may even stay completely online for the planned maintenance activities.
The following section describes the methods, benefits, and efforts available from SAP for ABAP-based, Java-based, and dual-stack systems.

Tools and Services for ABAP-Based Systems

Support Packages and Enhancement Packages

SAP support packages for ABAP-based SAP NetWeaver systems are deployed using the Support Package Manager tool (transaction SPAM). Support Package Manager ensures that support packages are imported only in the specified sequence. You can make various settings to adjust the tool to suit your needs, such as the number of parallel import processes.

If you plan to import a larger number of support package stacks, or if you want to reduce the downtime during your update event, use Software Update Manager (SUM) tool instead of the Support Package Manager. SUM can reduce the technical downtime by 50-70% compared to Support Package Manager.

Functional enhancements are shipped as SAP enhancement packages for SAP NetWeaver-based systems or SAP Business Suite products. The enhancement package strategy simplifies the way customers manage and deploy new software. Customers can selectively implement software innovations from SAP and activate them depending on business requirements. As a result, customers can isolate the impact of software updates, and bring new functionality online faster, through shortened testing cycles.

Using SUM, you set up a parallel shadow system, in which major deployment steps of the functional enhancement are made in parallel to the production operation. Technical downtime is necessary to perform the system and kernel switch and further downtime-relevant changes, to ensure the consistency of the database.

The Software Update Manager offers several downtime optimization features, some of which are active by default, and others can be activated on demand. These features reduce business downtime compared to current standard tools, because more of the downtime-relevant deployment phases are executed while system is still available for business users. Examples of these features are as follows:
- Schedule load generation on the shadow instance
- Schedule long-running after import methods on the shadow instance
- Mass generate enhancement objects and enqueue objects during uptime
- Near-zero downtime maintenance using the record an replay technique
- Deploy customer transports during uptime.

For information about the prerequisites for using SUM, see SAP Note 1759080.

**Release Upgrades**

A new software release comes with improvements and new functionality, and uses state-of-the-art technologies to meet users’ growing business requirements and increase their productivity. The transition to a new release presents data and system consistency and business downtime challenges. SAP offers a comprehensive set of technical tools and services to facilitate and safeguard upgrade projects.

SAP’s upgrade technology is continuously improved, with the focus on unattended operation, usability, and technical downtime reduction. The single point of access for all upgrade-related questions is the SAP Service Marketplace [https://support.sap.com/release-upgrade-maintenance/upgrade-info.html](https://support.sap.com/release-upgrade-maintenance/upgrade-info.html).

**OS/DB migrations, Unicode Conversions**

The Near Zero Downtime method (NZDT) was developed to reduce the planned downtime caused by SAP software updates or platform changes. This method is especially appropriate in large change events, such as Unicode conversions, migrations to SAP HANA or other OS/DB migrations. If the downtimes associated with the standards tools are not sufficient, you can use NZDT for classic upgrade projects and installing enhancement and support packages. Finally, you can use the NZDT method to bundle several maintenance activities and fit them into one maintenance window. During this window, production availability is ensured and core business processes are always available. However, customizing settings can be significantly limited for up to 14 days.


**Upgrade and Migration to SAP HANA**

If you need to upgrade the source system before migrating to SAP HANA, these activities can be combined by using the database migration option of the Software Update Manager. This combined approach reduces the total runtime significantly.

For more information on the database migration option, see the SAP Community Network at [http://scn.sap.com/docs/DOC-49580](http://scn.sap.com/docs/DOC-49580).

**Zero Downtime**

Responding to the continuously growing demands of system availability, SAP is developing a Zero Downtime Maintenance Option (ZDO).

The ZDO method can be used for any major change event and enables you to perform a non-disruptive update of the SAP software stack. This includes enhancement and support package stack implementations, as well as implementation of custom code or configuration changes.

In the standard upgrade procedure, the production system needs to be taken out of operation to finalize the upgrade downtime activities. The ZDO method uses a temporary bridge system to continue operations during the downtime of the original system. The bridge system is connected to the data of the original system, so users can continue to use the data in the original system, so no clone is necessary.
This method uses the same technology as the current SUM tool. During the implementation of the new software, the core functions are running and are available to the users. Some features will, however, only be available in read-only mode.

ZDO is available for most critical ABAP-based components, including SAP ERP Central Component (ECC).

**Available Tools for the Java-Based Systems**

The Near-Zero Downtime Maintenance (nZDM) tool for SAP Process Orchestration is based on the nZDM tool for SAP Enterprise Portal which is included in the SL Toolset. As SAP Solution Manager 7.1 SP10, the tool has been extended to an nZDM tool for SAP NetWeaver Java (nZDM Java). It is the main downtime minimization tool for the Java stack in general and it covers all maintenance activities.

For more information on downtime reduction and availability see SAP Community Network at [http://scn.sap.com/community/it-management/alm/software-logistics/blog/2013/10/16/how-to-continue-with-downtime-minimization-for-java-applications](http://scn.sap.com/community/it-management/alm/software-logistics/blog/2013/10/16/how-to-continue-with-downtime-minimization-for-java-applications)

**Available Tools for Dual-Stack Process Integration (PI) Systems**

**Support Packages**

For SAP Process Integration (PI) systems up to release 7.31, nZDM for PI is a special tool that you can use to apply support packages and patches. This tool is available as of SL Toolset SP7 and can reduce the downtime for support package implementation to 30-60 minutes.

**Release Upgrade**

For upgrades to 7.31 / 7.40 of PI systems, the nZDM feature of the Software Update Manager is available on request only. It can reduce downtime to 10-30 minutes.
For more information on downtime reduction for PI systems see SAP Community Network at http://scn.sap.com/community/it-management/aim/software-logistics/blog/2013/10/16/how-to-continue-with-downtime-minimization-for-java-applications

**Recommendations for Other Dual-Stack-Based Systems**

As of SAP NetWeaver 7.0, including enhancement package 3 and SAP Business Suite 7i2011, which is based on SAP NetWeaver 7.0, including enhancement package 3, SAP dual-stack systems are no longer supported.

As of SAP Business Suite 7i2011, it will no longer be possible to upgrade an SAP dual-stack system to a higher release. You need to split any dual stack systems before the enhancement package deployment release upgrade.

Downtime minimization of single stack systems is less complex and cost-intensive as some features of standard maintenance tools can be used. For more information about how to split dual-stack systems, see SAP Note 1655335.

**Summary**

The business downtime is defined as the overall time a system is not available for the business during change or update events. The technical upgrade downtime is only one part of this business downtime.

Perform a sandbox test upgrade with a copy of the production systems during the preparation or blueprint phase of the upgrade project, including all critical tasks like Unicode conversion or other data conversions. This is the first opportunity to get reliable estimates for the business downtime.

Agree and decide as early as possible with business what the downtime requirements are. Only then it is possible to decide what measures need to be taken for downtime optimization and to have enough time to perform the optimization.

If you perform a database migration or Unicode conversion together with the upgrade, plan for two to three test cycles for downtime optimization on separate hardware in parallel to the upgrade project. For at least the final tests, use hardware that is comparable to the planned production environment.

The downtime of a technical upgrade does not depend on the overall database size, but only on the size of those tables that are subject to a data conversion (DDIC structure or content). However, general archiving or data cleansing activities will be beneficial for data-dependent tasks like backups, Unicode conversions, or database migrations.

Depending on the system type (ABAP or Java) and the maintenance activities planned (support packages, Unicode conversion, migration SAP HANA, data center migration, and so on) different tools or services exist, with different advantages and disadvantages. These need to be carefully evaluated to identify the best approach for your situation.

### 2.3.7 Preparation for Go-Live

This focus area helps you to avoid surprises when going live.

The ultimate goal of any project is that on after the upgrade, the users can run their business in the same way as before the upgrade. Ideally, they should not experience any differences or difficulties due to the upgrade.

Unfortunately, this goal is not always completely realistic. However, you can take steps to make sure that your upgrades cause the minimum possible disruption to end users. At the very least, you should complete the following tasks to minimize the risk of disruptions and ensure maximum business continuity:

- Understand the adjustment needs
- Define the right test scope
- Determine training requirements
- Set up a proper cutover plan
- Understand the change impact to solution operation

Adjustments
You have to understand the change and adjustment requirements introduced by the upgrade and their impact to business. This topic is discussed in more detail later in this document.

Testing
Due to the amount of program code and the complexity of the implemented business logics, do not assume that a technical analysis of the software alone will ensure maximum business continuity. Particularly, changes in the SAP standard business logic may not be visible in a technical analysis in the first place. Nevertheless, they could lead to serious issues, for example, in custom developments. Proper testing is the only way to detect such changes. This is why testing the business applications is usually the key cost driver of upgrade projects.

User Training
In addition to testing the processes, end user training is an important measure to ensure business continuity. Specifically, changed screens, menu paths or new or changed business functions can lead to a wrong handling or leave the end user unable to proceed. As a consequence, business performance decreases. This risk is the larger the greater the leap between source and target release or if the application has been redesigned on large scale. However, as with testing, you need to carefully determine the extent of the training measures in order to limit the necessary investments in this area. First, you have to determine the training requirements by identifying those areas with the largest changes to user experience. You should define key users for each of these areas. These users should receive a full delta training curriculum and may also take part in the integration or acceptance tests. Together with these key users, the project team has to define the training curriculum for the end users in each critical business area.

Efficient end user training concepts should not only rely on SAP courses or in-house classroom training. Instead, consider additional training techniques like e-learning sessions or short remote info sessions on special topics. Well-documented business changes is a key to organizing these trainings well. Finally, a special help desk should be set up during the initial stabilization phase after the upgrade to provide quick answers to user questions and problems. Using Internet communication techniques, like FAQ pages, discussion forums, or Wiki pages through your intranet will help you share information efficiently with end users.

Solution Operations
An SAP upgrade may affect all areas of solution operations. However, effective solution operations are key for a smooth transition to the new release. Therefore, the current procedures for daily, weekly, monthly, and annual tasks must be kept up to date; and the operation staff has to be trained accordingly. This may include the following tasks:

- If you are implementing new business processes, application components, or technologies in the course of the upgrade project (functional upgrade), the new parts of the solution need to be integrated into the existing operations concept.
- If you are migrating hardware, operating system, or database software to a different platform, the system administration and monitoring procedures must be analyzed and updated.
- If you are changing interface technologies or adding new interfaces, the interface management concepts and handling procedures need to be inspected and adopted to ensure performance and stability of the business processes.

Usually, updating the general security concept is also necessary.
End-user support during the initial stabilization phase is the first challenge the solution operations team has to handle after the upgrade. Of course, the upgrade project team should provide support in this phase until a formal handover to the regular operations team. This phase normally lasts 2-3 weeks. However, we strongly recommend already including the responsible operations team during the testing and initial stabilization phases to enable maximum knowledge transfer. This allows the operations team to gain experience on the new or updated operation procedures as early as possible.

Summary

Testing, training and proper planning of the cutover activities are key measures to ensure maximum business continuity.

To limit the efforts for these activities, focus on the things that really matter. This involves prioritizing your business processes according to their business criticality and understanding the upgrade impact to the critical business processes as much as possible.

Use the ASU toolbox to analyze and prepare your applications for the upgrade.

Do not forget to test and train any changed operation procedures. Wrong system administration or application support can be as dangerous as user handling errors or software bugs.

2.3.8 Protection of Investment

This focus area helps you to ensure the upgradability of your solution.

An SAP upgrade to a higher release is a change event that usually happens only once every few years. An upgrade means a massive investment in the IT solution that needs to be justified by a solid business case. When looking at the key drivers for costs and risks described in the previous chapters, you will find that many of them are related to daily operations topics.

This means the way how you organize and run your solution operations has a strong impact on the upgradability of that solution. Therefore, reviewing and reconsidering these established practices before the launch of an upgrade project will help reduce or limit the investment needed for the upgrade and will lead to a faster return on any investments that you do make.

In this context, the term “upgradability” describes the ability to upgrade as SAP solution in accordance with SAP best practices and standards. Following these standards is the key to minimizing risks and costs. Any deviation from these best practices and standards regarding the setup of software, hardware, business processes, and operations, usually leads to extra investments when upgrading the solution in the future. For example, modifications and custom code are the main cost drivers of upgrades and, for that reason, need to be kept to a minimum. Therefore, it is crucial to implement a change management process that prevents the introduction of unnecessary modifications or custom developments. This includes regular checks of intended and existing custom development against standard functionality as well as regular monitoring of the usage of existing custom developments. The SAP Standards for Change Management and Custom Code Management help you to set up the appropriate processes in your company. An upgrade project is a good opportunity to analyze the existing modifications in your SAP system and analyze whether they are still needed or can be replaced by SAP standard functionality during the course of the upgrade project.

Another example is test management. Improvements in test environment, test management or test tools could have a large impact on the bottom line of an upgrade projects’ costs. SAP Solution Manager, for example, offers a comprehensive test workbench with integration to the eCATT test tool and also offers integration with the HP test environment. For more information on the recommended test management procedures and tools, see the SAP Standard for Test Management.
Make changes to your solution operations procedures as soon as possible before starting an upgrade project. This allows the new procedures or tools to become established in your everyday business before a major project like an upgrade takes place. In turn, this provides the benefits of improving your solution operations while minimizing any potential risks.

Generally, SAP recommends reviewing the upgradeability of the current solution on a regular basis depending on the number and frequency of changes to your solution operations. Ideally, a first solution operations review should be a part of the initial implementation. We recommend following the RunSAP methodology for implementing SAP solution operations standards for this purpose. This ensures the best upgradeability of the SAP solution and helps to reduce your TCO.

### 2.4 Upgrade Support Tools

To help you plan and execute your upgrade projects. By using SAP Solution Manager, you have access to the latest upgrade methodologies and related tools. SAP provides the best practices for the functional and technical aspects of upgrading an entire SAP landscape; documentation about configuration support, testing, and e-learning management, and a service desk.


This guide provides a detailed description of how to employ the tools provided by SAP.

---

**Figure 9**: Overview of Tools and Activities when Upgrading SAP Business Suite Products

This section describes the most important SAP Solution Manager tools and features related to upgrades.
2.4.1 Scope and Effort Analyzer

Scope and Effort Analyzer is an SAP Solution Manager function available as of SAP Solution Manager 7.1 SP11. It allows you to analyze the impact of SAP software changes triggered by support packages or SAP enhancement package deployments before you physically install them. A structured report guides you through all findings to plan, optimize, and set up the project.

Features

- Transparency – understand which modifications and custom code objects need adjustment
- Prediction – understand the impact to business processes and what to test
- Optimization – focus your test and adjustment efforts on used functionality only
- Flexibility – run the analysis at any time in the planning phase of your update project

Benefits

- No physical deployment of software changes to calculate the impact
- Fast and flexible implementation of software changes
- Visibility into the impact on business processes and custom code
- Significant reduction of costs for regression tests and development adjustments

For more information, see SAP Service Marketplace at http://service.sap.com/sea

2.4.2 Work Centers

SAP Solution Manager work centers bundle role-based content with task-specific authorizations and a state-of-the-art, Web-based user interface. Work centers deliver all the functionality, components, and tools needed to manage an entire landscape throughout the IT lifecycle. The Implementation/Upgrade work center provides the central point of access for SAP Solution Manager tools and processes related to project management and upgrades grouped by the major Upgrade Roadmap project phases.

2.4.3 Project Management Tools

To enable you to deal with impending bottlenecks quickly and efficiently, it is essential that you have professional, transparent project management procedures and a clearly defined project scope. With these in place, you will be able to safeguard your company’s investments, deploy resources for specific objectives, set priorities, and ensure that everyone involved in a project communicates well.

SAP Solution Manager’s role during this phase is to provide you with roadmaps – proven procedures for project management – and to structure project management activities clearly using tools such as project administration functions. The basis for upgrade projects is the SAP Upgrade Roadmap. Additionally, the RunSAP Roadmap should be used for topics related to solution operations.
2.4.4 Solution Documentation Tools

To create a baseline for your upgrade project, SAP Solution Manager helps you document your current IT landscape, as well as implement your business processes.

The SAP Solution Manager System Landscape repository, accessed using transaction **SMSY** or from the System Landscape Management work center, provides a storage location for your IT infrastructure. The maintained repository is also the basis for system administration and monitoring tools like EarlyWatch Alert (EWA).

The Solution Documentation Assistant in SAP Solution Manager evaluates business processes on production systems. It prepares upgrade projects, evaluates new functionality, and analyzes custom developments. This tool can help you determine which business processes are used in your production systems and display the results graphically.

The SAP Solution Manager also helps you to draw up a well-defined, transparent conceptual design (Business Blueprint) for your project in the business blueprint transaction **SOLAR01**. It provides valuable support, especially for mapping and documenting the processes with regard to implementation.

2.4.5 Innovation Discovery

The SAP Innovation Discovery for SAP Products provides detailed delta information for new capabilities and business functions that have been added between two SAP software releases or any of the available enhancement packages. It also helps identifying their respective business benefits. For more information see [http://www.sapsolutionbrowser.com](http://www.sapsolutionbrowser.com).

2.4.6 Upgrade Dependency Analyzer

The Upgrade Dependency Analyzer (UDA) helps you analyze known dependencies when upgrading technical components in their landscape. You can access the UDA from SAP Service Marketplace at [http://service.sap.com/uda](http://service.sap.com/uda) or from the Implementation / Upgrade and Change Management work centers. This tool provides dependency information to support the planning of upgrade projects. It compares the system that you want to upgrade with another system in the solution landscape. The output is an online dependency statement and an optional reference to an SAP Note. Customers can also provide their feedback through the tool.

Note

The Upgrade Dependency Analyzer is a high-level planning tool for upgrades; it does not cover all dependencies that are relevant for planning an upgrade. Therefore, you should always consult the SAP master and upgrade guides before planning an upgrade. They refer to specific SAP Notes that always reflect the most current status regarding release restrictions.

The dependency statement does not generally indicate whether two components are technically compatible. It refers to upgrade scenarios and is relevant only for functions that were already in use before the upgrade. The check is performed on the technical component level only. It does not provide information at the business process level. Information on business processes is available in the Scenario and Process Component List.
2.4.7 Custom Code Management

The main objective of custom code management is to provide guidance and guidelines on how to create good custom code with maximum business benefit and minimal technical effort. SAP Solution Manager provides the following features to support effective custom code management.

<table>
<thead>
<tr>
<th>Feature</th>
<th>Description</th>
</tr>
</thead>
</table>
| Custom Development Management Cockpit (CDMC) | The Custom Development Management Cockpit (CDMC) helps you to centrally leverage the visibility of the SAP changes triggered by an upgrade or implementation of support package stack, as well as their impact on your custom developed solutions. This helps you to optimize upgrades of these developments and reduce costs. The cockpit provides the following functions:  
  - Analyze custom developments in your SAP system and identify obsolete developments (Clearing Analysis)  
  - Identify the potential consequences an upgrade can have on custom developments (Upgrade / Change Impact Analysis)  
  - Analyze objects that are contained in transport requests (Change and Transport System Analysis)  
  - Calculate the necessary effort for adjusting the custom developments affected by an upgrade or support package installation |
| Custom Code Lifecycle Management (CCLM) | The Custom Code Lifecycle Management application (transaction CCLM) is the central repository for all custom code objects and the point of access for all functions that are used to manage the lifecycle of these objects from creation to deletion. You can monitor and track changes to KPIs, such as code inspector errors, test coverage in Q-system, and so on. CCLM helps you to plan upgrade tests more efficiently by identifying custom objects that are currently in use. |
| Custom Code Analysis | You can access additional tools for custom code analysis, such as Clone Finder, from transaction /SDF/CD_CCA or by choosing the link in the Custom Code Management work center. |

2.4.8 Test Management Tools

After you have executed the delta changes and adjustments, SAP Solution Manager helps you plan, execute, and monitor your integrated testing based on your business process documentation and the delta changes that you selected. SAP Solution Manager provides the following tools to help you with test management.

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Test Workbench</td>
<td>Helps you to document and to manage your test cases.</td>
</tr>
<tr>
<td>SAP Business Process Change Analyzer (BPCA)</td>
<td>Analyzes your business processes using application traces. As a result, it provides a list of all technical objects used in this process known as the technical bill of material (T-BOM), including objects that have been called only dynamically at runtime. Based on this result, you can conduct a change</td>
</tr>
</tbody>
</table>
### Tool Description

<table>
<thead>
<tr>
<th>Tool</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Too</td>
<td>analysis for implementing support packages, enhancement packages, or customer transports, which helps you to identify affected business processes and to create specific test cases for those processes.</td>
</tr>
<tr>
<td>SAP Test Acceleration and Optimization (TAO)</td>
<td>SAP TAO is a suite of applications that provides automated test case creation, system change detection, test data management, reporting, and analysis. It includes test cases for the most important business processes and for industry solutions.</td>
</tr>
<tr>
<td>Extended Computer-Aided Test Tool (eCATT)</td>
<td>Allows you to automate the testing process and to create test data for user acceptance tests.</td>
</tr>
</tbody>
</table>

Additionally, SAP Solution Manager provides access and interfaces to external tools, such as SAP Quality Center by HP and the SAP LoadRunner by HP.

### 2.5 Basic Architecture

This section provides recommendations on how to set up your system in order to minimize upgrade risks and reduce the code-freeze period.

We typically distinguish between three landscape environments during the project:

- **Productive landscape**
  The productive landscape is the landscape that you want to upgrade. It typically consists of a development system, a quality assurance (or test) system, and the production system.

- **Temporary maintenance landscape**
  This landscape consists of temporary development and quality assurance systems.

- **Upgrade project systems**
  Usually, this line consists only of the sandbox or upgrade project system, in which the initial upgrade and functional tests are carried out in the blueprint phase. Later, the system could also be used for additional upgrade tests, for example, to optimize the downtime. However, some projects may require additional temporary upgrade test systems, for example, to perform the vocabulary scan for a Unicode conversion.

The figure below is based on a classical three-tier transport landscape for the production environment. All systems in the landscape are connected to the productive instance of SAP Solution Manager.
The systems are typically created and made available according to the following process:

1. Copy the production system to a sandbox system.
2. Upgrade the sandbox system.
3. Copy the current development system to the temporary maintenance development system and link to current QA system.
4. Upgrade the development system in the productive landscape.
5. Copy the production system to the temporary maintenance QA system and link to the production system.
6. Upgrade the QA system in the productive landscape.
7. Upgrade the productive system and link to the upgraded development and QA systems.
8. Remove the temporary maintenance landscape and upgrade project system.

Alternatively, if you have a development system available that contains a good excerpt of data on which you can base the upgrade tests, this system could used as a basis for the sandbox.

In reality, there are many variations on this simple landscape. Many organizations may have far more complicated system landscapes due to their business and geographic requirements. This leads to deviations of the landscape and sequence above depending on your specific situation and requirements.

Generally, the complexity increases for a system landscape that contains either fewer than or more than three systems in a transport landscape. Each system adds an additional upgrade, but having fewer than three systems limits the possibilities for testing and can lead to a less efficient upgrade.

For example, if you already use a five-tier production landscape with a separate development and test system for the development of new projects and another development and QA system for corrections of the production system, you need not to create a separate maintenance landscape during the project. The existing maintenance system can still be used for this purpose, while the project systems can be used for the upgrade project work. However, of course, after completing the upgrade project work, all five systems of the landscape have to be upgraded.

If you have NetWeaver PI, EP, and BW systems, or smaller Business Suite systems, a two-tier production landscape is often used, which consists only of a combined development and quality assurance system and the production system. Here, the QAS systems in the recommended project landscape can be omitted as well. However, you should still set up a sandbox system created from a production copy.
Upgrade projects, which often consist of multiple projects themselves, require a well-managed upgrade rollout program to ensure that all systems in a landscape are correctly and appropriately upgraded. This involves timing and spending decisions, as well as a lot of coordination.

### 2.6 Roles and Responsibilities

The following figure shows how project activities are mapped to the SAP Solution Operations Standards organizational model.

![Figure 11: Upgrade Project Tasks and Responsible Customer Units](image)

For more information on the roles during the plan and build phases, see the relevant sections.
3 Lifecycle of Upgrade Management

The figure below shows an overview of the most important upgrade project phases, the major objectives of each phase, and key project milestones.

This overview implies that the SAP Upgrade Roadmap mainly focuses on the execution of the upgrade project in the build phase. However, it also describes the most important planning tasks as part of the project preparation phase. In this document, these planning tasks, for example, an analysis of new functions and features, a definition of business requirements, and the creation of a business case, have been moved to the separate phases Upgrade Discovery and Upgrade Evaluation to outline that the related planning tasks are usually performed several months before a project organization is really set up.

Additionally, a possible outcome of the planning phase could also be that a release upgrade is not carried out or is postponed. Therefore, it makes sense to separate these planning phases from the implementation phase. If you conduct an upgrade project based on the SAP Upgrade Roadmap, you should regard the planning tasks as prerequisites for the project that are finally verified in the project preparation phase.

The following figure outlines the standard sequence of steps that are usually comprised in a technical upgrade. In the following subchapters, the objectives, tasks, and deliverables of each phase are described in more detail.
3.1 Plan

3.1.1 Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Organization</th>
<th>Tasks</th>
</tr>
</thead>
</table>
| Quality Manager for Protection of Investment | Application Management / Customer Center of Expertise | • Coordinate planning process  
|                                            |                                             | • Create upgrade master plan                      |
| Quality Manager for Business Process Improvements | Application Management / Customer Center of Expertise | • Document solution  
|                                            |                                             | • Facilitate collection and definition of business requirements |
| Business Stakeholders and Board representatives | Corporate Management Team                  | • Define corporate strategies and constraints  
|                                            |                                             | • Approve project                                 |

Figure 13: Sequence of Tasks in an Upgrade Project
### 3.1.2 Upgrade Discovery

**Objectives**
The goal of the Upgrade Discovery phase is to get an understanding of whether possible new software releases support your operational and strategic business objectives and requirements better than your current solution.

**Tasks**
- Identify and document current status of business solution
- Define business requirements
- Define IT requirements
- Obtain functional and technical information about new product release
- Map requirements to possible target solutions

**Deliverables**
- Documentation of existing business processes in a central business process repository
- Documentation of software components, hardware, and IT infrastructure used in the solution in a central solution landscape directory
- List of known functional gaps and future business and IT requirements
- Documentation of advantages and disadvantages of possible target software releases with regard to operational and strategic business and IT requirements

**Milestones**
There are no standard milestones for this phase.
3.1.3 Upgrade Evaluation

Objectives

In the Upgrade Evaluation phase you define the target release and the transition roadmap to the new solution. This includes defining the project goals and scope, which form the basis for the project master plan. You can then use these to create the business base and propose the upgrade project.

Tasks

- Define upgrade scope and goals
- Create upgrade master plan
- Create business case
- Get project approval

Deliverables

- Definition of target solution landscape (releases and software components)
- Definition of strategic IT requirements that should be accomplished in connection with the upgrade (for example, platform changes, hardware exchange, Unicode conversion)
- Evaluation of solution transition roadmaps describing pros and cons, decision criteria and final recommendation
- Upgrade master plan for recommended target landscape and transition path including high-level description of required change tasks, infrastructure investments, upgrade strategy, project setup, and project schedule
- Business case providing estimates for upgrade project, change, and maintenance costs of the future landscape, as well as a return on investment calculation
- Decision memo for project approval by steering committee

Milestones

- Upgrade Project Released

When this milestone is reached, all tasks of the discovery and evaluation phases have been completed. All relevant parties at the corporate level have been involved to define the project objectives and create and approve the overall business case. Finally, the upgrade project is decided by the corporate steering board. In case of an approval, a project leader is defined that takes over responsibility for all further activities of the upgrade implementation phase, which starts after passing this milestone.

3.2 Build

3.2.1 Roles

<table>
<thead>
<tr>
<th>Role</th>
<th>Organization</th>
<th>Tasks</th>
</tr>
</thead>
<tbody>
<tr>
<td>Project Manager</td>
<td>Business Units or Application Management</td>
<td>Manage project</td>
</tr>
<tr>
<td>Role</td>
<td>Organization</td>
<td>Tasks</td>
</tr>
<tr>
<td>----------------------------------</td>
<td>---------------------------------------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
</tbody>
</table>
| Project Quality Manager          | Application Management / Customer Center of Expertise | • Project quality management  
• Coordinate activities of CCoE Quality Manager |
| Project Sponsors and Business Stakeholders | Corporate Management, Business Units, and Program Management Office | • Supervise project |
| Test Coordinator                 | Business Units or Application Management          | • Organize and coordinate integration tests  |
| Training Coordinator             | Business Units or Application Management          | • Organize and manage user training  |
| Business Process Champions       | Business Units                                    | • Design business processes  
• Support test plan definition  
• Sign-off business changes  |
| Technology Specialist            | SAP Technical Operation and IT Infrastructure teams | • Upgrade infrastructure  
• Minimize technical downtime  |
| System Administrator             | IT Infrastructure Team                            | • Provide technical upgrade infrastructure  
• Adapt technical system security  |
| Authorization Specialist         | Security Team                                     | • Adapt authorization concept  |
| Key User                         | Business Units                                    | • Test business processes  
• Support go-live  |
| Developers                       | Custom Development Team                           | • Analyze and implement changes (coding, data model, customizing)  |
| Interface Experts                | Custom Development Team                           | • Analyze and implement changes to interfaces to legacy systems  |
| Third-Party Software Specialists | Custom Development Team                           | • Analyze and implement changes to non-SAP software  |
3.2.2  Project Preparation

Objectives
In this phase, you define the project plan and carry out the organizational preparation of the project so that all human and technical resources are in place ready for the upgrade project kick-off.

Tasks
- Create project structure plan and time schedule
- Plan temporary and future IT Infrastructure
- Facilitate delta training for project staff
- Define test concept

Deliverables
- Project plan describing the project organization with roles, tasks, and responsibilities, as well as exact timelines, milestones and deadlines
- Nominated and trained project resources (internal/external)
- Defined test focus and framework
- Prepared sandbox system for the first test upgrade

Milestones
- Project Prepared

The end of the project preparation phase is reached when a project plan and organization has been defined and the project team is ready to start the project work. After passing this milestone, the project is normally started with a kick-off meeting of all project members.

3.2.3  Upgrade Blueprint

Objectives
In this phase, you carry out the detailed planning and design of the new solution. Additionally, you finalize the procedure, required resources, and timelines for performing the upgrade.

Tasks
- Upgrade sandbox system
- Define future IT infrastructure
- Perform detailed planning of upgrade project based on test upgrade results and process design
- Define maximum business downtime and specify upgrade strategy and optimization measures (for example, archiving, usage of INCV)
- Define maintenance and code freeze strategy during the project
- Perform initial business tests
- Design business processes
- Start application adjustments
• Review solution operation concept
• Create test plan
• Plan end-user training and documentation
• Set up temporary maintenance landscape and prepare development system upgrade

**Deliverables**

• Refined project plan and timelines
• Fully specified new business processes, process enhancements, and replacements of custom developments with the SAP standard
• Adjustment requirements and documented adjustment activities (SPDD, SPAU, custom developments, customizing)
• Existing core business processes run in sandbox system without errors, at least for standard regression test scenarios
• Full documentation of upgrade procedure, issues, and problem resolutions exists in central upgrade script
• Documentation and first analysis of technical downtime performed during sandbox upgrade

**Milestone**

• Blueprint completed

The end of the blueprint phase is reached when the planned business processes to be implemented or upgraded as well as the baseline scope are described and the technical and integration design of the future solution is completed.

The customer project team, especially the business process owners, proceeds to the readout of the created business blueprint document that is then approved and signed-off by the customer’s project management – including project sponsor, steering committee and business process and IT owners. This sign-off is a formal approval of the blueprint for the SAP upgrade to proceed to the realization phase. This ensures that all parties have a common understanding of the scope, effort, schedule, and outcome of the project, as well as the remaining challenges to overcome.

### 3.3 Run

The overall duration of upgrade projects usually varies greatly depending on the specific conditions of your solution, as well as your project goals. However, when evaluating customer experiences, you can derive some general rules for the duration of upgrade projects and its main influencing factors. This is particularly true for ERP upgrades, for which SAP has built up an Upgrade Experience Database, which provides a statistical analysis of customer experiences. This section describes the Upgrade Experience Database and the major findings that you can derive from it.

#### 3.3.1 ERP Upgrades and Experience Database

The SAP Upgrade Experience Database provides experiences and statistics from completed upgrade projects. It provides benchmarking data or project statistics from SAP, gathered from completed upgrades by other customers.
It tracks the following important upgrade aspects:

- Additional hardware requirements
- Project duration
- Business downtime
- Reasons for upgrade

You can use the benchmark data to plan your upgrade and discover how other organizations experienced the upgrade project, with information gathered in similar upgrades. You access the upgrade experience database from the SAP Service Marketplace at http://service.sap.com/upgradedb.

### 3.3.2 Evaluation of Upgrade Experience

When analyzing the SAP Upgrade Experience Database, it is important to determine the minimum and maximum project duration exists for ERP upgrades. The minimum duration is mainly determined by the number of systems in the transport landscape to be upgraded and the number of additional technical tasks like OS or database updates. For a typical three-tier transport landscape this is about 3-4 weeks. However, such times can only be achieved if no, or only a few, application adjustment requirements exist and if the business scope of the system is small. The business scope is mainly defined by the number of used business scenarios, the number of users, and the business volumes.

At the other end of the duration spectrum, on rare occasions, projects can take longer than 18 months. Even for complex system landscapes for which multiple upgrades are performed, this period of time is sufficient to accomplish technical upgrades.

Within this timeframe, the project duration mainly depends on the upgrade efforts. The two main drivers for those efforts are the leap between start and target release versions and the number of custom developments and modifications.

The reason for the first effort driver is intuitively clear; the older a release is the bigger the delta is in SAP functionality and user experience. Therefore, the upgrade project efforts and duration increase with the distance between the start release and the target release.

![Timeline of Average Upgrade Phases from Release R/3 4.6c to ERP 6.0](image)

**Figure 14:** Timeline of Average Upgrade Phases from Release R/3 4.6c to ERP 6.0

The average overall duration for the project is 5 months (+/- 2 months). The statistic is based on 157 customer cases.
For release upgrades from releases older than R/3 4.6c and from ECC 5.0, no reliable statistical data exists. However, the existing data supports the trend above, that is, upgrades from ECC 5.0 take less than 4 months on average and releases older than 4.6c take more than 5 months.

Looking at the data for a specific start-target release combination, you can also see that the project duration and efforts depend on the business scope of the system. This correlation is simply because the more a system is used; the more it is affected by change events like upgrades. However, the data also shows that this is actually only valid for comparatively large release jumps. For example, you can clearly see this correlation with an upgrade from R/3 4.6c to ECC 6.0; however, this is typically not the case for upgrades from ECC 5.0 to ECC 6.0.

The average overall duration for the project is 4 months (+/- 2 months). The statistics is based on 85 customer cases.

The second major upgrade effort driver is the number of custom developments (modifications, custom enhancements, or developments in the customer namespace). The more custom developments, the more effort has to be spent adjusting and testing these developments. Therefore, it is easily possible that 30-50% of the overall upgrade budget is spent for these activities. This also clearly has an impact on project duration, even though the times do not necessarily increase linearly with the efforts because parallelization of work (by more resources) helps to limit the duration.

Other influencing factors of the project duration are activities that have to be done in addition to the pure technical upgrade. Therefore, the implementation of new functionality or business scenarios certainly can extend the project duration. How much depends on the scope of the implementation. Another factor is the required Unicode conversion of MDMP systems when upgrading to ERP 6.0. Particularly for systems on R/3 4.6c or below, the average project duration is extended by around two months. Single code page system that are converted or MDMP systems on SAP Basis 6.20 and above are affected less.

Finally, the project duration depends on the number of system upgrades that are in the scope of the project.

### 3.3.3 Upgrade Realization

#### Objectives

In this phase, the solution described in the blueprint phase is implemented and tested in a project environment, which consists of upgraded copies of the development and quality assurance system of the current production environment. The upgrade procedures are optimized and the technical downtime is minimized.
Tasks

- Upgrade development system
- Freeze development and IT projects that are not part of the upgrade project scope
- Set temporary maintenance development system live and start double maintenance period for emergency corrections in current production system
- Complete applications adjustments (in upgraded development system)
- Update solution operation concept
- Optimize technical downtime
- Upgrade quality assurance system
- Prepare integration test
- Prepare end-user training
- Perform main integration and acceptance test

Deliverables

- Upgraded development system and quality assurance system
- Double maintenance and code freeze phase for current production environment
- Completed and unit-tested changes or enhancements to business processes, customizing or custom developments in the development system
- Core business processes and interfaces tested in quality assurance system
- Updated solution operation tools and procedures tested in quality assurance system
- Documented open test issues and including an action plan with completion criteria and upgrade impact description exists
- Business downtime optimized to meet target time windows

Milestones

- Solution Built
  This milestone represents the point in time during the upgrade realization phase at which all necessary coding, data dictionary or interface adjustments have been completed and passed a first functional unit test. Additionally, the necessary upgrade delta customizing has been completed. In case of functional or strategic upgrades, the extensions and new functionalities have been implemented as well. When reaching this milestone, the development system has been upgraded and code freeze period has started.
- Integration, performance and system tests completed
  This milestone marks the end of all test activities. This includes all unit (developer) tests, application, interface, integration and mass tests as well as all system tests and test upgrades, except a possible final dress-rehearsal test. All test results have been documented and issues have been either resolved or at least addressed. Successfully tested functions are signed-off by business. At this point in time, the quality assurance system of the production landscape has been up-graded.
3.3.4 Preparation for Cutover

Objectives
The goals of this phase are to get the final approval for the upgrade of the production system by the project steering committee and to prepare this production system upgrade.

Tasks
- Define detailed cutover schedule
- Define support concept for stabilization phase including additional requirements for staffing, service level agreements, support processes, escalations paths, and support infrastructure during this phase
- Perform final integration and system tests
- Sign off production upgrade (project steering committee and business stakeholders)
- Conduct end-user training
- Roll out documentation
- Deploy new productive infrastructure
- Start upgrade preparation phase for long running tasks like incremental conversion (INCV)

Deliverables
- Cutover schedule and upgrade script
- Final integration and system tests completed without issues
- Infrastructure ready for production system upgrade

Milestones
- Cutover Prepared
The completion of the cutover planning for the production system upgrade is the final milestone before the production system upgrade. All test activities have been completed. Any critical issues have been resolved and documented in the upgrade script. The results and experiences of the tests have been used to create the cutover plan. An optional mock upgrade has been performed based on this cutover plan. A support and disaster handling concept for the production cutover and the stabilization phase after the upgrade has been defined.

As a result of the successful completion of this project phase, the technical upgrade of the production system is signed-off, which includes the go/no go decision by the steering board and the confirmation of the cutover plan.

3.3.5 Production Cutover and Support

Objectives
The objective of this phase is to upgrade the production system and to ensure that the core business processes function in the production system landscape.

Tasks
- Upgrade production system
- Perform final business user acceptance
- Release upgraded solution for user operation
• Provide support for go-live
• Hand over solution operation to the operating organization
• Close project

**Deliverables**

• Upgraded and released production system
• Standard operating organization with responsibility for the solution
• Signed off and closed project

**Milestones**

• Start of Production
  This milestone marks the end of the upgrade of the production system and the final sign-off by the customer’s project management including project sponsor, steering committee and business process and IT owners. The major goal of this milestone is to ensure that the core business processes function in the production system landscape. For this purpose, final acceptance tests by IT and business have been carried out. If the upgraded solution cannot be released, the system landscape has to be restored to the status prior to the upgrade.

• Handover to Production
  At this milestone, the solution operation organization resumes responsibility for running the solution. For this purpose, the project team transfers the solution and the updated administration concept to the operation organization. Prerequisite for this step is that the upgraded solution runs stable, consistent, and is performing sufficiently. After this milestone, the project is closed and the project organization dissolved.

### 3.4 Optimize

The quality management of upgrade projects should follow the established best practices for quality management of any type of project. One fundamental concept within these best practices is the introduction of special check or milestone sign-off points called quality gates Q-Gates). At a Q-Gate, the project situation is assessed against an agreed set of quality criteria. This can take place at a milestone (point in time) or at an important phase of a project. Establishing Q-Gates helps you to achieve the following benefits:

• Review the status and progress of the open risks and issues identified the previous Q-Gate
• Identify new issues and risks
• Take corrective actions throughout the project lifecycle
• Ensure that the deliverables satisfy the given requirements with regards to quality standards
• Reduce the risk of failure
• Keep the project efforts aligned with the project targets

You normally schedule these quality checks of an SAP project prior to the deadlines for all major deliverables. The outcome of these assessments provides important input for the customer project stakeholders to evaluate whether the criteria to pass a Q-Gate and proceed to the next phases have been met.
You do not have to include every single Q-Gate. Depending on your situation and engagement, some of them can be skipped.

There are three possible results at Q-Gate:

- **Green**: All deliverables, key activities, and documents have been completed and accepted prior to this Q-Gate. There is no risk to the project.

- **Yellow**: Identified documents, key activities, or deliverables have either not been completed, are in the process of being completed, or have not been accepted as satisfactory by the client. With client approval, you can proceed beyond the Q-Gate. However there is elevated risk to the project, so proceed with caution.

- **Red**: One or more critical deliverables or activities have not been completed or have not been completed to the satisfaction of the client. SAP does not recommend continuing past this Q-Gate because there is a high risk to the project. Client override is necessary to move forward with the project.

The project quality manager must ensure that all relevant quality gates are part of the project schedule. Ideally, this role should be assigned to the CCoE quality manager responsible for protection of investment or another member of the application management team. As a result, the project schedule should be adapted to include relevant Q-Gates, and the resource plan should take into consideration that the experts must be made available to support the underlying activities.

SAP offers the SAP Safeguarding for Upgrade service to perform or support the quality management project. In this case, the SAP Technical Quality Manager, established with the Safeguarding for Upgrade engagement, can take over the role of the project quality manager.
4 Training

SAP offers a variety of training courses across a range of topics. For more information, see the SAP Training and Certification Shop at https://training.sap.com.

4.1 Expert Guided Implementation Sessions

For Enterprise Support Customers, SAP offers Expert Guided Implementation Sessions (EGI). Expert Guided Implementation (EGI) sessions are a combination of remote training, live configuration, and on-demand expertise, which allow you to perform complex activities with the help of experienced SAP support engineers. The instructor will demonstrate what to do step by step. Afterwards, you can perform the relevant steps in your own version of SAP Solution Manager. If you have any questions, you can then contact an SAP expert via phone or e-mail.

More information regarding the content of the EGIs and time schedule can be found at the EGI calendar at http://www.service.sap.com/~sapidb/011000358700001780312008E
## More Information

### Getting Started

<table>
<thead>
<tr>
<th>Topic</th>
<th>Information Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Upgrade Infocenter</td>
<td><a href="https://support.sap.com/release-upgrade-maintenance/upgrade-info.html">https://support.sap.com/release-upgrade-maintenance/upgrade-info.html</a></td>
</tr>
<tr>
<td>SAP Upgrade News</td>
<td><a href="http://service.sap.com/upgrade-news">http://service.sap.com/upgrade-news</a></td>
</tr>
<tr>
<td>ERP Upgrade in the SAP Community Network</td>
<td>Forum</td>
</tr>
<tr>
<td>Overview and details of SAP ERP Upgrades</td>
<td><a href="http://service.sap.com/upgrade-erp">http://service.sap.com/upgrade-erp</a></td>
</tr>
<tr>
<td>SAP Enhancement Packages</td>
<td>SAP ERP: <a href="http://service.sap.com/erp-ehp">http://service.sap.com/erp-ehp</a></td>
</tr>
<tr>
<td></td>
<td>SAP NetWeaver: <a href="https://www.sdn.sap.com/irj/sdn/nw-mainreleases">https://www.sdn.sap.com/irj/sdn/nw-mainreleases</a> &gt; choose NW 7.0 (or later) &gt; enhancement packages</td>
</tr>
</tbody>
</table>

### Upgrade Roadmaps and Best Practices

<table>
<thead>
<tr>
<th>Topic</th>
<th>Information Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>Upgrade project plan, standards and guidelines</td>
<td>SAP Upgrade Roadmap: <a href="http://service.sap.com/upgraderoadmap">http://service.sap.com/upgraderoadmap</a></td>
</tr>
</tbody>
</table>

### Planning an Upgrade Project

<table>
<thead>
<tr>
<th>Topic</th>
<th>Information Sources</th>
</tr>
</thead>
<tbody>
<tr>
<td>SAP Maintenance strategy</td>
<td><a href="http://service.sap.com/maintenance">http://service.sap.com/maintenance</a></td>
</tr>
<tr>
<td>Technical prerequisites and limitations</td>
<td>Product availability matrix: <a href="http://service.sap.com/pam">http://service.sap.com/pam</a></td>
</tr>
<tr>
<td>Scope and Effort Analyzer</td>
<td><a href="http://service.sap.com/SEA">http://service.sap.com/SEA</a></td>
</tr>
</tbody>
</table>
### Usage & Procedure Logging
- Information Sources: [http://service.sap.com/~sapidb/011000358700000427102012E](http://service.sap.com/~sapidb/011000358700000427102012E)

### Unicode conversion
- Information Sources: [http://service.sap.com/unicode](http://service.sap.com/unicode), [http://service.sap.com/unicode@sap](http://service.sap.com/unicode@sap)

### Availability of country versions
- Information Sources: [http://service.sap.com/localization](http://service.sap.com/localization)

### Dependencies in your application landscape when upgrading a system
- Information Sources: Upgrade dependency analyzer: [http://service.sap.com/uda](http://service.sap.com/uda)

### Benchmark information for project duration, project effort, and so forth for preparing upgrade project cost and effort estimation
- Information Sources: [SAP upgrade experience database: http://service.sap.com/upgradedb](http://service.sap.com/upgradedb)

### Media library in the Upgrade Information Center containing useful presentations, e-learnings, customer testimonials, and much more
- Information Sources: [http://service.sap.com/upgrade → Media Library](http://service.sap.com/upgrade → Media Library)

### Planning Landscape Changes
- Information Sources: [https://websmp208.sap-ag.de/~sapidb/011000358700003999172005E.pdf](https://websmp208.sap-ag.de/~sapidb/011000358700003999172005E.pdf)

## Executing an Upgrade Project

### Upgrade Guides
- Information Sources: [http://service.sap.com/instguides](http://service.sap.com/instguides)

### SAP Solution Manager
- Information Sources: [http://service.sap.com/solutionmanager (general)](http://service.sap.com/solutionmanager (general)), [Solution Manager Tools for Upgrade](http://service.sap.com/solutionmanager (general))

### Application server resizing
- Information Sources: [http://www.service.sap.com/sizing](http://www.service.sap.com/sizing)

### Getting Started with the ABAP Test Cockpit for Developers
- Information Sources: [http://scn.sap.com/docs/DOC-32628](http://scn.sap.com/docs/DOC-32628)

### Application Specific Upgrade toolbox
- Information Sources: [http://service.sap.com > ASU](http://service.sap.com > ASU)

### Testing SAP solutions
- Information Sources: [http://service.sap.com/testing](http://service.sap.com/testing)

## SAP Upgrade Services

### SAP Upgrade Service Portfolio
- Information Sources: [http://service.sap.com/upgradeservices](http://service.sap.com/upgradeservices)

### SAP Enterprise Support
- Information Sources: [https://support.sap.com/support-programs-services/programs/enterprise-support.html](https://support.sap.com/support-programs-services/programs/enterprise-support.html)

### SAP Safeguarding for Upgrade
- Information Sources: [https://support.sap.com/support-programs-services/programs/activeembedded.html](https://support.sap.com/support-programs-services/programs/activeembedded.html)